Automated Negotiation of Collaboration-Protocol Agreements Specification Version 0.10

OASIS ebXML Collaboration Protocol Profile and Agreement Technical Committee

Date TBD

Status of this Document

This document specifies an ebXML SPECIFICATION for the eBusiness community.

Distribution of this document is unlimited.

The document formatting is based on the Internet Society's Standard RFC format.

This version:
URL TBD

Errata for this version:
URL TBD

***Previous version*:**
URL TBD

1 Automated Negotiation Sub Team Members

Martin Sachs, Cyclone Commerce, team lead
Keith Babo, Sun Microsystems
Jamie Clark, OASIS staff
Brian S. Hayes, Extended Thinking (individual member)
Neelakantan Kartha, Sterling Commerce
Kevin Liu, SAP
Heiko Ludwig, IBM
Monica Martin, Sun Microsystems
Dale Moberg, Cyclone Commerce
Himagiri Mukkamala, Sybase
Peter Ogden, Cyclone Commerce
Yukinori Saito, ECOM
Nikola Stojanovic, Encoda Systems
Jean Zheng, Vitria

2 Table of Contents

3 Introduction

3.1 Summary of Contents of Document

This document contains a specification for automatically negotiating the contents of an ebXML *Collaboration Protocol Agreement* (*CPA*)[ebCPP]. This specification is a component of the suite of ebXML specifications.

This document is organized as follows:

- Section 3 introduces the specification and discusses various procedural matters
- Section 4 summarizes the design objectives.
- Section 5 is a system-level overview.
- Section 6 discusses the *CPA Template*.
- Section 7 discusses content of *CPPs* and *CPA Templates* with respect to negotiation.
- Section 8 gives the rules for constructing a *Negotiation CPA*, the *CPA* that governs the *Negotiation Protocol*.
- Section 9 discusses negotiability of elements and attributes in the *CPA*.
- Section 10 defines and discusses the *Negotiation Descriptor Document (NDD*) that is used to describe offers and counter offers.
- Section 11 defines the contents of the *Negotiation Message*s.
- Section 12 defines the *Negotiation Protocol* including the ebXML *Business Process Specification Schema*[ebBPSS] instance *Document* that is used to describe the Negotiation Transactions and their choreography.
- Section 13 discusses negotiation algorithms.
- The appendices include XML Schemas for the *NDD* and *Negotiation Messages*, the negotiation BPSS instance *Document*, examples of an *NDD* instance *Document* and negotiation *Message* instance *Documents*, non-normative aspects of *CPA* composition, and a glossary of terms.

3.2 Definition and Scope of this Specification

The goal of this specification is to define a means of automatically negotiating the contents of a *CPA*. The focus is on negotiating both long-term partner relationships and spontaneous (perhaps for a single business exchange) relationships. Automated negotiation of *CPAs* is a critical element of spontaneous e-commerce since it will enable business to be conducted with minimal delay, as soon as two potential trading partners discover each other. Automated negotiation also will enhance the ability of an enterprise to maintain large numbers of partner relationships. It will reduce the need for manual intervention in maintaining those relationships, thereby simplifying life-cycle management of the relationships.

This specification defines the rules for automated negotiation of *CPAs*. It defines the *Negotiation Protocol* and the contents of the *Documents* that are part of the *Negotiation Protocol*.

3.3 Document Conventions

Terms in *Italics* are defined in Appendix H or in the glossary of the CPPA specification[ebCPP]. Terms listed in ***Bold Italics*** represent the element and/or attribute content of the XML *CPP,* *CPA*, or related definitions.

In this specification, the term “item”, when used in the context of an *NDD* or counter offer *Message* denotes an element, attribute, or subtree that is negotiable.

The term “BPSS instance *Document*” refers to an XML document that is an instance *Document* of the XML schema of the *Business Process Specification Schema*[ebBPSS] ebXML specification.

In this specification, indented paragraphs beginning with "NOTE:" provide non-normative explanations or suggestions that are not mandated by the specification.

References to external documents are represented with BLOCK text enclosed in brackets, e.g. [RFC2396]. The references are listed in Section 14.

The keywords MUST, MUST NOT, REQUIRED, SHALL, SHALL NOT, SHOULD, SHOULD NOT, RECOMMENDED, MAY, and OPTIONAL, when they appear in this document, are to be interpreted as described in [RFC 2119].

> NOTE: Vendors SHOULD carefully consider support of elements with cardinalities (0 or 1) or (0 or more). Support of such an element means that the element is processed appropriately for its defined function and not just recognized and ignored. A given *Party* might use these elements in some *CPPs, CPAs,* negotiation *Messages*, or *NDDs* and not in others. Some of these elements define parameters or operating modes and SHOULD be implemented by all vendors. It might be appropriate to implement elective elements that represent major run-time functions, such as various alternative communication protocols or security functions, by means of plug-ins so that a given *Party* MAY acquire only the needed functions rather than having to install all of them.

By convention, values of [XML] attributes are generally enclosed in quotation marks; however those quotation marks are not part of the values themselves.

3.4 Versioning of the Specification, Schema, and Related Documents

Whenever this specification is modified, it SHALL be given a new version number.

It is anticipated that during the review period, errors and inconsistencies in the specification and in the schemas may be detected and have to be corrected. All known errors in the specification as well as necessary changes to the schema will be summarized in an errata page found at

TBD http://www.oasis-open.org/committees/ebxml-cppa-negotiation/documents/cppa-negotiation-Errata.shtml

The specification when approved SHALL carry a version number of “1_0”. At that time, the schemas SHALL have a version number of “1_0a” and the suffix letter after “1_0” will be advanced as necessary when bug fixes to the schemas have to be introduced. Such versions of the schemas SHALL be found at the following URLs:

Deleted: <#>Versioning of the Specification, Schema, and Related Documents¶
Whenever this specification is modified, it SHALL be given a new version number. ¶
¶
It is anticipated that during the review period, errors and inconsistencies in the specification and in the schemas may be detected and have to be corrected. All known errors in the specification as well as necessary changes to the schema will be summarized in an errata page found at¶
¶
TBD http://www.oasis-open.org/committees/ebxml-cppa-negotiation/documents/cppa-negotiation-Errata.shtml¶
¶
The specification when approved SHALL carry a version number of “1_0”. At that time, the schemas SHALL have a version number of “1_0a” and the suffix letter after “1_0” will be advanced as necessary when bug fixes to the schemas have to be introduced. Such versions of the schemas SHALL be found under the directory¶
¶
TBD http://www.oasis-open.org/committees/ebxml-cppa/schema/NegotiationMessage.xsd¶
¶
TBD http://www.oasis-open.org/committees/ebxml-cppa/schema/NDD.xsd¶
¶
In addition, the latest version of the schemas SHALL always be found at¶
¶
TBD http://www.oasis-open.org//schema/2_0.xsd¶
¶
These URIs are also the namespace URIs used for the schemas defined by this specification. The latest versions will then be directly resolvable from the namespace URI.¶
¶
TBD SHOULD WE HAVE A VERSION ATTRIBUTE IN THE SCHEMAS?¶
The value of the version attribute of the Schema element in a given version of the schemas SHALL be equal to the version of the schema.¶

Formatted: Bullets and Numbering

TBD http://www.oasis-open.org/committees/ebxml-cppa/schema/NegotiationMessage.xsd

TBD http://www.oasis-open.org/committees/ebxml-cppa/schema/NDD.xsd

In addition, the latest version of the schemas SHALL always be found at

TBD http://www.oasis-open.org//schema/2_0.xsd

These URIs are also the namespace URIs used for the schemas defined by this specification. The latest versions will then be directly resolvable from the namespace URI.

SHOULD WE HAVE A VERSION ATTRIBUTE IN THE SCHEMAS?
The value of the version attribute of the Schema element in a given version of the schemas SHALL be equal to the version of the schema.

3.5 Definitions

Technical terms related to the subject of this specification are defined in Appendix H. Technical terms related to *Collaboration Protocol Profiles* and *Agreements* and to the overall vocabulary of ebXML are defined in [ebCPP].

3.6 Audience

One target audience for this specification is implementers of ebXML services and other designers and developers of middleware and application software that is to be used for conducting electronic *Business.* Another target audience is the people in each enterprise who are responsible for creating *CPPs* and *CPAs.*

3.7 Assumptions

It is expected that the reader has an understanding of XML and is familiar with the ebXML CPPA specification[ebCPP].

3.8 Related Documents

Related documents include ebXML specifications on the following topics:

- ebXML Collaboration Protocol Profile and Agreement Specification[ebCPP]
- ebXML Business Process Specification Schema[ebBPSS]
- ebXML Message Service Specification[ebMS]

See Section 14 for the complete list of references.

3.9 Acknowledgments

- To Duane Nickull, XML Global, for his ebXML Automatic CPA Negotiation proposal, Feb, 14, 2001.
- To The ebXML *Business Process* Team, for its automated contract negotiation pattern in [bpPATT].

4 Design Objectives

This specification defines the protocol, *Messages*, and *Business Documents* associated with automatically negotiating the contents of a *CPA*. It does NOT define negotiation algorithms in detail. The negotiation algorithm is part of the private process at each *Party* and MAY embody private or proprietary strategies. This specification does define the rules that ensure interoperability between two *Parties'* negotiation algorithms.

Following are the objectives for the design of this specification.

- The design is based on negotiating the contents of a *CPA* starting with a *CPA Template* (draft *CPA*) that one prospective trading partner sends to the other as an initial offer. See Section 6.2 for a discussion of *CPA Template* and draft *CPA*. A *CPA Template* contains elements and attributes that need to be negotiated with a prospective trading partner. A *Party* can publish a *CPA Template* in a *Registry* or can create one from its *CPP* and the prospective trading partner's *CPP*.
- The specification defines the *Negotiation Protocol* transactions and choreography by means of an ebXML *Business Process Specification Schema*[ebBPSS] instance *Document*.
- The *Negotiation Protocol* is governed by a *Negotiation CPA* (*NCPA*). The *NCPA* is a standard ebXML *CPA* that defines a minimal set of function that all *Parties* can be expected to support without *Parties* having to negotiate the *NCPA* before negotiating the *CPA* for their *Business Collaboration*.
- Avoid requiring changes to the CPPA and BPSS specifications, at least for version 1 of the negotiation spec.
- Use deterministic algorithms
- The *Negotiation Process* SHOULD converge rapidly.
  - ♦ The process SHOULD either succeed or fail.
  - ♦ The process SHOULD invoke human intervention on failure
  - ♦ The design SHOULD avoid deadlock such as iterative loops that don't advance the state of the negotiation. An example is reiteration over the same offer or counter offer that was previously rejected by either or both parties.
    - ▪ The specification SHOULD state rules that avoid such iterative loops even if it is decided that automatic detection of loops is out of scope for version 1.
- It MUST be absolutely clear at any point in the negotiation which *Party* (i.e., only one *Party*) has the initiative to send the next request (counter offer).
  - ♦ The design SHOULD avoid race conditions in which both parties simultaneously send an a counter offer. The choreography should make this an error condition.

    NOTE: It is probably not possible to avoid or detect the case where two *Parties* send each other initial offers. This condition should be recognized by people.

- The design SHOULD minimize the amount of state that has to be saved.
- Offer rejection semantics SHOULD be strong; rejection SHOULD not be a tactical maneuver.

5 System Overview

The *CPA Negotiation Protocol* begins when one *Party* makes an initial offer to a second *Party*. The initial offer consists of a *CPA Template* and a *Negotiation Descriptor Document (NDD)* that describes what is negotiable in the *CPA Template*. See the discussion in Section 6.2.

In the *CPA Negotiation Protocol*, a *CPA Template* is verified as suitable for both *Parties* and modified until a suitable *CPA* is constructed. It might also be discovered that agreement cannot be reached until one *Party* (or both) acquires additional software capabilities. The term "*CPA Template*" was chosen to emphasize its use as the starting point for *CPA* negotiation. In general, a *CPA Template* constitutes a proposal about an overall binding of a *Business Process* to a delivery agreement with some items left open; negotiation is then used to arrive at detailed values for the open items in order to achieve a final agreement. The *NDD* identifies what items have to be negotiated and defines ranges or sets of acceptable values for those items.

5.1 Main Components of CPA Negotiation

Figure 1 illustrates the main components of *CPA* negotiation.

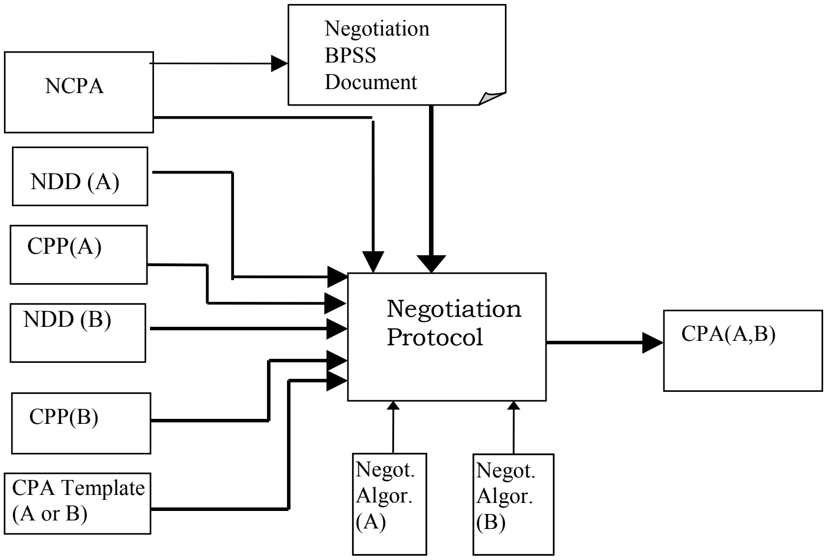


Figure 1. Components of *CPA* Negotiation

The following entities are shown in the figure:

- NCPA: The Negotiation CPA controls the Negotiation Protocol.

- Negotiation BPSS Instance *Document*: An ebXML *Business Process Specification Schema*[ebBPSS] instance *Document* that is used to define the negotiation collaborative protocol. This BPSS instance *Document* is referenced from an *NCPA*.
- *CPP*: Parties A and B publish their *CPPs* in an ebXML *Registry*[ebRS] or otherwise exchange them when they discover each other.
- *CPA Template*: A *CPA* in which some items remained to be filled in by one or the other *Party*, or negotiated between them.
- *NDD*: The *Negotiation Descriptor Document*, a *Document* associated with a *CPP* or a *CPA Template* that defines what is negotiable, ranges of numeric values, etc. The *NDD* is used in the *Negotiation Protocol*. Each *Party* might have a separate *NDD* that refers to its own *CPP* and represents its own viewpoint of what is negotiable, acceptable ranges of values, etc.
- *Negotiation Messages*: The *Messages* used to exchange offer and counter-offer information between negotiating *Parties*.
- *Negotiation Protocol*: The collaborative protocol that produces a negotiated *CPA*. Although shown as a single box in this figure, the *Negotiation Protocol* is executed between the two *Parties* or between each *Party* and an intermediary.
- Negotiation algorithm: The negotiation algorithm is the private process at each negotiating *Party* that implements that *Party's* private negotiation strategy. Note that the *Negotiation Protocol* is distinct from the negotiation algorithm. The former is the public protocol, captured by the BPSS instance *Document*. Each *Party* uses its negotiation algorithm, in conjunction with the *CPA Template*, *CPPs* and the *NDD*, to arrive at an offer or counter offer in the *Negotiation Protocol*. The negotiation algorithm is out of scope for version 1 of this specification. See Section 13 for additional discussion.

Two *Parties* can negotiate a *CPA* as follows. First, they publish their *CPPs* in an ebXML *Registry*, or similar *Registry*, so that potential trading partners can discover them. A *Party* MAY publish an *NDD* along with the *CPP*. This *NDD* describes what is negotiable in the *CPP*.

When *Party* B discovers *Party* A as a potential trading partner, *Party* B composes a *CPA Template* from its own *CPP* and *Party* A's *CPP*. If *Party* A published an *NDD* along with its *CPP*, *Party* B MAY use the information in *Party* A's *NDD* along with its own *NDD* in composing the *NDD* for the initial offer.

Alternatively, *Party* A MAY publish a *CPA Template* and *NDD*. In that case, *Party* B creates an initial offer by filling in basic information about itself (e.g. its *Party* ID and transport endpoint address). It then creates a new *NDD* by adding its own negotiability information to that from *Party* A's *NDD*.

In order to negotiate, *Parties* A and B MUST obey the rules defined by an *NCPA*. The *NCPA* MAY be virtual or real. "Virtual" means that the two *Parties* MUST configure their systems to conform to a published default *NCPA*. This avoids the need to negotiate the details of an *NCPA* before negotiating a *CPA*. *The Parties* can obtain each other's endpoint address by out-of-band means such as phone, fax, or discovery in an ebXML or other registry. See Section 12.7 for additional discussion.

Deleted: conform to the *NCPA* defined in Appendix C but there might not be a physical *NCPA* document

NOTE: It is the intention of the OASIS CPPA negotiation subcommittee to design one or a family of suitable default *NCPAs*.

“Real” means that the *Parties* create and deploy a specific *NCPA* using, for example, the following procedure:

1. *Parties* A and B publish *NCPA Templates* (that they are willing to abide by) in a *Registry*. They are *NCPA Templates* (as opposed to *NCPAs*) because some information (such as the prospective trading partner’s *Party* ID and endpoint address) is missing from an *NCPA* template. In many cases, a *Party’s NCPA Templates* might differ from each other only with regard to which of several negotiation BPSS instance *Documents* they refer to.
2. *Party* B discovers *Party* A and wants to conduct trade.
3. *Party* B chooses an *NCPA* template of *Party* A that it can live with (say, by looking at the BPSS instance *Document* pointed to by this *NCPA* template).
4. *Party* B then fills in this *NCPA* template with its own name, endpoint address, etc. (so that now it becomes an *NCPA*) and sends it to *Party* B.
5. *Party* B then starts the *Negotiation Protocol* by sending an initial offer to *Party* A. The initial offer consists of a *CPA* template and the corresponding *NDD.*

The two *Parties* can then perform the *Negotiation Protocol*, exchanging counter offers until they create an agreed *CPA*. They are then ready to do electronic *Business*.

5.2 Overview of CPA Negotiation

Figure 2 is a high-level view of the *Negotiation Process*. Following are some details of the *Negotiation Process* illustrated in Figure 2.

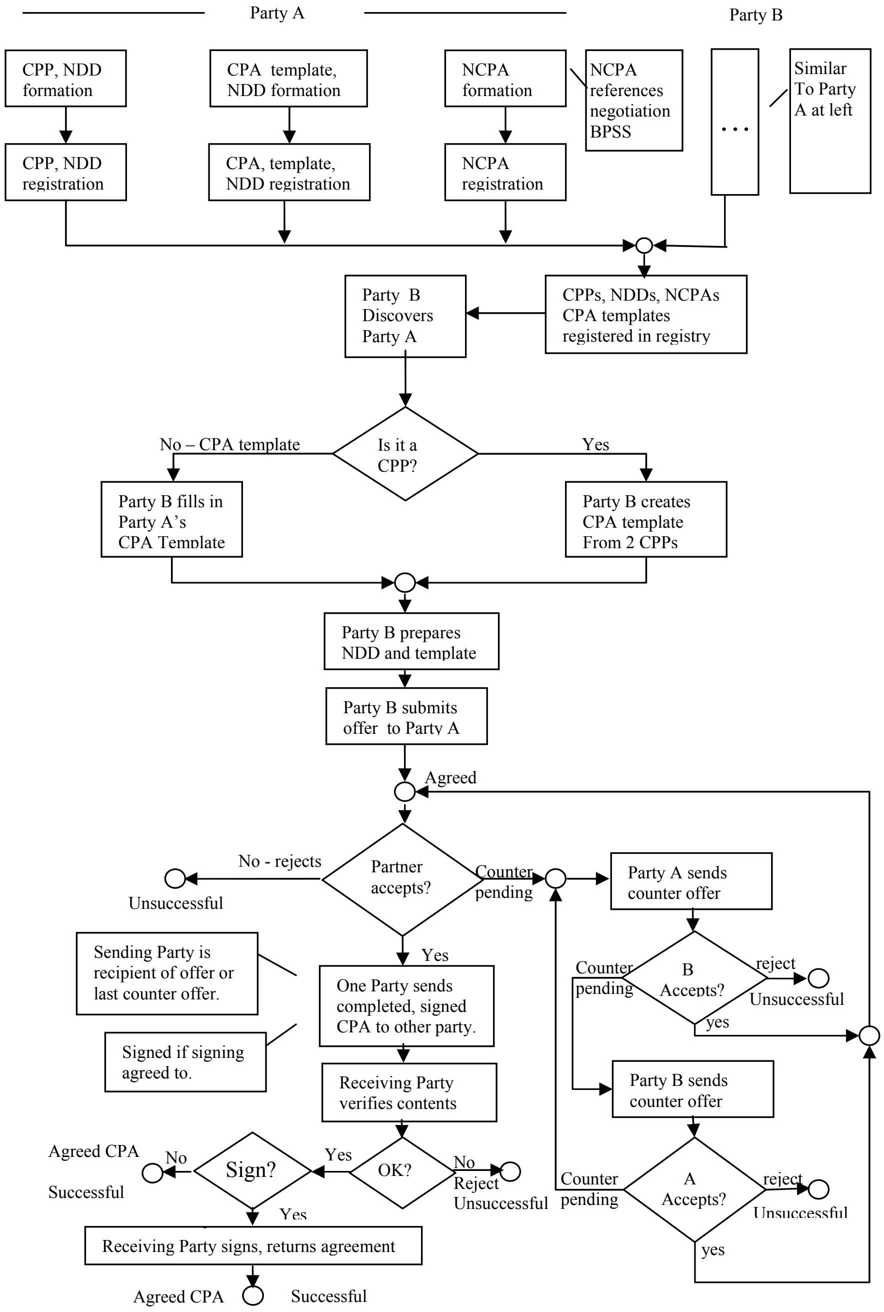

Party A
Party B
CPP, NDD formation
CPA template, NDD formation
NCPA formation
NCPA references negotiation BPSS
. . .
Similar To Party A at left
CPP, NDD registration
CPA, template, NDD registration
NCPA registration
Party B Discovers Party A
CPPs, NDDs, NCPAs CPA templates registered in registry
No – CPA template
Is it a CPP?
Yes
Party B fills in Party A's CPA Template
Party B creates CPA template From 2 CPPs
Party B prepares NDD and template
Party B submits offer to Party A
Agreed
No - rejects
Unsuccessful
Partner accepts?
Counter pending
Party A sends counter offer
Yes
Sending Party is recipient of offer or last counter offer.
One Party sends completed, signed CPA to other party.
Signed if signing agreed to.
Counter pending
B Accepts?
reject
Unsuccessful
yes
Party B sends counter offer
Receiving Party verifies contents
Agreed CPA
Successful
No
Sign?
Yes
OK?
No
Reject
Unsuccessful
Counter pending
A Accepts?
reject
Unsuccessful
Yes
yes
Receiving Party signs, returns agreement
Agreed CPA
Successful

Figure 2. Negotiation Process

- Initial inputs:
  - *CPPs* and the associated *NDDs* of two prospective partners or a *CPA Template* and *NDD* that one partner provides to a prospective partner.
    - For the case of the *CPA Template* and *NDD*, the *CPA Template* might be generated by one of the *Parties*, might be a copy of a *CPA* used by someone else that is almost exactly what is needed, or might be supplied by a third-*Party* negotiation service.
  - Proposed Process-Specification *Document* (BPSS instance document)
    - The *Parties* can negotiate about which BPSS instance *Document* to use based on the name of the BPSS instance *Document* (i.e. syntactic negotiation) but not over the details within a given BPSS instance *Document* (semantic negotiation).
- One *Party* prepares a *CPA Template* and an *NDD* that describes what is negotiable and submits the *CPA Template* and *NDD* to the other *Party* as an initial offer.
- The two Parties then exchange counter offers until they arrive at a mutually acceptable *CPA*. Offer and counter-offer information is in *Negotiation Messages* exchanged using negotiation *Business Transactions* defined in the *NCPA* and BPSS instance *Document*.
- Result of negotiation:
  - A successful result is a *CPA* that is ready to sign and use, possibly subject to human approval.
  - An unsuccessful result means that agreement was not possible on some items in the *CPA*. Possibly, further human interaction could resolve the incompatibilities.
- Concluding negotiation
  - The *Party* that received the last counter offer builds the complete *CPA* by filling in details such as its *Party* ID and transport endpoint address and sends it to the other *Party*. (If it is the case that no counter offers were received during the *Negotiation Protocol*, that is, if the *Party* that received the initial offer accepted it without sending a counter offer, that *Party* builds the complete *CPA* by filling in details such as its *Party* ID and transport endpoint address and then sends it to the other *Party*)
    - If it was agreed that the *CPA* is to be signed, the *Party* that sends the final *CPA* signs it before sending it.
  - The other *Party* verifies the contents of the completed *CPA* including, perhaps validation of the first *Party's* signature. If these tests are successful, that *Party* signs the new *CPA* (if signing was agreed to) and returns it to the first *Party*.
  - The two *Parties* now deploy the new *CPA* and begin doing business.

5.3 Pre-Conditions for Negotiation

This section discusses conditions that MUST be met before negotiation. If these conditions are not met, a successful outcome is unlikely. The discussions relate to *CPPs* or a *CPA Template* as appropriate

The two partners MUST agree on what *Negotiation Protocol* to follow, i.e. what *NCPA* to use for negotiation. (The *NCPA* identifies the *Negotiation BPSS Instance Document* to be used.)

There MUST be a minimum level of matching (i.e. compatibility) between two *CPPs.*

- There MUST be at least one transport protocol in common.
- There MUST be a minimum level of compatibility between at least one ***DocumentExchange***

element in each *CPP*.

Deleted: See Section 7 for related information.¶

5.4 CPP and NDD Formation and Editing

NOTE: These are pre-discovery steps that are out of scope for the negotiation specification, they are included here in the interest of completeness. Following is a non-normative list of the elements of *CPP* and *NDD* formation.

- *CPP* template (a prototype CPP that could be used for creating CPPs).
  - ♦ Supplied with software installation (configured options)
  - ♦ Edited to reflect preferences
- *NDD* formation.
  - ♦ Although *NDD* formation is out of scope, the *NDD* schema is a key component of this specification.
- Tool for custom *CPP* formation
- Tool for *CPA* and *CPA Template* formation.
- Tool for *NDD* formation
- Service(s) for supplying *CPPs* or *CPA Templates*
  - ♦ UDDI advertised, SOAP, ebXML, simple HTTP GET, and so on.
- ebXML *Registry* submission (publication)

In principle, a *Party* SHOULD be able to publish both a *CPP* and a *CPA Template*. However, this would lead to a problem that a given prospective trading partner might find either one. If a *Party* intends that some prospective trading partners negotiate with a *CPP* while other are expected to accept a *CPA Template*, then the *Party* SHOULD probably publish only the *CPP* and decide whether to send a *CPA Template* based on its knowledge of who the prospective trading partner is.

5.5 Discovery of CPPs and CPA Templates

NOTE: The discovery process is out of scope for the negotiation specification; this discussion is included here in the interest of completeness. Following are some points concerning the discovery process.

Deleted: it

- The minimum requirement is to be able to perform an HTTP GET of a CPP from a URL obtained by means outside the scope of this specification.
- UDDI ebXML *Registry* bootstrap. This permits CPPs to be advertised in either UDDI or the ebXML Registry.
- Search and retrieval in ebXML *Registry* or similar *Registry*.
- Well-known address of the *Registry*.
- A *Registry* might provide added function, perhaps as value-added services, such as
  - ♦ Notification of *CPP* expirations,
  - ♦ Acceptance filled-out *CPA Templates*,

Deleted: Can a

Deleted: further

Deleted: ?

Deleted: ?

Deleted: ?

5.6 Negotiation through an Intermediary

NOTE: Negotiation through an intermediary (negotiation broker) is out of scope for this version of the specification. A *Message*-forwarding intermediary that is not aware of the purpose of the *Messages* can be used if it conforms to the manner in which [ebMS] supports intermediaries.

6 CPA Template

This section provides an overview of the use of a *CPA Template*.

6.1 CPA Template and Draft CPA

The *Negotiation Protocol* defined in this specification is based on the use of a *CPA* that is incomplete in that it contains items that are negotiable or MUST be filled in by the *Party* that receives an initial offer. Negotiable items can have "dummy" values that will later be replaced by the agreed values arrived at during the *Negotiation Process*. Such an incomplete *CPA* can be categorized as a *CPA Template* or a draft *CPA*.

A *CPA Template* will normally contain dummy values for the *Party*-specific values of the *Party* to which the *CPA Template* is being sent as well as dummy values for other items that the offering *Party* considers negotiable. A draft *CPA*, on the other hand, will typically have been formed by pruning and combining *CPPs* of each of the *Parties* in the *Negotiation Process*, and so can contain all "real" values. That is, using a *CPA Template* will typically require a counter proposal while using a draft *CPA,* the *Party* making the initial offer might only be asking for approval of the draft *CPA* rather than offering to negotiate some items. For convenience, both kinds of *Document*, though having different origins, will be referred to by the term "*CPA Template*" because the process of negotiation proceeds the same way for either *CPA Templates* or draft *CPAs*. Sensible use of *CPA Templates* requires that the dummy values be indicated as negotiable and that acceptance does not occur until the dummy values have been replaced. In this specification, the *NDD* is the means of indicationg what is negotiable.

A *CPA Template* can encompass a wide range of negotiating possibilities. At one end of the range, it might amount simply to a take-it-or-leave-it offer, its *NDD* indicating only those items that MUST be filled in to customize it to the other *Party*. At the other end of the spectrum, its *NDD* might indicate that virtually everything is negotiable.

In the simplest case, the accompanying *NDD* might be very simple and would simply indicate which elements and attributes need to be completed by the prospective trading partner, such as *Party* ID and transport endpoint address. For this case, the *NDD* facilitates identifying the items to be filled in, avoiding the need to label the items to be filled in within the *CPA Template* and the need to parse the *CPA Template* to find those items.

6.2 Advantages of Starting Negotiation with a CPA Template

If negotiation is performed with the two *Parties' CPPs* and an *NDD* for each, everything in the *CPPs* is potentially negotiable and has to be considered during the *Negotiation Process*. The process of composing a *CPA Template* from two *CPPs* will often narrow down the amount of negotiation relative to the negotiation possibilities expressed in the *NDDs* that accompany the *CPPs*. The reason is that many of the differences between the two *CPPs* can be "mechanically" resolved by finding compatible choices and matching values of some elements or attributes. For example, there might be only one transport protocol that is common to the two *Parties*. After the *CPA Template* is constructed, a new *NDD* MUST be constructed that includes only the items in the *CPA Template* that remain to be negotiated.

The result is that the non-controversial aspects of the agreement are recorded in the *CPA Template* before negotiation starts. This simplifies the *Negotiation Process* by removing from consideration all subjects that were resolved during the composition process. The *Negotiation Process* operates on a smaller set of items and will converge rapidly. In addition, the process of composing the new *NDD* will uncover any incompatibilities between the *Parties* before the start of the *Negotiation Process*. The two *Parties* can either resolve those incompatibilities by human to human contact or conclude that no resolution is possible, without having first to go through a fruitless *Negotiation Process*.

6.3 CPA Template composition

Composition of a *CPA Template* is the same as composing any *CPA* from two *CPPs*. Appendix E, "*CPA* Composition (Non-Normative)", of [ebCPP] contains a detailed discussion of *CPA* composition from two *CPPs*.

7 CPP and CPA Template Content

This section discusses content of the *CPP* and *CPA Template* from the viewpoint of negotiability.

7.1 Validation of CPP and CPA Template

The rules discussed below ensure that the negotiable *CPP* or *CPA Template* can be validated by an XML parser while not appearing to constrain negotiability.

In general, since the negotiability details are provided in the *NDD*, it SHOULD be acceptable to include any valid arbitrary value or choice for a negotiable item in the pre-negotiation *CPP* or *CPA Template*. In other words, the *NDD* overrides what is in the pre-negotiation *CPP* or *CPA Template* for all negotiable items.

- Numerical values: Any valid value can be stated for a negotiable item in the pre-negotiation *CPP* or *CPA Template*.
- Cardinality: All acceptable choices that are to be negotiated MUST appear in the pre-negotiation *CPP* or *CPA Template*.

7.2 Preference Order

Enumerations MUST always be stated in preference order (highest preference first). In most cases, preference order is REQUIRED by the CPPA specification[ebCPP]. Following are examples:

- ***PartyId*** elements under the same ***PartyInfo*** element.
- CanSend and CanReceive elements under the ServiceBinding element (NEED TO VERIFY THIS)
- AccessAuthentication elements under the same TransportSender element
- EncryptionAlgorithm elements under the same TransportClientSecurity or TransportServerSecurity element.
- ***TransportProtocol*** elements under the same ***Transport*** element
- ***AnchorCertificate*** elements under the same ***Certificate*** element

7.3 Conflicts between two Parties' Preferences

When composing a *CPA Template* from its and another *Party's CPP*, a *Party* might encounter unresolvable conflicts. For example, *Party* 1 might allow alternative elements X and Y with a preference for X while *Party* 2 might allow elements X and Y with a preference for Y. In cases like these, the choice can be left open in the *CPA Template* and negotiated later.

7.4 CPA Period of Validity

The values of the ***Start*** and ***End*** elements in the *CPA Template* SHOULD be consistent with each other (start time MUST precede end time) and SHOULD be consistent with the expiration times of all the certificates. It is preferable that the *CPA* expire before any of its certificates expire. All of these times are negotiable but it will simplify matters if the times in the *CPA Template* are mutually consistent. If the ***Start*** and ***End*** elements do not appear in the *CPPs*; they MUST be added when the C*PA Template* is composed from the *CPPs*.

8 Negotiation CPA (NCPA)

CHANGES MAY BE NEEDED TO THE EXISTING TEXT IN THIS SECTION AND TO THE NCPA INSTANCE DOCUMENT TO HARMONIZE IT WITH THE LATEST VERSIONS OF THE NEGOTIATION MESSAGS AND BPSS INSTANCE DOCUMENT.

THIS SECTION NEEDS TO BE REVIEWED IN DETAIL TO SEE WHAT ADDITIONAL MATERIAL IS NEEDED.

The purpose of this section is to:

- Explain how to construct the *Negotiation CPA* such that it does not have to be negotiated;
- Explain the negotiation aspects of the *NCPA*. Principally, these aspects are the elements that define the interface between a *CPA* and the BPSS instance *Document*, i.e., the ***CollaborationRole***, ***ProcessSpecification***, and ***Role*** elements.

In general, an *NCPA* SHOULD be the simplest possible *CPA* that conforms to the [ebCPP] schema. With the possible exception of selection of a negotiation BPSS instance *Document* and *Party*-specific information such as *Party* name, *Party* ID**,** and endpoint address, it SHOULD be possible for any pair of *Parties* to use it.

Message exchanges are asynchronous. The ***ResponseToURL*** element in the *Negotiation Message* provides the URL for a response to the message.

The *NCPA* defines the interactions between two *Parties* that are negotiating the contents of a *CPA*. It identifies the BPSS instance *Document* that defines the negotiation choreography. An example of an *NCPA* is in Appendix C.

The following are minimalist requirements on the contents of the *NCPA* that help avoid the need to negotiate the *NCPA*. Depending on the particular function, negotiation can be avoided either by mandating choices or values in this specification or by mandating that a function with cardinality that includes zero be omitted.

Deleted: *negotiation*

8.1 Document Exchange

The following rules eliminate the need for negotiating the *Document*-exchange specifications for the *NCPA*:

- Omit the following child elements of the ***ebXMLSenderBinding*** and ***ebXMLReceiverBinding*** elements: ***ReliableMessaging***, ***PersistDuration***, ***xxxNonRepudiation***, and ***xxxDigitalEnvelope***. This means that reliable *Messaging* and *Message* security are not used.
- In the ***MessagingCharacteristics*** elements, specify the value "never" for the attributes ***ackRequested***, ***ackSignatureRequested***, and ***duplicateElimination*** (they are used only with reliable Messaging). For the ***actor*** attribute, specify either of the permitted values; this attribute is ignored when the value of the ***ackRequested*** attribute is "never".

NOTE: the negotiation subteam plans to define the following capability: *Messaging* could be specified to use basic SOAP or W3C XML Protocol (when available). In this context, “basic” means that values or choices that normally have to be negotiated will either be omitted or will be given fixed values by this specification.

8.2 Transport

- Use HTTP PUT or POST to send a proposed *CPA* to a URL.
- The response to an offer or counter offer is always synchronous. This avoids the need for the responder to know the URL for a response.

For more details, see the [ebMS] appendices that discuss synchronous exchanges.

Deleted: S

Deleted: B that discuss synchronous exchanges

8.3 Packaging

COMPLETION OF THE PACKAGING DEFINITION (E.G. SIMPLEPART DEFINITIONS) AWAITED COMPLETION OF THE NDD AND NEGOTIATION MESSAGE SCHEMAS. THE PACKAGING DEFINITION SHOULD BE COMPLETED NOW IN CONFORMANCE WITH THE CURENT NDD AND NEGOTIATION MESSAGE SCHEMAS.

Deleted: *S*

8.4 Security

Security of the negotiation protocol is not defined in this version of the specification. Negotiating *Parties* can use SSL with basic authentication to obtain a degree of security.

8.5 Explanation of NCPA Example

The instance document for the *NCPA* example is in Appendix C.

Deleted: text of

THE CURRENT NCPA EXAMPLE INSTANCE DOCUMENT IS UNCHANGED FROM THE JANUARY, 2003 DRAFT. IT MAY HAVE TO BE CHANGED TO CONFORM TO THE BPSS INSTANCE DOCUMENT IN THIS DRAFT, ESPECIALLY WITH REGARD TO THE CHANGES THAT HAVE BEEN MADE TO THE BPSS INSTANCE DOCUMENT IN THE AREA OF CONCLUSION OF NEGOTIATION.

EXPLANATORY TEXT IS NEEDED HERE.

9 Negotiability of CPA Elements and Attributes

This section discusses the negotiability of the different elements and attributes in the *CPA* and is concerned mostly with composing a *CPA* from two *CPPs*. It focuses on those cases that involve special considerations.

9.1 Enumerations

There are several cases of enumerations:

- Some enumerations are laid out in the *CPP* instance *Documents* (e.g. certificates).
- Some enumerations are laid out in the CPPA schema itself.
- Some enumerations might be defined only in the text of the CPPA specification and would have to be put into the *NDD* schema.
- Some enumerations are not listed in full anywhere (e.g. the W3C forms of encryption algorithm name)
- Some might be defined elsewhere, perhaps as a set of URIs.

In some cases, especially those that are defined in the CPPA schema, only the items in an enumeration that are acceptable to the *Party* that is preparing the *NDD* instance *Document* have to be listed in the *NDD*. An example is the versions of the specification that are acceptable to the *Party*.

The CPPA schema itself is input to the *Negotiation Process*. Therefore, enumerations that are defined in full in the CPPA schema don't necessarily have to be defined in full in the *NDD* schema.

9.2 CollaborationRole element and its child elements

The normal case is that the two *CPPs* are being composed into a *CPA Template* specify the same BPSS instance *Document*. In version 1 of this specification, the contents of the BPSS instance *Document* cannot be negotiated using the negotiation functions defined in this specification. Two prospective trading partners SHOULD agree on the same BPSS instance *Document* and assignment of *Roles* before beginning to negotiate the *CPA*. In many cases, agreement will be established by the fact that the two prospective trading partners have compatible ***CollaborationRole*** subtrees in their *CPPs*. The following considerations relate to establishing compatible ***CollaborationRole*** subtrees.

- If both *CPPs* specify the same *Role* (e.g. both specify "buyer"), the situation cannot be resolved automatically. Human contact is needed and one *CPP* MUST be changed to specify the other *Role*.
- If both *CPPs* specify both *Roles* (i.e. two ***CollaborationRole*** elements with opposite *Roles*), this cannot be resolved automatically. Human contact is needed and the two *Parties* MUST agree on which *Party* plays which *Role*.
- If *CPP* A specifies one *Role* and *CPP* B specifies both *Roles*, chose the *Role* in *CPP* B which is opposite to the *Role* specified in *CPP* A.
- If both *CPPs* specify more than one BPSS instance *Document* but there is only one in common to the two *Parties*, use that one.
- If both *CPPs* specify more than one BPSS instance *Document* that is in common to both of

them, human contact is needed to decide whether all the common ones are to be used in the *Business Collaboration* or which one is to be used.

From the viewpoint of *CPA* composition and negotiation, the best practice is to include only one BPSS instance *Document* in each *CPP*.

> NOTE: A *Party* can describe the *Business Collaboration* using any desired alternative to the ebXML *Business Process Specification Schema*. When an alternative *Business-Collaboration* description is used, the *Parties* to a *CPA* MUST agree on how to interpret the *Business-Collaboration* description and how to interpret the elements in the *CPA* that reference information in the *Business-Collaboration* description. The affected elements in the *CPA* are the ***Role*** element, the ***CanSend*** and ***CanReceive*** elements, the ***ActionContext*** element, and some attributes of the ***BusinessTransactionCharacteristics*** element. The two *Parties* also have to come to a common understanding of how to negotiate the negotiable elements and attributes whose interpretations are changed by the use of the alternative *Business Collaboration* description.

9.3 Elements or Attributes whose Cardinality Includes Zero

Regarding elements or attributes whose cardinalities include zero (omission), the main negotiable thing is "presence or absence". However, if it is agreed to include (one or more of) that element or attribute, it is then necessary to negotiate the value (or child elements in the case of an element) of each one that is included. ***PersistDuration*** is an example. If the two parties agree to include it, they then have to negotiate its value.

9.4 Values

For negotiating values, the negotiation depends on the type of value. It could be a range of values, a step size, members of an enumeration, etc. The type information is in the CPPA schema and might not have to be repeated in the *NDD*.

9.5 Items that are Referred to

If some element, ***A***, refers to another element, ***B***, by means of an IDREF and element ***A*** is negotiated, then there is also a need to negotiate what element ***A*** refers to (element ***B***). Element ***B*** must be inserted into the *CPA*-under-construction. For example, an anchor certificate reference points to a trust anchor that must be added to the *CPA.*

9.6 Transport Endpoints

Any *Party* can define whatever endpoints it chooses. There might be issues of matching endpoint characteristics. One example is the endpoint type. *Parties* might need to negotiate what endpoint types are used.

Deleted: Transport endpoints are not really negotiable since

9.7 Security

Negotiation of security agreements for collaboration can involve decisions to add or forego particular security features. For example, it might not be feasible to configure software for PKI based authentication, and the decision could be made to use weaker forms of basic (username and password) authentication combined with a transport data confidentiality option, such as SSL-3.

NDDs might indicate a willingness to change security features at the higher ***BusinessTransactionCharacteristics*** level as well as details related to the type of security to be used (its strength, allowed algorithms, and so forth.)

Finally, it might simply not be feasible to arrange for implementing certain security features that are regarded as essential for the collaboration contract. Until compatible software is obtained, or other security contracts are made, negotiation could fail to arrive at a mutually acceptable *CPA*.

It should be noted that negotiation on certificates might require human input. A Party's unwillingness to handle the proposed trust model is a reason for failure of the negotiation.

9.7.1 Trust Anchor and Certificate Alignment

This section discusses the kinds of negotiation that might take place for aligning ***SecurityDetails*** and ***TrustAnchors*** with various ***Certificates***.

The need for "aligning" ***TrustAnchors*** with various ***Certificates*** arises from the validation process for certificates. In this process, a certificate is examined to see whether the association of name and the distributed public key is to be trusted in one of its particular uses, such as for checking a digital signature or for encrypting a symmetric key as part of key exchange. One aspect of this validation process checks a certificate to see whether its issuer (or iteratively, an issuer of its issuer) is itself trusted as vouching for the certificate being a public key of the named entity. It does this by checking the signature on the subject certificate against the issuer's certificate. The certificates of the trusted issuers are called trust anchors. (For a self-signed certificate, where the issuer and subject are the same, the trust anchor certificate is also the distributed certificate.) (Other aspects of validity tests involve seeing whether certificates have expired or been revoked; these tasks are not discussed here.)

For collaborators to align ***TrustAnchors*** with ***Certificates,*** either one party MUST obtain a certificate that will be validated by the other party's ***TrustAnchors,*** or one party MUST update its ***TrustAnchors*** so that the other party's certificate is validated.

There are 3 major levels for alignments in public-key infrastructure (PKI).

1. Transport-level security
2. *Messaging*-level security (digital envelope and digital signature)
3. Application-level security

For transport-level security, (transient) encryption and authentication alignment are needed. Both server-side and client-side SSL or TLS need to have the trust anchors synchronized with corresponding certificates.

For messaging-level (persistent) security, both digital envelopes for data confidentiality and digital signatures for non-repudiation (of origin and/or receipt) also require alignment.

Deleted: <#>**Trust Anchors and Related Matters¶**
This section discusses the kinds of negotiation that might take place for aligning ***SecurityDetails*** and ***TrustAnchors*** with various ***CertificateRefs***.¶

Inserted: <#>**Trust Anchors and Related Matters¶**
This section discusses the kinds of negotiation that might take place for aligning ***SecurityDetails*** and ***TrustAnchors*** with various ***CertificateRefs***.¶
Security¶

Deleted: Security

For application-level (persistent) security, digital envelopes and non-repudiation (of origin and/or receipt) by means of digital signatures require alignment.

There are two main cases to be dealt with:

1. Add a certificate authority (CA) as a trust anchor.
2. Allow direct trust using a self-signed certificate. The self-signed certificate must be referenced as a trust anchor.

In general, if the proposer of the draft *CPA* seeks to request that the other party acquire a new certificate, the draft *CPA* may contain a placeholder for the certificate, and the *NDD* will indicate a need to supply a certificate. If the proposer seeks to request that the other party change its ***TrustAnchors,*** the proposer should make this addition to the other party's ***TrustAnchors,*** and then mark the item as negotiable (in case, a different issuer certificate somewhere in a chain of issuers is to be added as a trust anchor instead of the proposed issuer certificate.)

It is worth explicitly discussing the case where a certificate can be a self-signed certificate. The draft *CPA* proposer might add a self-signed certificate to the othe party's ***TrustAnchors***/***AnchorCertificateRef*** list. If the self-signed certificate was found in the *CPP* of the other party, the proposer might also update its own ***TrustAnchors*** so that the other party's self-signed certificate is referenced. These proposals can be viewed as adopting a direct trust model, rather than a hierarchical model involving certificate authorities.

Finally, when aligning ***TrustAnchors*** with the self-signed certificate fails, the security function resting on this PKI alignment might be changed. In this last case, the negotiation option might involve a change in the value of an attribute under ***BusinessTransactionCharacteristics***, and so would need to be reviewed against any contracts in effect. This review would presumably not be an automated process in the near term.

As a result of the *CPA Template* formation process, various details could be up for negotiation.

First, a change to the PKI might be proposed. For the self-signed certificate addition option, the negotiatee might want to:

1. Reject adding a self-signed certificate and indicate rejection of the security function resting on this PKI alignment
2. Insist on the proposer getting a certificate from an existing CA.
3. Propose issuing another certificate signed by an acceptable authority.

For case 1, the negotiation "space" would involve a change in the value of an attribute under ***BusinessTransactionCharacteristics***.

For case 2, the negotiatee would have to indicate rejection of the *CPA Template* and indicate that until the *CPP* certificate value changes, there will be no forward progress. The proposer would have to go out and get a new certificate.

Deleted: For transport-level security, (transient) encryption and authentication alignment are needed. Both server-side and client-side SSL or TLS need to have the trust anchors synchronized with corresponding certificates.¶
¶
For *Messaging*-level (persistent) security, digital envelopes and non-repudiation (of origin and/or receipt) by means of digital signatures require alignment. ¶
¶
For application-level (persistent) security, digital envelopes and non-repudiation (of origin and/or receipt) by means of digital signatures require alignment. ¶

Deleted: Failure to validate a certificate need not prevent formation of a *CPA Template*. First, the sender's signing certificate can be a self-signed certificate. If so, a reference to this self-signed certificate can be added to the receiver's ***TrustAnchors*** and ***AnchorCertificateRef*** lists. This proposal amounts to proposing to agree to a direct trust model, rather than a hierarchical model involving certificate authorities. Second, a proposal to add a trusted root might be made, again by appropriate revision of the ***TrustAnchors*** element.¶

For case 3, the negotiatee would propose a different certificate issued by its own CA. The negotiatee would have to install it and use it for this transaction. This is not yet a common practice, though it is logically possible. This would involve one side being a CA for the *Business Process* and the ability of the other side to use more than one certificate for its existing key-pair. The *CPA* proposed to do this would go outside of anything strictly derivable from the *CPP* (only the old X.509 certificate would be used to put together a new X.509 certificate from a new issuer).

Next, for the PKI trust anchor certificate addition option, the negotiatee might want to:

1. Reject adding a new CA to its trust anchors and indicate rejection of the security function resting on this PKI alignment.
2. Insist on the proposer getting a certificate from some already trusted existing CA.
3. Propose accepting another certificate signed by its own signing authority.
4. Propose a different trust anchor either higher or lower in the validation chain than the one proposed by the other side.

Again, as for adding a self-signed certificate, for case 1, the negotiation "space" would involve a change in the value of an attribute under the ***BusinessTransactionCharacteristics*** element. For case 2, the response would have to be rejection with a call for a change in *CPP*. For case 3, the negotiatee proceeds as described in case 3 above.

The new case 4 is logically possible but still exotic. In effect, the negotiation SHOULD not matter to the other side, because it is just an adjustment to which trust anchor is added to one side's PKI trust list and the certificate used would still validate to the alternative trust anchor. Yet it would reflect a slight change in security details.

9.8 Discussion of Various Elements and Attributes

This section discusses some examples of negotiation of various elements and attributes. It is not intended to be an exhaustive discussion of everything in the *CPA*.

cpaid attribute: The value of the ***cpaid*** attribute can be negotiated. In order to negotiate the value of the ***cpaid*** attribute, it SHALL be a URI.

PartyInfo element: This element cannot be negotiated. There MUST be one in the *CPA* for each *Party*. Various attributes and child elements can be negotiated.

Start and ***End*** elements: The value of the ***Start*** element MUST precede the value of the ***End*** element and the times stated in the ***Start*** and ***End*** elements MUST NOT be outside the certificate validity periods. If the values of the ***Start*** and ***End*** elements are negotiable, the *CPP* SHALL specify the earliest acceptable start time and the latest acceptable end time.

Status element: The ***Status*** element is not negotiable; its value identifies the state of the negotiation. The negotiation algorithm is responsible for changing the state at appropriate times.

ConversationConstraints element: First the *Parties* MUST agree on whether this element is to be used. Then, they MUST agree on the values of its ***invocationLimit*** and ***concurrentConversations*** attributes.

defaultMSHChannelId: Since a delivery channel contains both *Parties'* properties, the two *Parties* have to agree on both *Parties'* default delivery channels.

PartyInfo element: The presence of the ***PartyInfo*** element is not negotiable; there MUST be one for each *Party*. The contents of the ***PartyInfo*** element are negotiable.

> NOTE: The case where one or both *CPPs* contains more than one ***PartyInfo*** element will be discussed in a future version of this specification.

PartyId type: The ***type*** attribute of the ***PartyId*** element identifies the naming system to which the ***PartyId*** belongs (e.g. DUNS). The *Negotiation Process* SHOULD select one possible ***PartyId*** type for each *Party* and eliminate any others that are in the *CPPs*. Each *Party's* ***PartyId*** type MUST be understandable by the other *Party*. Eliminating the others ensures that each *Party* will always use the same ***PartyId*** for the other *Party*.

PartyRef: One reason to negotiate is that a *Party* might not be able to understand the other *Party's* ***PartyRef*** *Document*. For example, the geographical contexts might not match. While negotiating the contents of the ***PartyRef*** *Document* is out of scope for this specification, negotiating the contents might lead to negotiating the schema (type), which is in scope.

CollaborationRole: the cardinality is one or more.

name attribute of the ***ProcessSpecification*** element: This is not negotiable unless a future version of [ebBPSS] provides for more than one ***ProcessSpecification*** element in a BPSS instance *Document*.

Role: The two *Parties* have to have opposite *Roles* in a *Business Collaboration*. This MUST be validated. There is no known use case for negotiating it.

ApplicationCertificateRef: This is negotiable because one party's certificate authority might not be acceptable to the other *Party*. The value of the ***certId*** attribute could be an enumeration of possible certificates. There can be zero or more ***ApplicationCertificateRef*** elements.

ThisPartyActionBinding: In general, each *Party* has to know the name that the other *Party* uses for each action but they don't need to negotiate since there is no reason for the names to match.

PackageId might be negotiable.

ActionContext: This is not negotiable. If BPSS is not being used, ignore the ***ActionContext*** element.

CollaborationActivity: This allows a ***Party*** to specify a complete path inside the BPSS instance

Deleted: possible

Deleted: ***version*** attribute of the ***ProcessSpecification*** element: The two *Parties' CPPs* might specify the same BPSS instance *Document* but different versions of it.¶

¶

THE VERSION ATTRIBUTE OF THE BPSS PROCESSSPECIFICATION ELEMENT IS ACTUALLY THE VERSION OF THE SPECIFICATION. THE BPSS SPECIFICATION DOES NOT DEFINE A VERSION ATTRIBUTE OF A BPSS INSTANCE DOCUMENT. DECISIONS HAVE TO BE MADE ON HOW TO RESOLVE THE DISCREPANCY BETWEEN [EBCPP] AND [EBBPSS] AND THEN TO REWRITE REFERENCES TO THE VERSION ATTRIBUTE IN THIS SPECIFICATION. ONE SOLUTION IS TO CORRECT [EBCPP] TO REFER TO THE VERSION OF [EBBPSS] AND CHANGE THIS SPECIFICATION ACCORDINGLY. ANOTHER IS TO DEFINE A BPSS INSTANCE DOCUMENT VERSION AND CORRECT BOTH [EBCPP] AND THIS SPECIFICATION TO AGREE WITH THE BPSS SPECIFICATION. THE LATER SOLUTION IS FOR A FUTURE VERSION OF ALL THREE SPECIFICATIONS.¶

¶

Deleted: ds:***Reference*** child of ***ProcessSpecification*** element: ***IT IS TO BE DETERMINED WHETHER BOTH PARTIES MUST HAVE DS:REFERENCE IF EITHER HAS IT. IT HAS BEEN SUGGESTED THAT THIS IS NECESSARY SO THAT IF EITHER PARTY VALIDATES THE BPSS INSTANCE DOCUMENT USING DS:REFERENCE, BOTH PARTIES SHOULD VALIDATE.¶***

¶

Document. Its value is completely determined by the structure of the BPSS instance *Document* and is therefore not negotiable.

channelId: The *Parties* can negotiate which delivery channels to use or add new ones.

Certificate: An enumeration of ***keyinfo*** types might be useful to help decide which certificates are acceptable.

DeliveryChannel: Cardinality is negotiable. It is suggested that a new delivery channel be created rather than modifying an existing one.

Signing the CPA: Negotiation of signing is accomplished by negotiating presence of the *CPA* ***Signature*** element and its child ***ds: Signature*** elements. See Section 12.12 for details.

Comment: [ebCPP] states that all comments in both *CPPs* SHALL be included in the *CPA* unless the *Parties* agree otherwise. Therefore, each ***Comment*** element is separately negotiable. Since comments are arbitrary text strings, negotiation about ***Comment*** elements MUST be by human to human contact.

10 Negotiation Descriptor Document

NEELAKANTAN KARTHA POINTED OUT THAT THE FOLLOWING ANALYSIS IS NEEDED:

- ***CHECK WHETHER THERE IS SUFFICIENT SUPPORT IN THE NDD AND NEGOTIATION MESSAGES TO DO WHAT IS NEEDED TO SUPPORT THE PKI INFRASTRUCTURE, SUCH AS ADDING A NEW CERTIFICATE AUTHORITY.***
- ***MAKE SURE THAT THE MESSAGES SUPPORT ALL THE INTERACTIONS SUGGESTED BY THE NDD. A WAY TO DO THIS IS TO TAKE THE SAMPLE NDD INSTANCE DOCUMENT AND VERIFY THAT ALL THE INTERACTIONS SUGGESTED IN IT CAN BE CARRIED OUT USING THE MESSAGES.***

Formatted: Bullets and Numbering

The *Negotiation Descriptor Document* (*NDD*) describes what is negotiable in the accompanying *CPP* or *CPA Template*. It SHALL describe only the negotiable elements and attributes and SHALL omit those elements and attributes that are not negotiable.

The *NDD* identifies the *CPP* or *CPA Template*. The *CPP* or *CPA Template* does not identify the *NDD* since a *Party* might have many different *NDDs* associated with the same *CPP* or *CPA Template*. These could be for different *Negotiation Processes*, different categories of partner, etc.

10.1 Use of NDD

- An *NDD* can be placed in a *Registry* along with the *CPP*. The *NDD* and *CPP* would have to be connected by *Registry* metadata. Alternatively, a *Party* might choose not to include an *NDD* in the *Registry*. Instead, when a *Party* is discovered by a prospective trading partner, the *NDDs* can be exchanged prior to the opening step of the negotiation. This permits a *Party* to send an *NDD* that it considers appropriate for the particular prospective trading partner.
- An *NDD* is sent from the *Party* making the initial offer to the other *Party* during initialization of the *Negotiation Protocol*. After that, the *NDD* is not modified during negotiation and is not again sent from one *Party* to the other. All information about the state of negotiation of the negotiable items is exchanged in the *Negotiation Messages*.

  NOTE: This means that an item which is initially not negotiable cannot be made negotiable during the *Negotiation Protocol*.

10.2 General Principles of Contents of NDD

The *NDD* has been defined in an abstract manner to enable it to be applied to any kind of XML agreement. This avoids the need to define a new *NDD* schema for each kind of *Document* to be negotiated.

> NOTE: The abstract level of the *NDD* is an opportunity for tool vendors to produce *NDD* composition tools. Such a tool would have a GUI that would tailor the view of the *NDD* to the specific kind of *Document* to be negotiated. The tool would reference the schema of the *Document* being negotiated along with the *NDD* being constructed. This will supply the tool

with sufficient information to make the views understandable by someone who is composing an *NDD*. This would enable that person to communicate with the tool in terms of the specifics of the *Document* to be negotiated. The tool could then construct the *NDD* instance *Document* in accord with the *NDD* schema.

The *NDD* references both the *CPA Template* and the *CPPA* XML schema.

The *NDD* consists of a variable length (cardinality 1 or more) set of [XPATH] expressions, each of which refers to a negotiable element or attribute.

With each XPATH expression, the negotiability of the element or attribute is defined by child elements. These child elements represent the negotiability characteristics of the element or attribute identified by the XPATH statement. Examples are:

- For a numeric value, minimum, maximum, and negotiation step size
- For choices, XPATH statements, ID attribute values, qnames, element values, etc. which identify the specific choices within the *Document* being negotiated. Examples in the *CPA* are certificates, delivery channels, transport protocols, and signature algorithms.

Deleted: <#>Cardinality (range of permitted cardinalities)¶

The following rules define what is negotiable at the point referenced by an XPATH expression:

1. If the XPATH expression references a non-leaf element, that element, and the whole sub tree below that element, are negotiable.
2. If the XPATH expression references any attribute, it means that only that attribute is negotiable and doesn't imply anything about the containing element or the rest of the sub tree descended from the element containing that attribute.
3. If the XPATH expression references a leaf element, only that element and its contained attributes are negotiable.

10.3 Composition of an NDD for a CPA Template

Formally, the negotiation defined in this specification begins when an offering *Party*, *Party* A, presents an initial offer, consisting of a *CPA Template* and the corresponding *NDD*, to *Party* B. If *Party* A cannot obtain *Party* B's *NDD*, *Party* A can offer its own *NDD*. If *Party* A can obtain *Party* B's *NDD*, *Party* A SHALL make use of *Party* B's *NDD* in composing the *NDD* for the initial offer to avoid situations in which the negotiation is certain to fail. The new *NDD* would be a composite of the two sets of requirements that is acceptable to both *Parties* as a starting point in negotiation. In the rest of this section, the normative statements are normative only if the offering *Party* has obtained the other *Party's NDD*.

Deleted: one

Deleted: its

Deleted: another

Deleted: However, the following RECOMMENDATIONS on constructing an *NDD* for a *CPA Template* might facilitate the *Negotiation Process*.¶

Deleted: ¶
If the initiating *Party* has access to the other *Party's NDD* that goes with its *CPP*, the initiating *Party* SHOULD use both its and the other *Party's NDD* to establish the *NDD* and *CPA Template* to be used as the initial offer.

Party A's taking *Party* B's *NDD* into account will speed up convergence as well as recognition of fatal incompatibilities and reduce the possibilities of unnecessary rejects during negotiation. In other words, composing a *CPA Template* and combined *NDD* before starting negotiation simplifies the *Negotiation Process* by:

1. Removing subjects from negotiation that can be handled by simple matching.
2. Quickly recognizing the existence of fatal incompatibilities. For fatal incompatibilities, human to human contact to resolve the incompatibilities is RECOMMENDED.

In composing the *NDD* of the *CPA Template*, *Party* A SHALL exclude from the new *NDD* anything that *Party* A understands (from *Party* B's *NDD*) is not negotiable or is unacceptable to *Party* B. For example, for an enumeration, the new *NDD* SHALL include only those choices that are common to both of the original *NDDs*. For a range of values, *Party* A SHALL put in the new *NDD* only the common range. If, for some element, *Party* A had specified values of 1-9 and Party B had specified values of 3-12, the new *NDD* SHALL specify values 3-9. The intersection process might identify items with no common ground, making successful negotiation unlikely.

Deleted: f *Party* A is

Deleted: SHOULD

Deleted: SHOULD

Deleted: SHOULD

Deleted: SH

Deleted: OULD

Party A SHALL NOT include items in the new *NDD* that were not in *Party* B's original *NDD* because *Party* B did not intend to negotiate on the items that it did not put in its original *NDD*. For those items that were not in *Party* B's initial *NDD*, *Party* A MUST either accept what is in *Party* B's *CPP* or MUST recognize that there is an irreconcilable conflict.

Deleted: SH

Deleted: OULD

10.4 Explanation of Contents of NDD

This section discusses the schema and example of an *NDD* instance *Document*. See Appendix A for the schema and Appendix F for an example of an instance *Document*.

The ***NegotiationDescriptor*** element is the top element of an *NDD*. It is is a container element, that contains one or more instances of the negotiable parts called ***NegotiableInformationItem*** elements. The ***documentLocation*** attribute of the ***NegotiationDescriptor*** element is a URI that points to the XML *Document* to which this *NDD Document* corresponds. For instance, if the *NDD* pertains to a *CPA*, the ***documentLocation*** attribute points to that *CPA*.

Each ***NegotiableInformationItem*** element contains an ***xpath*** attribute that identifies the negotiable information item with respect to the *Document* pointed to by the ***documentLocation*** attribute of the ***NegotiationDescriptor*** element.

Each negotiable information item (which could be an XML element or an attribute) is one of the following types, depending on what kind of negotiation that one needs to perform on this negotiable information item.

1. Value: For negotiating the value of the item, if the value of the item needs to be just filled in by one party (instead of negotiated), the attribute ***mustBeFilledIn*** can be used to specify this possibility.
2. UnorderedValue: For negotiating the presence or absence of a member of a set of unordered values.
3. OrderedValue: For negotiating to choose among the members of a set of ordered values, where the preference is of a simple kind (namely, a preference for earlier values or later values in the set).
4. ValuesWithPreferenceMeasure: For negotiating to choose among the members of a set of values, where the preference measure is of a more complicated nature. ; For instance, it is often possible to express the preference measure by a piecewise linear function. The preference measure is approximated by a piecewise linear function, which then is specified by defining each piece. Each piece is defined by giving the two (x,y) coordinates that define

Deleted: (for instance, expressed by a piecewise linear function or a function defined by an equation).

the start and end of the piece; it is assumed that the (x,y) coordinates that define the end of one piece are identical to the (x,y) coordinates that define the start of the next piece. The y-axis has an arbitrary scale (say 1-100) that gives the range of preferences.

5. PresentOrNot: For negotiating the presence or absence of a value. This type allows one to express that a *Party* (a) insists that a value MUST be present; (b) insists that a value is absent; (c) is ok with the value being present or absent, but has a preference for one or the other or (d) is o.k with the value being present or absent, and has no preference.
6. IntegerValues: For expressing (a) whether an integer value is present or not (as in PresentOrNot) and then (b) the choice between different integer values using simple preference measures (such as smaller ones being preferred or larger ones being preferred) or more complicated preference measures (such as those expressible via piecewise linear functions). This type is provided mostly for convenience, since there are many entries in a *CPP* or *CPA* that impose these kinds of negotiation requirements.
7. Preference: For expressing preference among values of a similar nature (such as multiple elements at the same level, e.g., the ***PartyInfo*** element)
8. Cardinality: This is for expressing preferences among different cardinalities using simple preference measures (such as smaller cardinalities being preferred or larger cardinalities being preferred) or more complicated preference measures (such as those expressible via piecewise linear functions).
9. BooleanValues: For expressing (a) whether a Boolean-valued item is present or not and then (b) for expressing preference for either true or false as the value of the boolean-valued item.
10. DurationWithPreference: For expressing (a) whether a duration-valued item is present and then (b) to give maximum and minimum possible values of the duration and to express a preference for smaller values or larger values.

Deleted: Similar to IntegerValues.

For more details, comments and examples of using each of these types, the reader is directed to the *NDD* schema (Appendix A) and instance *Document* (Appendix F).

11 Negotiation Messages

SEE THE NOTE IN THIS TYPE FACE AT THE BEGINNING OF THE NDD SECTION.

A *Negotiation Message* includes the details of a offer or a counter offer, identification of the *NDD* and *CPA Template* being negotiated, and information that controls the *Negotiation Protocol*. Some *Messages* include the *NDD* and the *CPA Template* or their URLs.

This section defines and discusses the details in the *Negotiation Message* in terms of the individual XML elements and attributes. The discussion is illustrated with XML fragments.

See Appendix B for the complete *Negotiation-Message* XML Schema. See Appendix G for examples of *Negotiation-Message* instance *Documents*.

THIS SECTION HAS BEEN COMPLETELY REPLACED.

11.1 Negotiation Message Structure

This section discusses the overall structure of the *Negotiation Message*. Subsequent sections discuss each of the elements in more detail.

```
<NegotiationMessage
        xmlns:tp="http://www.oasis-open.org/committees/ebxml-cppa/schema/cpa-negot-1_0.xsd"
        xmlns:xsi=http://www.w3.org/2001/XMLSchema-instance xmlns:cppa="http://www.oasis-
        open.org/committees/ebxml-cppa/schema/cpp-cpa-2_0.xsd"
        xsi:schemaLocation="http://www.oasis-open.org/committees/ebxml-cppa/schema/cpa-negot-
        1_0.xsd"
        businessMsgId="busMsg002"
        binding="false"
        inresponseTo="busMsg001"
        negotiationDialogId="negotDialog001"
        offerId="offer001"
        messageType="CounterOffer"
        error=”ExpiredCPP”>
 <NCPA uri="http://..." />
 <CPAIdentity>
        …
 </CPAIdentity>
 <cppa:SecurityDetails cppa:securityId="ID">
        …
 </cppa:SecurityDetails>
 <InitiatingParty>
        …
 </InitiatingParty>
 <RespondingParty>
        …
 </RespondingParty>
 <BPSSBusinessDocumentName name="CPA_Counter_Offer_Doc" />
 <ExpirationDate>…</ExpirationDate>
 <BusinessDocuments>
        …
 </BusinessDocuments>
```

```
  <NegotiationContent>
        …
  </NegotiationContent>
  <ResponseToURL>
  …
  </ResponseToURL>
  <Offer-CounterOffer-Acceptance-Time>
  </Offer-CounterOffer-Acceptance-Time>
  <Comment/>
</NegotiationMessage>
```

11.1.1 NegotiationMessage element

The ***NegotiationMessage*** element is the root element of the *Negotiation Message* xml *Document*. The *Negotiation Message Document* contains the following REQUIRED[XML] Namespace[XMLNS] declarations:

- The default namespace: xmlns== "http://www.oasis-open.org/committees/ebxml-cppa/schema/cpa-negot-1_0.xsd"
- The schema instance namespace: xmlns:xsi=http://www.w3.org/2001/XMLSchema-instance
- The ebXML CPPA namespace: xmlns:cppa= http://www.oasis-open.org/committees/ebxml-cppa/schema/cpp-cpa-2_0.xsd

The ***NegotiationMessage*** element contains the following attributes:

- a REQUIRED ***businessMsgId*** attribute that uniquely identifies the current *Business Message* within the scope of one *Negotiation Dialog*,
- a REQUIRED ***negotiationDialogId*** attribute that uniquely identifies an ongoing *Negotiation Dialog* (See Section 12.1) that connects multiple offer/counter-offer transactions that pertain to the same *CPA Template*,
- an IMPLIED ***offerId*** attribute that uniquely identifies each instance of an offer or counter offer,
- an IMPLIED ***inresponseTo*** attribute that identifies the unique *Business Message* of the previous offer or counter-offer that this *Business Message* is responding to. Its value can be null for the initiating offer of the *Negotiation Dialog*,
- a REQUIRED ***binding*** attribute that indicates whether the current *Message* is legally binding,
- a REQUIRED ***messageType*** attribute that indicates the status of current negotiation. The legal values for the ***messageType*** attribute are:
    - ♦ “Offer”
    - ♦ “CounterOffer”,
    - ♦ “CounterOfferPending”
    - ♦ “Rejected”
    - ♦ “Accepted”
    - ♦ “Expired”
    - ♦ “SinglePartySigned”
    - ♦ “Signed”
    - ♦ “Unsigned”

- An IMPLIED **error** attribute that identifies the error code in case of rejection for a CounterOffer.

The ***NegotiationMessage*** element SHALL consist of the following child elements:

- One REQUIRED ***NCPA*** element to identify the current *NCPA*
- One REQUIRED ***CPATemplateId*** element to identify the *CPA* that is being negotiated.
- One REQUIRED ***cppa:SecurityDetails*** element to describe the security requirements of current negotiation.
- One REQUIRED ***InitiatingParty*** element to describe the initiating *Party* of the current *Negotiation Dialog*.
- One REQUIRED ***RespondingParty*** element to describe the responding *Party* of the current *Negotiation Dialog*.
- One REQUIRED ***BPSSBusinessDocumentName*** element that indicates the name of the *Business Document* this offer relates to.
- One REQUIRED ***ExpirationDate*** element that specifies the date when this offer or counter offer expires. Its XML data type is “dateTime”.
- One REQUIRED ***BusinessDocuments*** element that describes the *CPA Template*.
- One REQUIRED ***NegotiationContent*** element that itemizes accepted, rejected, updated elements within the current *CPA*.
- One IMPLIED ***ResponseToURL*** element that identifies the return (http) address where the responding *Party* can send its response asynchronously. See Section 11.1.10 for more information.
- One IMPLIED ***Comment*** element that can be used to record free text. For example, if the counter offer is a rejection, the *Party* can list its contact information here for a more traditional negotiation that will involve humans.

11.1.2 NCPA element

The ***NCPA*** element contains one REQUIRED ***uri*** attribute that SHALL have a value that is a URI that conforms to [RFC2396] and identifies the location of the *Negotiation CPA* xml instance *Document*.

11.1.3 CPATemplateId element

The ***CPATemplateId*** element contains a REQUIRED ***id*** attribute and a REQUIRED ***version*** attribute. The definitions of the ***id*** and ***version*** attributes are the same as the definitions of the ***cpaid*** and ***version*** attributes, respectively, of the [ebCPP] ***CollaborationProtocolAgreement*** element.

11.1.4 cppa:SecurityDetails element

The ***cppa:SecurityDetails*** element is defined in detail in the “***SecurityDetails*** element” section of [ebCPP].

11.1.5 InitiatingParty element

The ***InitiatingParty*** element describes the *Party* that initiated the current *Negotiation Dialog*. This element contains a REQUIRED ***cppa:PartyId*** element (per the definition in [ebCPP]), a REQUIRED ***CPPId*** element, and a ***CPPNDD*** element (cardinality 0 or 1) that identifies the *NDD* that is associated with the initiating *Party’s CPP*.

The ***CPPId*** element has two attributes:

- A REQUIRED ***id*** attribute, which SHALL contain the correct value as specified in [ebCPP] for the ***cppid*** attribute of the ***CollaborationProtocolProfile*** element in a *CPP* document.
- A REQUIRED ***version*** attribute. See [ebCPP] for the definition of the ***version*** attribute of the ***CollaborationProtocolProfile*** element.

The ***CPPNDD*** element contains either a child ***BinaryDoc*** element or a child ***Uri*** element. The ***BinaryDoc*** element has the xml type "base64Binary". Its value is the *NDD Document*. The ***Uri*** element SHALL have a value that is a URI that conforms to [RFC2396] and identifies the location of the *NDD* XML *Document*.

11.1.6 RespondingParty element

The ***RespondingParty*** element describes the *Party* that the initiating *Party* wishes to establish a *CPA* with. This element has the same structure as the ***InitiatingParty*** element.
The ***RespondingParty*** element contains a REQUIRED ***cppa:PartyId*** element, a REQUIRED ***CPPId*** element (per [ebCPP]), and a ***CPPNDD*** element (cardinality 0 or 1) that identifies the *NDD* that is associated with the *CPP*.

The ***CPPId*** element has two attributes:

- A REQUIRED ***id*** attribute that SHALL contain the value specified for the ***cppid*** attribute of the ***CollaborationProtocolProfile*** element in a *CPP* document. See [ebCPP].
- A REQUIRED ***version*** attribute. See [ebCPP] for the definition of the ***version*** attribute of the ***CollaborationProtocolProfile*** element.

The ***CPPNDD*** element contains either a child ***BinaryDoc*** element or a child ***Uri*** element. The ***BinaryDoc*** element has the xml type "base64Binary". Its value is the *NDD* Document. The ***Uri*** element SHALL have a value that is a URI that conforms to [RFC2396] and identifies the location of the *NDD* XML *Document*.

Deleted: xml

11.1.7 BPSSBusinessDocumentName element

The ***BPSSBusinessDocumentName*** element identifies the *Business Document* that is defined in the negotiation BPSS instance *Document.* The ***BPSSBusinessDocumentName*** element has one REQUIRED ***name*** attribute. The value of this attribute is the name of the *Business Document* and SHALL be one of the following:

- `“CPA_Offer_Doc”`
- `“CPA_Accept_Offer_Doc”`
- `“CPA_Counter_Pending_Offer_Doc”`
- `“CPA_Counter_Offer_Doc”`
- `“CPA_Reject_Offer_Doc”`
- `“CPA_Final_Doc”`
- `“CPA_Final_Response_Doc”`
- `“CPA_Final_Response_Doc_Signed”`
- “CPA_Final_Response_Reject_Doc”

11.1.8 BusinessDocuments element

The ***BusinessDocuments*** element has a ***CPATemplateDoc*** child element. The ***CPATemplateDoc*** element contains the following child elements:

- A REQUIRED ***NDD*** element that identifies the *NDD* associated with the *CPA Template* in the initial offer. It has either a child ***BinaryDoc*** element or a child ***Uri*** element. The ***BinaryDoc*** element has the xml type "base64Binary". Its value is the *NDD Document*. The ***Uri*** element SHALL have a value that is a URI that conforms to [RFC2396] and identifies the location of the *NDD* xml *Document*.
- A REQUIRED ***CPATemplate*** element that has either a ***BinaryDoc*** child element that has the type base64Binary and whose value is the proposed *CPA Template Document*, or a ***Uri*** element whose value is a URI that conforms to [RFC2396] and references the location of proposed *CPA Template* xml *Document*. The ***CPATemplate*** element is also used to send or reference the final *CPA* at the end of a *Negotiation Dialog.*

Throughout one negotiation process, the most up-to-date ***CPATemplate*** SHALL be available for the other *Party*. Therefore, each *Party* SHALL include either the updated *CPA Template* or a URI reference to it in each message. The receiving *Party* can reference it while examining the ***NegotiationContent*** element, which identifies the delta between the current copy of ***CPATemplate*** and the one that was sent out earlier. See Section 12.10 for a definition of "updated *CPA Template".*

11.1.9 NegotiationContent element

For an initial offer, the ***NegotiationContent*** element can be empty. If the offering *Party* had created the initial *CPA Template* and *NDD* by modifying information in the other *Party's CPP* and *NDD,* the ***NegotiationContent*** element in the initial offer SHALL describe the changes made by the offering *Party* to the information in the other *Party's CPP* or *CPA Template* when forming the *CPA Template* of the initial offer. For counter offers within this *Negotiation Dialog,* the ***NegotiationContent*** element SHALL describe the sending *Party's* proposed modifications to the *CPA Template.*

The ***NegotiationContent*** element SHALL list all items accepted by the sending *Party* since the start of the *Negotiation Dialog* (including the ones being accepted by this *Message*). The ***NegotiationContent*** element does not contain items accepted by the other *Party* since the start of the negotiation.

The ***NegotiationContent*** element SHALL list all items updated, deleted, or inserted by the sending *Party* since the sending *Party* received the previous offer or counter offer. Any item that has been deleted by one party can no longer be re-inserted in future counter offers.

It is up to the receiver of this *Message* to decide whether to continue negotiate, accept, or reject changes listed within the ***NegotiationContent*** element.

The ***NegotiationContent*** element contains the following child elements. In the following information, the Xpath of the item points to the item in the most recent update of the *CPA Template*, i.e. the one that is included in or referenced by this *Message.*

THE ELEMENT AND ATTRIBUTE NAMES SHOULD BE ADDED TO THE SECOND-LEVEL BULLETS. THE TERM "ELEMENT" OR "ATTRIBUTE" SHOULD BE ADDED AFTER EACH ELEMENT OR ATTRIBUTE NAME BELOW.

- ***AcceptedItem***[1](1…n): These are the items that have been accepted by the sending Party during all exchanges prior to this Message and within the same Negotiation Dialog.
  - Xpath of item
- ***DeletedItem*** (0…n)
  - Xpath of item
- ***UpdatedItem*** (0…n)
  - Xpath of item
  - Original value of item
  - Proposed value of item
  - The status that indicates update of this item is "Required" or "Preferred".
- ***InsertedItem*** (0…n)
  - Xpath of item
  - Proposed value of item
  - The status that indicates addition of this item is "Required" or "Preferred".

WE NEED TO EXPLAIN "REQUIRED" AND "PREFERRED".

One use case when ***InsertedItem*** will be useful is when the responding party wants to add in its own ***Certificate*** element under ebXML CPA/PartyInfo.

```
<CollaborationProtocolAgreement…>
 <PartyInfo…>
<!--  Certificates used by the "Seller" company   -->
   <tp:Certificate tp:certId="CompanyA_AppCert">
   <ds:KeyInfo>
   <ds:KeyName>CompanyA_AppCert_Key</ds:KeyName>
   </ds:KeyInfo>
  </tp:Certificate>
 <tp:Certificate tp:certId="CompanyA_SigningCert">
 <ds:KeyInfo>
 <ds:KeyName>CompanyA_SigningCert_Key</ds:KeyName>
 </ds:KeyInfo>
 </tp:Certificate>
  <!--  Certificates used by the "Buyer" company   -->
<tp:Certificate tp:certId="CompanyB_EncryptionCert">
 <ds:KeyInfo>
 <ds:KeyName>CompanyB_EncryptionCert_Key</ds:KeyName>
 </ds:KeyInfo>
 </tp:Certificate>
<tp:Certificate tp:certId="CompanyB_ServerCert">
 <ds:KeyInfo>
```

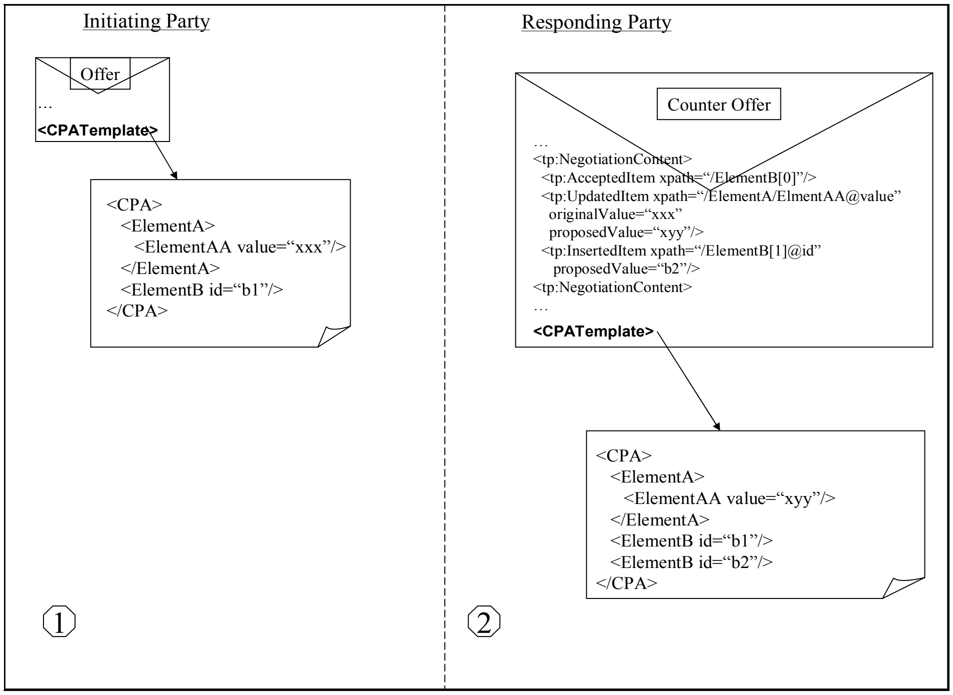
[1] An item can be either an element or an attribute.

```
 <ds:KeyName>CompanyB_ServerCert_Key</ds:KeyName>
 </ds:KeyInfo>
</tp:Certificate>
 :
 </PartyInfo…>
:
</CollaborationProtocolAgreement>
```

ADDITIONAL DISCUSSION OF INSERTIONS WOULD BE HELPFUL. EXAMPLES ARE:

- ***ONE CAN'T INSERT ANYTHING THAT ISN'T MENTIONED IN THE NDD.***
- ***INSERTION AND DELETION OF INSTANCES OF A REPEATING ELEMENT.***

Any of the items under the ***NegotiationContent*** element can be either a leaf node or non-leaf node. A non-leaf node indicates that the entire subtree under that node is subject to the corresponding change action. In other words, the entire subtree has been accepted, deleted, updated, or inserted though for update, not all items within the subtree have necessarily changed. See Figure 3 and Figure 4 for an example of how these elements can be used in a negotiation.



```
…
<CPATemplate>
```

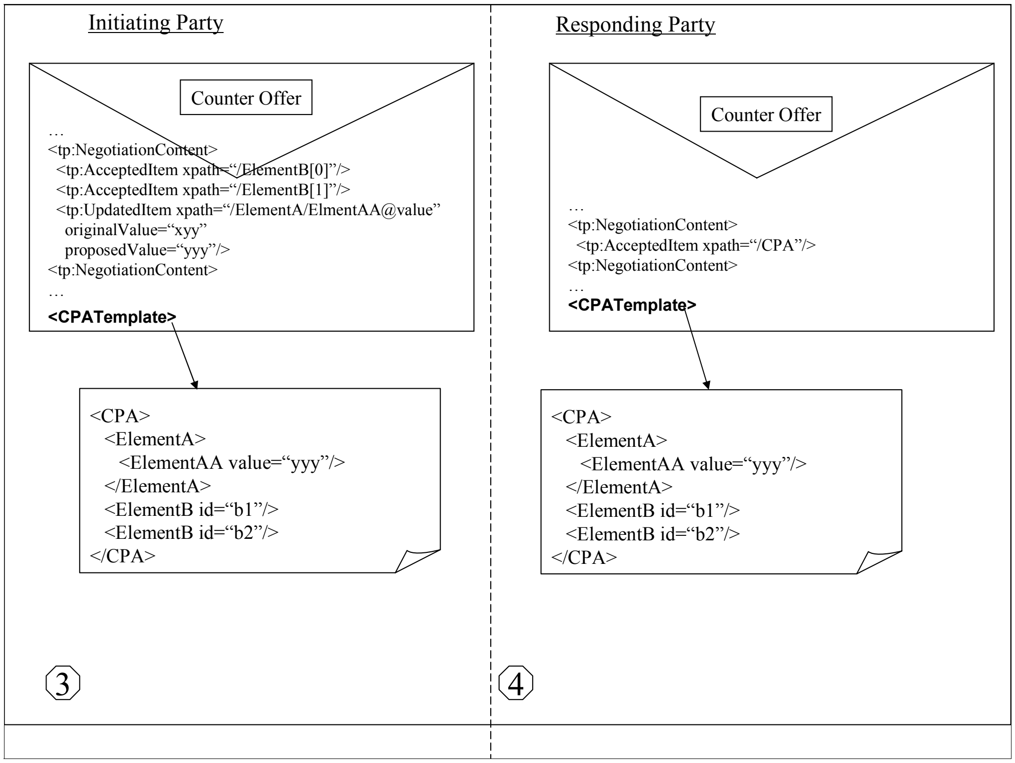
```
<CPA>
  <ElementA>
    <ElementAA value="xxx"/>
  </ElementA>
  <ElementB id="b1"/>
</CPA>
```

```
…
<tp:NegotiationContent>
  <tp:AcceptedItem xpath="/ElementB[0]"/>
  <tp:UpdatedItem xpath="/ElementA/ElmentAA@value"
    originalValue="xxx"
    proposedValue="xyy"/>
  <tp:InsertedItem xpath="/ElementB[1]@id"
     proposedValue="b2"/>
<tp:NegotiationContent>
…
<CPATemplate>
```

```
<CPA>
  <ElementA>
    <ElementAA value="xyy"/>
  </ElementA>
  <ElementB id="b1"/>
  <ElementB id="b2"/>
</CPA>
```

Figure 3. Example of how to use NegotiationContent element - Step 1

In Figure 3 , the initiating *Party* created a *CPA* that contains one element ***A*** and one element ***B***. The responding *Party* accepted element ***B***, updated the given element ***A*** value from 'xxx' to 'xyy', and inserted a second element, ***B***, which has the attribute, ***id***, whose proposed value is ‘b2’. The attached/referenced *CPA Template* now reflects the above changes.



```
…
<tp:NegotiationContent>
  <tp:AcceptedItem xpath=“/ElementB[0]”/>
  <tp:AcceptedItem xpath=“/ElementB[1]”/>
  <tp:UpdatedItem xpath=“/ElementA/ElmentAA@value”
    originalValue=“xyy”
    proposedValue=“yyy”/>
<tp:NegotiationContent>
…
<CPATemplate>
```

```
<CPA>
  <ElementA>
    <ElementAA value=“yyy”/>
  </ElementA>
  <ElementB id=“b1”/>
  <ElementB id=“b2”/>
</CPA>
```

```
…
<tp:NegotiationContent>
  <tp:AcceptedItem xpath=“/CPA”/>
<tp:NegotiationContent>
…
<CPATemplate>
```

```
<CPA>
  <ElementA>
    <ElementAA value=“yyy”/>
  </ElementA>
  <ElementB id=“b1”/>
  <ElementB id=“b2”/>
</CPA>
```

Figure 4. Example of how to use the NegotiationContent element - Step 2

In Figure 4, the initiating *Party* received the changes proposed by the responding *Party*, made the following changes, and sent back another proposal: Accepted newly added element ***B*** and updated element ***A***/element ***AA***'s attribute value from 'xyy' to 'yyy'. The responding *Party* accepted this change and their corresponding *CPA Templates* reflected their changes.

When the ***NegotiationMessage*** element has its ***messageType*** attribute set to "`CounterOfferPending`", the counter offer might consist of:

- Deleted elements and attributes.
- Inserted elements and attributes.
- Re-ordered elements using a [XPATH]-based list of changes with status of "Required" or "Preferred".
- Changed values of elements and attributes.
- Rejection: with reason(s) for rejection. See Section 11.4 for additional information. Rejection is final. It ends the *Negotiation Dialog* and the two *Parties* SHOULD make human to human contact to resolve their incompatibilities.

11.1.10 ResponseToURL element

The ***ResponseToURL*** element SHALL have a value that is a URI that conforms to [RFC2396] and identifies the return address where the counter offer can be sent asynchronously. If the sender of the *Message* omits the ***ResponseToURL*** element, the responding *Party* MUST send its response *Message* synchronously.

11.1.11 Offer-CounterOffer-Acceptance-Time

The ***Offer-CounterOffer-Acceptance-Time*** element is an element that used in an offer. It is used to specify the time by which a counter offer must be received by the initiating party. It is permissible to specify a different acceptance time for each offer or counter offer.

Deleted: by

11.2 CPA ID, Negotiation Dialog ID, Unique Business Message ID, and InResponseTo

The values of the ***id*** and ***version*** attributes of the ***CPATemplateId*** element SHALL remain the same throughout any *Negotiation Dialog.*

The *Negotiation-Dialog Identifier* is used to identify a particular *Negotiation-Dialog* thread. See the discussion of the *Negotiation-Dialog Identifier* in Section 12.3. The value of the ***id*** attribute of the ***CPATemplateId*** element SHALL NOT be used as the *Negotiation-Dialog Identifier*. See the discussion in Section 12.3.

The value of the ***businessMsgId*** attribute of the ***NegotiationMessage*** element is a unique identifier that identifies the current *Business Message* within the scope of one *Negotiation Dialog*.

The value of the ***InResponseTo*** element is the unique *Business Message* identifier of the last incoming offer or counter-offer *Message* that this current *Message* is responding to.

11.3 Offer and Counter Offer

An offer differs from subsequent counter offers. An offer always contains (or references) the *NDD Document* and the initial *CPA Template*. Each counter offer contains or references the latest version of the *CPA Template,* containing all changes made up to and including the changes accepted prior to this *Message.* The same XML schema defines both the offer and the counter offer *Documents*.

11.4 CPA Offer Rejected

When a *CPA* offer or counter offer is rejected, the rejecting party SHALL set the ***NegotiationMessage*** element's ***messageType*** attribute to "`Rejected`".

Deleted: should

The process of composing the *CPA* from *CPPs* will detect many error conditions before the *Negotiation Process* begins. Others might be discovered during the *Negotiation Process*. Examples are mismatched *Process Specification Document* and mismatched delivery-channel requirements. See the *CPA*-Composition appendix of [ebCPP] for information about error conditions that can be detected during composition of a *CPA Template*.

The rejection *Message* SHALL include reason, contact name, phone, and/or URL for further information. The ***Comment*** element is used for this purpose.

Following are some reasons for rejection. The reason is indicated by the value of ***error*** attribute of ***NegotiationMessage*** element:

- *CPA Template* contents. Examples:
  - ♦ `"ExpiredCPP"`: Expired *CPP*
  - ♦ `"UnableToFulfillSecurityRequirements"`: Unable to fulfill security requirements
  - ♦ `"ProposedSecurityPolicyInadequate"`: Proposed security policy is inadequate
  - ♦ `"OurOfSequenceCounterOffer"`: Out-of-sequence counter offer
  - ♦ `"FailedSignatureValidation_CPATemplate"`: The signature on the *CPA Template* failed validation.
  - ♦ `FailedSignatureValidation_CPA"`: The signature on the agreed *CPA* failed validation
    - The *CPA* is not signed until it is agreed to.
  - ♦ `"UnsupportedPackaging"`: proposed *Packaging* not supported
  - ♦ `"UnsupportedSignal"`: unable to support signals requested (BPSS instance *Document*)
- Business relationship
  - ♦ `"UnsupportedBusinessRelationship"`: *CPA* unsupported without existing *Business* relationship.
- Negotiation Process
  - ♦ `"FailedToConverge"`: In the judgment of the rejecting *Party*, too many counter offers were tried with no forward progress toward convergence.

    NOTE: A future version of this specification might formulate a definition of and protocol for detecting "no forward progress".

  - ♦ `"PreviouslyRejectedCPA"`: Proposed *CPA* previously received and not accepted.
- `"ExpiredOffer"`: The current offer's validity interval has expired.
- `"FormatError"`: CPA format problems
  - ♦ Examples: parsing error, data invalid
- `"UnknownSystemError"`: Internal System Error

11.5 CPA Offer accepted

When a *CPA* offer is accepted, the final *CPA Document* contained in or referenced by the ***CPATemplate*** element SHALL be signed if both *Parties' NDDs* indicate they are capable of signing the final *Document.*

When a negotiation ends in success, the final document will contain, in effect, a completed *CPA* document, even though the element name is still ***CPATemplate***. The content is a fully qualifed *CPA*. The user should be aware and not be confused by the element name in this case.

12 Negotiation Protocol

12.1 General Principles of Negotiation Protocol

Figure 2 in Section 5.2 provides a high-level overview of the *Negotiation Process* including the discovery-related steps and the protocol to negotiate a *CPA*. This section describes the *Negotiation Protocol* in detail including a description of the negotiation BPSS instance *Document*.

> NOTE: Although a BPSS instance document is used to document the details of the *Negotiation Protocol*, there is no requirement to implement BPSS as part of an implementation of this specification. In other words, there is no requirement to deploy the *Negotiation* BPSS instance document into the *negotiation* runtime system.

A *Negotiation Dialog* is a complete execution of the BPSS *Negotiation-Protocol* choreography from the initial offer until the *CPA* is completed successfully or the negotiation fails. A single *Negotiation Dialog* negotiates a single *CPA*.

12.2 CPA Identifier

When a *Party* creates a *CPA Template*, that *Party* SHALL assign a valid value to the ***cpaid*** attribute in the *CPA Template*. See Section 9.8 regarding negotiability of the ***cpaid*** attribute.

12.3 Negotiation-Dialog Identifier

A *Negotiation-Dialog Identifier* identifies the *Negotiation Dialog* from initial offer to completion. Each *Party* SHALL separately maintain the ongoing state information in association with the *Negotiation-Dialog* identifier. The value of the *Negotiation-Dialog* identifier MUST be common to the two *Parties* and MUST be unique among all on-going negotiations between a pair of *Parties*.

The value of the ***cpaid*** attribute of the *CPA Template* SHALL NOT be used as the value of the *Negotiation-Dialog Identifier.* The value of the *Negotiation-Dialog Identifier* SHALL be determined independently of the value of the ***cpaid*** attribute. The reason is to ensure that if a negotiation fails and the same *CPA Template*, with the same value of the ***cpiId*** attribute is used in a second negotiation attempt, uniqueness of the *Negotiation-Dialog Identifier* is preserved.

> NOTE: Although it is not expected that *Negotiation Dialogs* involving the same *CPA Template* will overlap in time, the above rule ensures that saved state information from an earlier attempt at negotiation can be referenced by its *Negotiation-Dialog Identifer* during a later attempt with the same *CPA Template*.

12.4 Offer Identifier

A counter offer MUST be associated with the offer or counter offer to which it is replying. Each offer or counter offer SHALL have a unique *Offer Identifier* defined by the negotiation application. A counter offer states the *Offer Identifier* of the offer or counter offer to which it is replying. The identifiers and the references to them are in the negotiation-*Message* payload.

The *Offer Identifier* MUST be unique among the initial offer and all counter offers issued by a given *Party* within a *Negotiation Dialog*. The *Offer Identifier* is qualified by the *Party* Id of the issuer and the *Negotiation-Dialog Identifier*.

> NOTE: With ebXML *Messaging*, the ***messageId*** and ***refToMessageId*** attributes in the *Message* header could serve the purpose of the *Offer Identifier*. However, to enable alternative *Messaging* protocols, such as "vanilla SOAP", which do not have these identifiers, the *Offer Identifier* is defined at the application level.

12.5 Negotiation Status

The ***Status*** element in the *CPA* records the state of the composition and *Negotiation Protocol*. The states of its ***value*** attribute progress as follows:

- "Proposed" – This value is in the *CPA Template* sent with the initial offer and remains unchanged until an agreed *CPA* is completed.
- "Agreed" – This value is in the completed *CPA* that is sent from one *Party* to the other for validation if the *Parties* had agreed not to sign the *CPA*. This is the final state.
- "Signed" – If the *Parties* had agreed to sign the *CPA*, the *CPA* sent from one *Party* to the other *Party* is signed by the sending *Party* and the value of the ***value*** attribute is "Signed". This is the final state.

> NOTE: Because the ***Status*** element is included in the first *Party's* signature, the value of the ***value*** attribute cannot be changed when the second *Party* signs.

12.6 ebXML Conversation

A single *Negotiation Dialog* corresponds to a single ebXML *Conversation.*

For use with *Message* services, such as "vanilla SOAP", that have no *Conversation* construct, the *Negotiation-Dialog Identifier* serves the purpose of a *Conversation* identifier at the application level

12.7 Negotiation CPA

Prior to the initial offer, a *Negotiation CPA* MUST be activated between the two negotiating *Parties*. See Section 5.1 for a possible scenario.

> NOTE: The negotiation subteam plans to simplify the NCPA to the greatest extent possible. The goal is to define a default NCPA that could be built into the *Negotiation* run-time systems and not have to be explicitly composed, negotiated, or deployed.

12.8 Initial Offer

A *Party* (B) can create and send an initial offer to another *Party* (A) in different ways, depending on whether *Party* B is starting with *Party* A's *CPP* or *CPA Template*.

- If *Party* B discovered the *CPP* of *Party* A (a potential trading partner), *Party* B composes a *CPA Template* from its *CPP* and *Party* A's *CPP*. *Party* B then prepares an *NDD* that describes what is negotiable in the *CPA Template*. If *Party* A had also published an *NDD*, *Party* B SHOULD take that *NDD* into account in preparing the *NDD* for the initial offer.
- If *Party* B discovered the *CPA Template* and *NDD* of *Party* A, *Party* B modifies the *CPA*

Template to include information about itself, makes other modifications to negotiable items in the *CPA Template* that are indicated in the *Party* A's *NDD*, and prepares a new *NDD* to go with the modified *CPA Template*.

In either case, *Party* B is also responsible for inserting into the *CPA Template* the ***Start***, ***End***, and other elements that are present in a *CPA* but not in a *CPP*.

If *Party* B creates the initial offer by modifying *Party* A's published *CPP* or *CPA Template*, *Party* A SHOULD include a list of changes (Accepted, Deleted, Updated, Inserted) in the initial-offer *Message* (Negotiation Content section) in addition to the initial-offer information

Party B then submits the new *CPA Template* and *NDD* to *Party* A as an initial offer.

It is RECOMMENDED that the *CPA Template* in an initial offer be signed by the offering *Party*.

12.9 Simultaneous Initial Offers

Two *Parties* might simultaneously discover each other and send each other initial offers. Since the two initial offers will cause creation of two independent *Negotiation Dialogs*, this race condition might only be discoverable and resolvable at the application level. Human contact will be necessary to decide which *Negotiation Dialog* to proceed with.

12.10 Offer and Counter Offer

When a *Party* proposes an offer or counter offer, the details of the offer or counter offer are expressed in a negotiation *Message*. The original *NDD* SHALL NOT be altered during the course of the negotiation.

If *Party* A initiates the *Negotiation Dialog* by sending *Party* B an offer, *Party* B sends back a counter offer. In order to counter this counter offer, *Party* A sends another counter offer to *Party* B. In other words, only the initiating *Message* is an offer; the rest of the negotiation will be conducted by exchanging counter offers. Each counter offer message contains or references an updated version of the *CPA Template* that contains all changes up to, and including, the latest changes accepted prior to this *Message.*

Throughout the *Negotiation Dialog*, each *Party* can terminate the negotiation by sending a *Message* rejecting an offer or counter offer without proposing a counter offer. Human to human contact is encouraged after a *CPA* is rejected, in order to resolve any impasse before initiating a brand new *Negotiation Dialog*.

Deleted: "CPA offer Rejected" in responding to an incoming

Deleted: "CPA offer Rejected" is sent

A counter offer SHALL only refer to items that are listed in the *NDD*. Any offer or counter offer that is outside the limits defined in the *NDD* MUST be rejected.

A counter offer SHALL NOT propose a wholesale change of subject matter. For example a counter offer SHALL NOT propose changes in the *Roles* of the participants.

A *Party* that wishes to propose a different BPSS instance *Document* SHALL reject the received offer or counter offer and can then issue its own initial offer including the desired BPSS instance *Document*.

A counter offer SHALL NOT introduce a new *NDD*. To introduce a new *NDD*, a *Party* SHALL reject the received offer or counter offer and can then issue its own initial offer including the desired *NDD*.

When responding to an offer or counter offer, a *Party* SHALL indicate in its counter offer, which items in the prior offer or counter offer it accepted.

If a counter offer contains only indications of acceptance of items, the *Party* that sent it is indicating acceptance of the *CPA* as modified by the prior steps in the negotiation.

Once agreement has been reached on any part of the *CPA,* those elements and attributes SHALL NOT be reopened for negotiation.

12.10.1 Responses to Offer and Counter Offer

A number of responses can be given to an offer or counter offer. The responses fall into the following categories:

- Acceptance: Acceptance of an offer or counter offer means that the *Party* that received the offer or counter offer is accepting all remaining open items and hence the two *Parties* have reached agreement.
- Counter offer pending: The *Party* that received the offer or counter offer wishes to negotiate further on some or all open items and is going to send its own counter offer.
- Rejection: The *Party* that received the offer or counter offer believes that agreement cannot be reached. Human contact is REQUIRED in order to resove the incompatibilities.
- The responses are discussed in Section 11.4.

12.10.2 Offer-Counter Offer Acceptance Time

A maximum time (interval) for acceptance is associated with each offer or counter offer. The acceptance interval is a business-level timeout; processing it is independent of any *Document*-exchange or transport-level *Message*-loss recovery rules. When the acceptance interval expires without a response, the initiator SHALL record the current *Negotiation Dialog* as terminated.The acceptance interval is in the *Negotiation Message* and can be varied for each *Message.*

12.11 Conclusion of Negotiation

The negotiation concludes when agreement has been reached. This might happen either by one *Party* accepting the initial offer or following an exchange of counter offers.

If agreement is reached on the initial offer, and the *Party* that received the initial offer does not have to add any information to the *CPA Template*, the negotiation concludes immediately. The *Party* that received the initial offer SHALL send a *Message* indicating acceptance and the *CPA Template* becomes the agreed *CPA*. If signing is included in the initial offer, the offering *Party* SHALL sign the *CPA Template* before sending it. The receiving *Party* SHALL then sign and return the *CPA*. At this point, the *Parties* are ready to deploy the *CPA* into their run-time systems and commence business. If the second *Party* does not agree to sign, and signing is negotiable, it

SHALL respond with a counter offer that excludes signing instead of accepting the initial offer. See Section 12.12 for a discussion of negotiating signing of the *CPA*.

When agreement has been reached following exchanges of counter offers, the *Party* that received and accepted the final counter offer SHALL send the completed *CPA* (or its URL) to the other *Party* for approval. The receiving *Party* SHALL respond, indicating either approval or rejection. If signing was agreed to, the sending *Party* SHALL sign the *CPA* before sending it. The receiving *Party* SHALL check that the new *CPA* conforms to its understanding of the contents of the *CPA*. The receiving *Party* can also validate the first *Party's* signature. If the receiving *Party* approves the *CPA*, the receiving *Party* SHALL sign the *CPA* over the first *Party's* signature and return it to the first *Party*. Otherwise the receiving *Party* SHALL respond indicating rejection.

The *Party* that received the completed *CPA* SHALL respond in one of the following ways:

- *Message* indicating that a completed *CPA* was received and accepted (***BusinessDocument*** name = “CPA Final Response Doc”)
- *Message* that sends a completed *CPA* signed by the sender (***BusinessDocument*** name = “CPA Final Response Doc Signed”).
  - ♦ Used when signing was agreed to and the received *CPA* was signed by the sending *Party*.
- *Message* indicating that a completed *CPA* was received and rejected (***BusinessDocument*** name = “CPA Ffinal Response Reject Doc”).

Deleted: A separate indicator in the *Message* distinguishes between accept and reject.¶

Following are some reasons for rejecting the received *CPA*:

- The final *CPA* does not agree with the recipient's understanding of the contents of the *CPA* (some kind of state-tracking mismatch).
- The signature on the final *CPA* cannot be validated.
- The final *CPA* was not signed although signing was agreed to.

When signing by both *Parties* was agreed to, the *Party* that received the double-signed *CPA* SHALL test for the following conditions:

- The double-signed *CPA* is acceptable.
- The double-signed *CPA* is rejected. Reasons to reject this *CPA* include:
  - ♦ The second signature on the double-signed *CPA* cannot be validated.
  - ♦ An acknowledgment was received when a double-signed *CPA* was expected.

Acceptance and rejection of the double-signed *CPA* are indicated by business signals. See Section 12.13.2 for details.

Rejection at this stage is a fatal condition and the *Negotiation Dialog* SHALL be terminated. It is RECOMMENDED that the two *Parties* confer to resolve the discrepancy and then renegotiate the *CPA*. If the resolution of the discrepancy was successful, the renegotiation will generally consist of one *Party* sending a new offer that the other *Party* can accept without a counter offer.

12.12 Signing the CPA

Signing of the completed and agreed-to *CPA* is an item of negotiation. Refer to [ebCPP] regarding how to sign the *CPA*. Negotiation of signing is accomplished by negotiating the presence of the *CPA* ***Signature*** element and its child ***ds:Signature*** elements. Following are the

Deleted: Signing the completed CPA proves who signed it (“legal” signing) and provides the usual integrity check on the contents of the CPA. Signing of the completed and agreed-to CPA is an item of negotiation. Refer to [ebCPP] regarding how to sign the CPA.¶
¶

outcomes:

- Agree not to sign: The ***Signature*** element SHALL be omitted from the final *CPA*.
- Agree on 2-*Party* signing: The final *CPA* SHALL contain the ***Signature*** element with two ***ds:Signature*** elements.
- Agree on 3-*Party* signing: The final *CPA* SHALL contain the ***Signature*** element with three ***ds:Signature*** elements.

It is important to understand that the ***ds:Signature*** elements MUST be incorporated into the *CPA* one at a time, as the *Parties* sign. The ***Signature*** element MUST NOT be inserted into the *CPA* until the first *Party* signs. If it is incorporated earlier, the *CPA* will fail validation against the CPPA XML Schema because there will be no child ***ds:Signature*** elements.

If the *Parties* agreed to third-*Party* signing, they SHALL obtain the third *Party's* signature before commencing to do business under the *CPA*. The means of obtaining the third *Party's* signature are not defined in this specification.

12.13 BPSS Instance Document for Automated Negotiation

THIS SECTION IS A COMPLETE REPLACEMENT. PLEASE REVIEW IN FULL.

COMMENT FROM MAR. 2003 FACE TO FACE MEETING: WE NEED TO DO SOME RATIONALIZATION OF THINGS LIKE HOW ELEMENTS ARE ORDERED IN THE BPSS INSTANCE DOCUMENT.

The choreography of the *Negotiation Protocol* is defined by an instance *Document* of the ebXML *Business Process Specification Schema*[ebBPSS]. The BPSS instance *Document* for automated negotiation is in Appendix D.

> NOTE: Although in [ebCPP], the use of a BPSS instance *Document* to describe choreography is not required, this specification depends intimately on the selected choreography description. Therefore, the use of a BPSS instance *Document* is normative for this version of this specification.

This BPSS instance *Document* defines the negotiation choreography beginning with an exchange of an offer and response.

- If the response to the offer is to accept the offer, the choreography transitions to the final *CPA* exchange (see below).
- If the response to the offer is to reject the offer, the choreography immediately concludes.
- If the response to the offer is that a counter offer is pending, the choreography then goes into an alternation of counter offer and response between the two *Parties* which continues until:
    - ♦ Acceptance of the offer causes the choreography to transition to the final *CPA* exchange.
    - ♦ Rejection of the offer concludes the choreography.

Several *Business Document* names are defined directly under the ***ProcessSpecification*** element and referenced in various places as described below.

The BPSS instance *Document* defines initiator and responder *Role* names for each binary

Deleted: “
Deleted: ”
Deleted: “
Deleted: ”
Deleted: “
Deleted: ”
Deleted: A response of “accept offer”
Deleted: A response of “r
Deleted: ”

collaboration, collaboration activity, and binary transaction activity. For simplicity in the explanation below, in most cases, the terms “initiator” and “responder” are used. For each stage of the choreography, the *NCPA* associates *Role* names with actual *Parties* in the ***Action*** elements under the ***CollaborationRole*** elements.

12.13.1 Offer-Counter-Offer Choreography

A counter offer is a requesting *Document* in a new *Business Transaction*, not a response to an offer. To issue a counter offer, the recipient of an offer SHALL send a reply that indicates that a counter offer” is pending and then issue the counter offer as a new *Business Transaction*. This avoids a race condition with respect to which *Party* sends the next *Message*. It also avoids any need to for the two *Parties* to switch *Roles*.

Deleted: “

Deleted: pending

Deleted: ¶
THE FOLLOWING DISCUSSION HAS TO BE UPDATED TO CORRESPOND TO CHANGES IN THE BPSS INSTANCE DOCUMENT THAT WERE MADE SINCE THIS SECTION WAS WRITTEN.¶

The choreography begins with the “CPA_Offer_BTA” ***BusinessTransactionActivity*** element under the “CPA_Negotiation_BC” ***BinaryCollaboration*** element. A *CPA* offer *Message* is sent from the “CPA_Negotiation_Initiator_Role” *Party* to the “CPA_Negotiation_Responder_Role” *Party* by means of the “CPA_Offer_BT” *Business Transaction*. The “CPA_Offer_BT_ReqBA” ***RequestingBusinessActivity*** sends the “CPA_Offer_Doc” *Message* from the initiator *Party* to the responder *Party*. The “CPA_Offer_BT_RespBA” ***RespondingBusinessActivity*** then sends the response *Message* from the responder *Party* to the initiator *Party*. This *Message* is then evaluated as defined by the ***Failure*** and ***Transition*** elements under the “CPA_Negotiation_BC” binary collaboration. These are the elements whose ***fromBusinessState*** attribute has the value “CPA Offer BTA”. The value of the ***expression*** attribute in each of these elements is the name of the response *Message*, as follows:

Deleted: ”

Deleted: -

Deleted: CPA_Offer_ReqBA

Deleted: *Success*,

Deleted: ,

- ***Transition*** element:
  - ♦ If the response *Message* is “CPA Accept Offer Doc”, the proposed *CPA* has been accepted by the responder *Party* and the choreography transitions to the final *CPA* exchange. The ***toBusinessState*** attribute of the ***Transition*** element identifies “CPA Final BTA” (the name of the ***BusinessTransactionActivity*** element) as the next state in the choreography.
  - ♦ If the response *Message* is “CPA Counter Pending Offer Doc”, the responder *Party* will send a counter offer as the next *Business Transaction*. The ***toBusinessState*** attribute of the ***Transition*** element identifies “CPA Counter Offer CA” (the name of the ***CollaborationActivity*** element) as the next state in the choreography.
- ***Failure*** element: If the response *Message* is “CPA Reject Offer Doc”, the proposed *CPA* has been unconditionally rejected by the responder *Party* and the choreography concludes.

Deleted: ***Success***

If the response *Message* to the “CPA_Offer_Doc” *Message* was “CPA Counter Pending Offer Doc”, the transition described above takes place and takes the choreography to state “CPA Counter Offer CA”, i.e. to the ***CollaborationActivity*** element named “CPA Counter Offer CA”. This ***CollaborationActivity*** element references the “CPA_Negotiation_CounterOfferBC” ***BinaryCollaboration*** element.

Deleted: ***Transition*** element: If the response *Message* is “CPA Counter Pending Offer Doc”, the responder *Party* will send a counter offer as the next *Business Transaction*. The ***toBusinessState*** attribute of the ***Transition*** element identifies “CPA Counter Offer CA” (the name of the ***CollaborationActivity*** element) as the next state in the choreography.¶
¶

The initial request *Message* is under the “CPA_Counter_Offer_1_BTA” ***BusinessTransactionActivity*** element, which is the “from” state for the following. The *Party* which received the original *CPA* offer is now the initiator in this ***BusinessTransactionActivity***. The *Message* is sent from the “CPA_Negotiation_CounterOfferInitiator_Role” *Role* to the

“CPA_Negotiation_CounterOfferResponder_Role” *Role* by means of the “CPA_Counter_Offer_BT” ***BusinessTransaction***. The *Message*,”CPA_Counter_Offer_Doc”, is identified in the “CPA_Counter_Offer_BT_ReqBA” ***RequestingBusinessActivity***. The response *Message* is sent by means of the “CPA_Counter_Offer_BT_RespBA” ***RespondingBusinessActvity***. One of three response *Messages* can be sent, as discussed below. The response *Message* is then evaluated as defined by the ***Failure*** and ***Transition*** elements under the “CPA_Negotiation_CounterOfferBC” ***BinaryCollaboration***.

- ***Transition*** element:
  - ♦ If the response *Message* is “CPA Accept Offer Doc”, the proposed *CPA* has been accepted by the responder *Party* and the choreography transitions to the final *CPA* exchange. The ***toBusinessState*** attribute of the ***Transition*** element identifies the “CPA Final BTA Init Responder” *Business Transaction* Activity as the next state in the choreography.
  - ♦ If the response *Message* is “CPA Counter Pending Offer Doc”, the responder *Party* will send a counter offer as the next *Business Transaction*. The ***toBusinessState*** attribute of the ***Transition*** element identifies the “CPA Counter Offer 2 BTA” *Business Transaction* Activity as the next state in the choreography.
- ***Failure*** element: If the response *Message* is “CPA Reject Offer Doc”, the proposed *CPA* has been unconditionally rejected by the responder *Party* and the choreography concludes.

If the above transition takes place, it means that the *Party* that was the responder now becomes the initiator to supply a counter offer to the counter offer. The “CPA_Counter_Offer_2_BTA” ***BusinessTransactionActivity*** is now performed in the same manner as the “CPA_Counter_Offer_1_BTA” ***BusinessTransactionActivity***, described above. Both the ***BusinessTransactionActivity*** “CPA_Counter_Offer_1_BT” and “CPA_Counter_Offer_2_BTA” use the same ***BusinessTransaction*** “CPA_Counter_Offer_BT”.

The choreography then iterates between the “CPA_Counter_Offer_1_BTA” ***BusinessTransactionActivity*** and the “CPA_Counter_Offer_2_BTA” ***BusinessTransactionActivity*** until success or failure is achieved. If the response *Message* is “CPA Accept Offer Doc”, the choreography then transitions to the final *CPA* exchange. These transitions are indicated by a transition to “CPA Final BTA Init Responder” if the “CPA Accept Offer Doc” message is the response message from ***BusinessTransactionActivity “CPA Counter Offer 1 BTA”*** and “CPA Final BTA Init Initiator” if the “CPA Accept Offer Doc” is the response message from ***BusinessTransactionActivity “CPA Counter Offer 2 BTA”.*** Failure ends the choreography.

12.13.2 Final CPA exchange

When either the initial offer or a counter offer is accepted in full, the choreography transitions to the “*CPA* Final BT” ***Business Transaction***. The purpose of this *Business Transaction* is for the *Party* that accepted the offer or counter offer to send the completed *CPA* to the other *Party* and for the Responding *Party* of this ***Business Transaction*** to either accept or reject the final CPA. If required, the final CPA sent by the *Party* that accepted the offer can be signed and the response message can include a double signed CPA.

If the initial offer was accepted, the next *Business* state is the “*CPA* Final BTA”

Deleted: ***Success***,

Deleted: ,

Deleted: ***Success***

Deleted: <#>***Transition*** element: If the response *Message* is “CPA Counter Pending Offer Doc”, the responder *Party* will send a counter offer as the next *Business Transaction*. The ***toBusinessState*** attribute of the ***Transition*** element identifies the “CPA Counter Offer 2 BTA” *Business Transaction* Activity as the next state in the choreography.¶
<#>¶

Deleted: CPA

Deleted: Counter

Deleted: Offer

Deleted: 2

Deleted: CPA

Deleted: Counter

Deleted: Offer

Deleted: 2

Deleted: a

Deleted: Success causes the

Deleted: to

Deleted: se

Inserted: se

Deleted: are

Inserted: are indicated by

Deleted: s

Deleted: ”

Deleted: Init Responder

BusinessTransactionActivity, which references the “*CPA* Final BT” *Business Transaction*. The initiator *Role “CPA Negotiation Responder”* for the “Final_*CPA*_BT_ReqBA” requesting *Business* activity sends the “*CPA* Final Doc” *Message*, containing the *CPA* or its URL, to the other (responder) *Party* *“CPA Negotiation Initiator”* for this ***Business Transaction***. *“CPA Final Doc”* can include a signed *CPA* if this was negotiated. The responder *Party* checks the *CPA* and performs the responding *Business* activity conveying one of:

- The “CPA Final Response Doc” *Message*, to acknowledge receipt of an acceptable *CPA*.
- The “CPA Final Response Reject Doc” *Message*, to acknowledge receipt of an unacceptable *CPA* and reject the final proposed *CPA*. One of the reasons for rejecting the final proposed *CPA* could be that the signature on the proposed *CPA* may not be valid.
- The “CPA Final Response Doc Signed” *Message*, to acknowledge receipt of an acceptable signed *CPA* and return that *CPA* with the responder *Party’s* signature over the initiator *Party’s* signature.

The response *Message* is then evaluated as defined by the ***Success*** and ***Failure*** elements under the “CPA_Negotiation_BC” ***BinaryCollaboration***.

- ***Success*** element:
  - ♦ If the response *Message* is “CPA Final Response Doc”, the final *CPA* has been accepted by the responder *Party* and the choreography concludes.
  - ♦ If the response *Message* is “CPA Final Response Doc Signed”, the final *CPA* has been accepted by the responder *Party* and the choreography concludes. Please refer to the discussion below on how the choreography could conclude with a failure because of failure to verify the double-signed *CPA* by using a process level *Exception*.
- ***Failure*** element: If the response *Message* is “CPA Final Response Reject Doc”, the final *CPA* has been rejected by the responder *Party* and the choreography concludes.

If the counter offer was accepted, the next *Business* state is the “*CPA* Final BTA INIT Initiator” or “CPA Final BTA Init Responder” ***BusinessTransactionActivity*** depending on the state of “CPA_Negotiation_CounterOfferBC”. These ***BusinessTransactionActivity*** elements reference the “CPA Final BT” ***BusinessTransaction***. The sequence of request and response messages is similar to ***BusinessTransactionActivity*** “*CPA* Final BTA” as described above.

The response *Message* is then evaluated as defined by the ***Success*** and ***Failure*** elements under the “CPA_Negotiation_Counter OfferBC” ***BinaryCollaboration***.

- ***Success*** element:
  - ♦ If the response *Message* is “CPA Final Response Doc”, the final *CPA* has been accepted by the responder *Party* and the choreography concludes.
  - ♦ If the response *Message* is “CPA Final Response Doc Signed”, the final *CPA* has been accepted by the responder *Party* and the choreography concludes. Please refer to the discussion below on how choreography could conclude with a Failure because of failure to verify the double signed CPA by using a process level *Exception*.
- ***Failure*** element: If the response *Message* is “CPA Final Response Reject Doc”, the final *CPA* has been rejected by the responder *Party* and the choreography concludes.

The *Party* that receives the final (double signed) *CPA* SHOULD test it for possible error conditions as described in Section 12.11. The *Party* that received the double-signed *CPA*

Deleted: *Party*

Deleted: Req BA

Deleted: or rejection. Acceptance and rejection are indicated by values of the status indicator in the *Negotiation Message*

Deleted: ***THE ABOVE SENTENCE NEEDS TO BE REVISED TO USE THE CORRECT NAME OF THE STATUS ELEMENT OR ATTRIBUTE WHEN THE MESSAGE SCHEMA IS COMPLETED.¶***

Deleted: ***Success*** element:

Deleted: note

Deleted: F

Deleted: ***a***

Inserted: ***activity*** depending on the state of “CPA_Negotiation_CounterOfferBC”. These ***BusinessTransactionActivity***

Deleted: S

Inserted: Sequence of request and response messages is similar to ***BusinessTransactionActivity*** “*CPA* Final BTA” as described above.¶
¶
The response *Message* is then evaluated as defined by the ***Success*** and ***Failure*** elements under the “CPA_Negotiation_Counter OfferBC” ***BinaryCollaboration***. ¶
Success element:

Deleted: ***Success*** element:

Inserted: ***Success*** element:

Deleted: note

Inserted: note below on how choreography could conclude with a Failure because of failure to verify the double signed CPA by using a process level *Exception*.¶

SHALL reply with the AcceptanceAcknowledgment business signal if the *CPA* is acceptable or with the Exception business signal if the *CPA* is not acceptable. These signals are instance *Documents* of the business signals defined in [ebBPSS].

If a counter offer was accepted in full, the choreography transitions to the "CPA Final BTA Init Initiator" ***BusinessTransactionActivity***. That ***BusinessTransactionActivity*** references the "CPA Final BT" ***BusinessTransaction*** and proceeds as for acceptance of an initial offer.

12.13.3 Negotiation Business Signals

HIMA MUKKAMALA SUGGESTED CREATING EXAMPLES OF SIGNALS THAT SHOW HOW SOMEONE COULD SEND A NEGATIVE ACCEPTANCE ACKNOWLEDGMENT TO DECLINE A DOUBLE SIGNED CPA.

12.13.4 State Diagrams

The choreography is illustrated by the state diagram shown in Figure 5 and Figure 6.

THE NEW VERSION OF THE STATE DIAGRAM HAS A PROBLEM. IF THE DIAGRAMS ARE PASTED INTO THIS WORD DOCUMENT DIRECTLY FROM THE SOURCE HTM FILE, THEY CANNOT BE RENDERED PROPERLY BY ACROBAT ALTHOUGH THEY ARE RENDERED PROPERLY BY MICROSOFT WORD. TO OVERCOME THIS PROBLEM FOR THE TIME BEING, THE FOLLOWING PROCEDURE WAS USED.

- ***OPEN THE SOURCE HTM FILE IN IE.***
- ***ENLARGE THE WINDOW A BIT TO MAKE THE SMALL LABELS LEGIBLE.***
- ***PRINT SCREEN.***
- ***PASTE INTO PAINT.***
- ***SELECT THE DIAGRAM WITHIN THE PAINT WINDOW.***
- ***COPY, PASTE SPECIAL INTO MICROSOFT WORD AS A BITMAP IMAGE OBJECT.***

Deleted: Examples of these instance *Documents* are in Appendix E .

Deleted:

Deleted: *a*

Deleted:

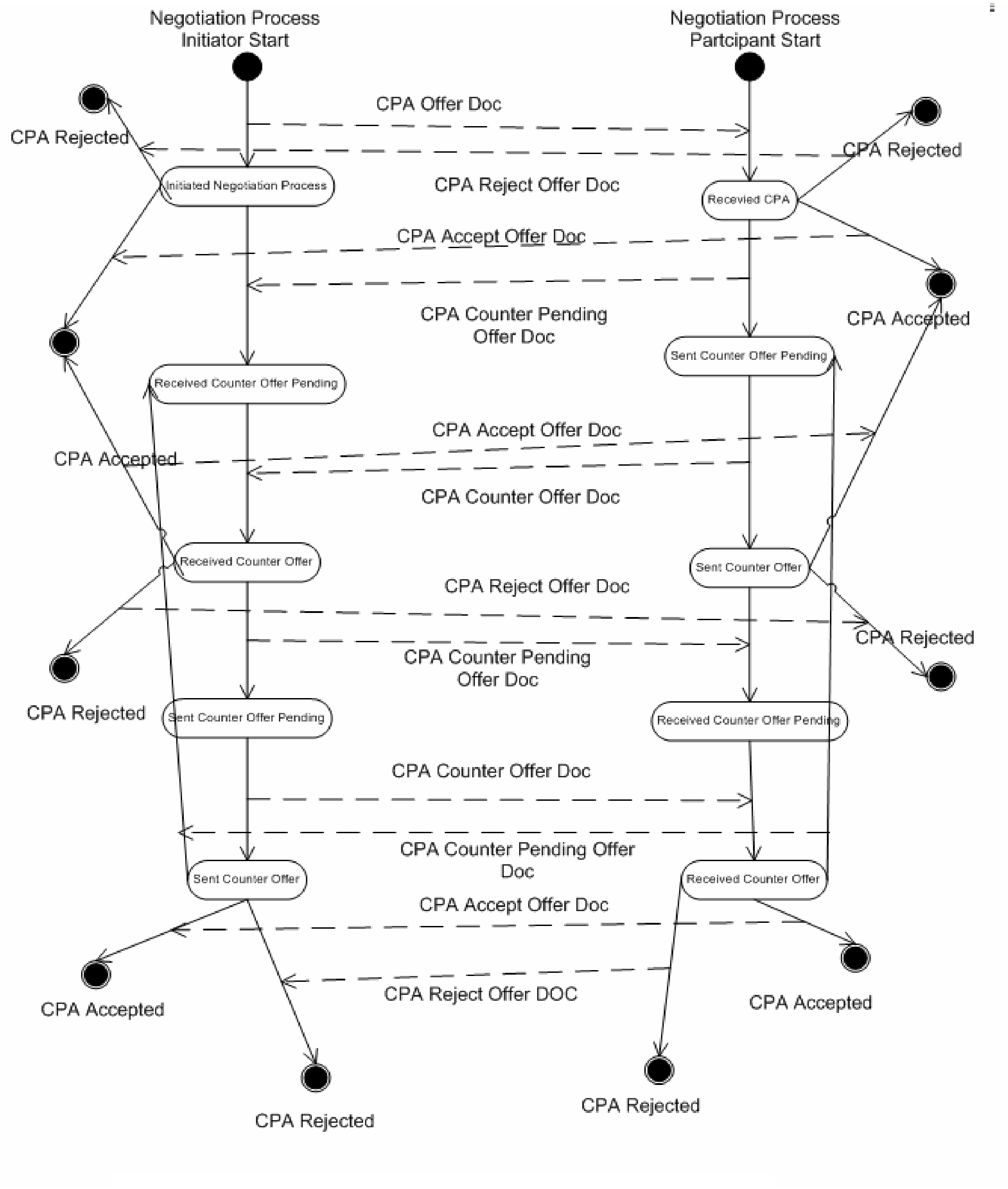
Deleted: *a*

Deleted:

Deleted: **<#>Negotiation Business Signals¶**
¶
ADD DISCUSSION OF THE CONTENTS OF THE BUSINESS SIGNALS.¶
¶
IS IT FEASIBLE TO CONSTRUCT EXAMPLES OF INSTANCE DOCUMENTS OF THE BUSINESS SIGNALS THAT ARE SPECIFIED IN THE NEGOTIATION BPSS INSTANCE DOCUMENT?¶

Deleted: ***THE STATE DIAGRAMS NEED SOME CORRECTIONS:¶***
- IN *Error! Reference source not found.*, ***THE DOCUMENT NAMES IN THE CENTER DO NOT ALWAYS AGREE WITH THE NAMES IN THE INSTANCE DOCUMENT. ¶***
- IN *Error! Reference source not found.*:¶
+ UNLIKE *Error! Reference source not found.*, ***THE DOCUMENT NAMES ARE NOT USED.¶***
THE RETURN OF THE DOUBLE-SIGNED CPA IS NOT SHOWN. PRESUMABLY + IT IS ANOTHER OUTPUT FROM THE "RECEIVED FINAL CPA" STATE.¶

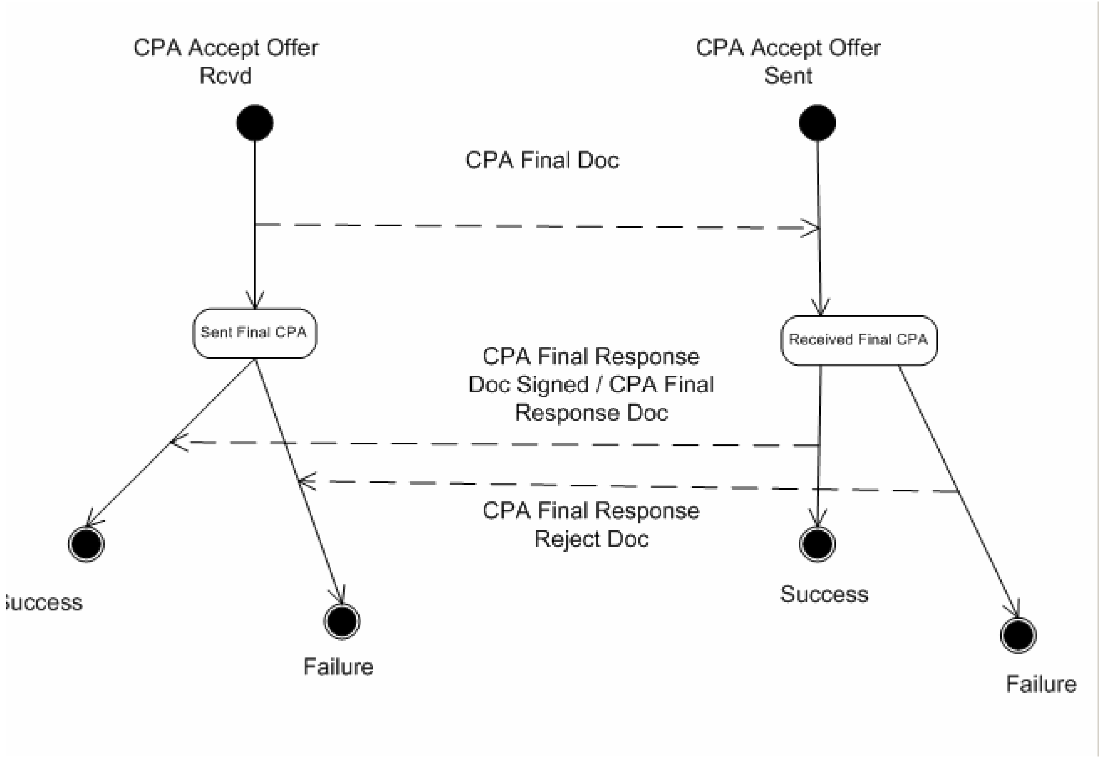
1943



1944

Figure 5. State Diagram for Initial Offer and Counter Offers

1945



1946
1947

Figure 6. State Diagram for Final Transaction

13 Negotiation Algorithm

The negotiation algorithm is an application (*Business Process*). It is embodied in the private process at each *Party*. Note that the BPSS instance *Document* describes only the choreography of the *Message* exchanges and not the private processes. This section discusses the normative aspects of negotiation algorithms, i.e. the rules that ensure interoperability between two *Parties'* implementations of the negotiation algorithm.

> NOTE: The negotiation algorithm is out of scope for version 1 of this specification. This section provides a brief introduction and serves as a place holder for material that might be introduced in future versions.

Historically, research on negotiation has categorized negotiations as follows:

1. Simple matchmaking: The subject (set of negotiable parameters) is static and the ontology is clear. The two *Parties* have a common understanding of the meanings, values, and interdependencies of the negotiable parameters. The utility functions are binary (acceptable *vs*. not acceptable). Negotiation in these situations can be easily automated.
2. Negotiations on static subjects: This is similar to (1) except that the utility functions are more complex (more than 2 choices, numerical values, etc.). These situations can be automated but might require human intervention.
3. Negotiations on dynamic subjects: Here, the negotiable parameter set can be expanded during the process of negotiation and the parameters are more likely to interact than in (1) and (2). Dynamic subject negotiations are much more difficult to automate.

Negotiation of a *CPA* is mostly category 1 with some amount of category 2. On the other hand, business-level negotiations involve much more complex subject matter with parameters that are very likely to interact (consider price and delivery time). Therefore, these negotiations can be expected to be a mixture of categories 2 and 3.

13.1 CPPs and NDDs

It is RECOMMENDED that the negotiation algorithms refer to the *Parties'* original *NDDs* (if available) that go with the *CPPs* as well as the *CPA Template* and its *NDD* to assist in evaluating offers and counter offers. The original *NDDs* might contain information, such as a *Party's* original preference ordering and other constraints that might have been lost when the composite *NDD* was composed for the *CPA Template*.

14 References

VERSION NUMBERS AND URLS TBD.
CHECK THE URLS THAT ARE HERE FOR CORRECTNESS.

[bpPATT] ebXML E-Commerce Patterns, version 1.0, http://www.ebxml.org/specs/bpPATT.pdf

[ebBPSS] ebXML Business Process Specification Schema

[ebCPP] ebXML Collaboration-Protocol Profile and Agreement Specification, version 2.0.

[ebMS] ebXML Message Service Specification, version 2.0.

[ebRS] ebXML Registry Services Specification

[RFC2119] Key Words for use in RFCs to indicate Requirement Levels, Internet Engineering Task Force RFC 2119, http://www.ietf.org/rfc/rfc2119.txt

[RFC2396] Uniform Resource Identifiers URI): General Syntax, Internet Engineering Task Force RFC 2396, http://www.ietf.org/rfc/rfc2396.txt

[SOAPATTACH] SOAP Messages with Attachments, John J. Barton, Hewlett Packard Labs; Satish Thatte and Henrik Frystyk Nielsen, Microsoft, Published Oct 09 2000. http://www.w3.org/TR/2000/NOTE-SOAP-attachments-20001211

[XML] Extensible Markup Language (XML), World Wide Web Consortium, http://www.w3.org/XML.

[XMLDSIG] XML Signature Syntax and Processing, Worldwide Web Consortium, http://www.w3.org/TR/xmldsig-core/

[XMLENC] XML Encryption Syntax and Processing, Worldwide Web Consortium, http://www.w3.org/TR/2002/CR-xmlenc-core-20020304/

[XMLNS] Namespaces in XML, Worldwide Web Consortium, http://www.w3.org/TR/REC-xml-names/

[XPATH] XML Path Language (XPath) Version 1.0, http://www.w3.org/TR/xpath

15 Conformance

In order to conform to this specification, an implementation:

a) SHALL support all the functional and interface requirements defined in this specification,
b) SHALL NOT specify any requirements that would contradict or cause non-conformance to this specification.

A conforming implementation SHALL satisfy the conformance requirements of the applicable parts of this specification.

The objective of conformance testing is to determine whether an implementation being tested conforms to the requirements stated in this specification. Conformance testing enables vendors to implement compatible and interoperable systems. Implementations and applications SHALL be tested using available test suites to verify their conformance to this specification.

Publicly available test suites from vendor neutral organizations such as OASIS and the U.S.A. National Institute of Science and Technology (NIST) SHOULD be used to verify the conformance of implementations, applications, and components claiming conformance to this specification. Open-source reference implementations might be available to allow vendors to test their products for interface compatibility, conformance, and interoperability.

15.1 NDD and Negotiation Messages

An implementation of a tool or service that creates or maintains ebXML instance *Documents* of the *NDD* and *Negotiation Messages* SHALL be determined to be conformant by validation of the instance *Documents*, created or modified by said tool or service, against the XML Schema[XMLSCHEMA-1] definition of these *Documents* in Appendix A and Appendix B, respectively, and available from

URLS TO BE SUPPLIED

by using two or more validating XML Schema parsers that conform to the W3C XML Schema specifications[XMLSCHEMA-1, XMLSCHEMA-2].

15.2 NCPA Instance Document

An implementation of a tool or service that creates or maintains *NCPA* instance *Documents* SHALL be determined to be conformant by validation of the *NCPA* instance *Documents*, created or modified by said tool or service, against the XML Schema[XMLSCHEMA-1] definition of the *CPA* in [ebCPP]and available from

http://www.oasis-open.org/committees/ebxml-cppa/schema/cpp-cpa-2_0.xsd

by using two or more validating XML Schema parsers that conform to the W3C XML Schema specifications[XMLSCHEMA-1, XMLSCHEMA-2].

15.3 Negotiation BPSS Instance Document

An implementation of a tool or service that creates or maintains negotiation BPSS instance

Documents SHALL be determined to be conformant by validation of the BPSS instance *Documents*, created or modified by said tool or service, against the XML Schema[XMLSCHEMA-1] definition of the BPSS in available from

URL TO BE SUPPLIED.

by using two or more validating XML Schema parsers that conform to the W3C XML Schema specifications[XMLSCHEMA-1, XMLSCHEMA-2].

15.4 Negotiation Business Signals

An implementation of a tool or service that creates or maintains negotiation *Business*-signal instance *Documents* SHALL be determined to be conformant by validation of the *Business*-signal instance *Documents*, created or modified by said tool or service, against the XML Schema[XMLSCHEMA-1] definition of the *Business* signals and available from

URL TO BE SUPPLIED.

by using two or more validating XML Schema parsers that conform to the W3C XML Schema specifications[XMLSCHEMA-1, XMLSCHEMA-2].

16 Disclaimer

The views and specification expressed in this document are those of the authors and are not necessarily those of their employers. The authors and their employers specifically disclaim responsibility for any problems arising from correct or incorrect implementation or use of this design.

17 Contact Information

Martin W. Sachs (sub team lead)
Cyclone Commerce
8388 E. Hartford Drive
Scottsdale, AZ 85255
USA
Phone: 203-226-0524
email: msachs@cyclonecommerce.com

Deleted: ***OTHERS TBD*¶**
¶

Dale W. Moberg (author)
Cyclone Commerce
8388 E. Hartford Drive
Scottsdale, AZ 85255
USA
Phone: 480-627-2648
email: mailto:dmoberg@cyclonecommerce.com

Notices

Portions of this document are copyright © 2002 OASIS.

Deleted: ***NEED TO DETERMINE OF UN/CEFACT HAS TO BE MENTIONED.¶***

OASIS takes no position regarding the validity or scope of any intellectual property or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; neither does it represent that it has made any effort to identify any such rights. Information on OASIS's procedures with respect to rights in OASIS specifications can be found at the OASIS website. Copies of claims of rights made available for publication and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification, can be obtained from the OASIS Executive Director.

OASIS invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights which may cover technology that may be required to implement this specification. Please address the information to the OASIS Executive Director.

OASIS has been notified of intellectual property rights claimed in regard to some or all of the contents of this specification. For more information consult the online list of claimed rights.

Appendix A XML Schema for Negotiation Descriptor Document (Normative)

The XML Schema document for the *NDD* is available as a text file at:

FILL IN THE URLS OF THE XML DOCUMENTS IN ALL THE APPENDICES.

THIS APPENDIX HAS BEEN COMPLETELY REPLACED.

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- edited with XML Spy v4.4 U (http://www.xmlspy.com) by neelakantan kartha (Sterling Commerce) -->
<schema targetNamespace="http://www.oasis-open.org/committees/ebxml-cppa/schema/cpp-cpa-negot-2_0.xsd"
xmlns:tns="http://www.oasis-open.org/committees/ebxml-cppa/schema/cpp-cpa-negot-2_0.xsd"
xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns="http://www.w3.org/2001/XMLSchema"
elementFormDefault="qualified" attributeFormDefault="unqualified">
    <element name="NegotiationDescriptor">
        <annotation>
            <documentation>This is the schema representing the NDD</documentation>
        </annotation>
        <complexType>
            <sequence>
                <element name="NegotiableInformationItem" maxOccurs="unbounded">
                    <complexType>
                        <sequence>
                            <choice>
                                <element name="Value" type="tns:ValueType"/>
                                <element name="UnOrderedValue" type="tns:EnumeratedValues"/>
                                <element name="OrderedValue" type="tns:OrderedEnumeratedValues"/>
                                <element name="ValueWithPreferenceMeasure"
type="tns:ValueWithPreferenceMeasureType"/>
                                <element name="PresentOrNot" type="tns:PresentOrNotType"/>
                                <element name="IntegerValues" type="tns:IntegerValuesType"/>
                                <element name="Preference">
                                    <complexType>
                                        <attribute name="value" type="xs:integer"/>
                                    </complexType>
                                </element>
                                <element name="Cardinality" type="tns:IntegerValuesType"/>
                                <element name="BooleanValue" type="tns:BooleanValuesType"/>
                                <element name="DurationWithPreference" type="tns:DurationWithPreferenceType"/>
                            </choice>
                        </sequence>
                        <attribute name="xpath" type="xs:string" use="required"/>
                    </complexType>
                </element>
            </sequence>
            <attribute name="documentLocation" type="xs:anyURI" use="required"/>
        </complexType>
    </element>
    <!--TYPE DEFINITIONS THAT ARE USED IN THE DEFINITIONS ABOVE-->
    <complexType name="ValueType">
        <attribute name="mustBeFilledIn" use="optional">
            <simpleType name="fillType">
                <restriction base="xs:string">
                    <enumeration value="Yes"/>
                    <enumeration value="No"/>
                    <enumeration value="Preferred"/>
                </restriction>
            </simpleType>
        </attribute>
    </complexType>
    <complexType name="EnumeratedValues">
        <sequence>
            <element name="PresentOrNot" type="tns:PresentOrNotType" minOccurs="0"/>
            <element name="Value" type="xs:string" maxOccurs="unbounded"/>
        </sequence>
    </complexType>
    <!--For also stating that the enumerted values have some order associated with them-->
    <complexType name="OrderedEnumeratedValues">
        <complexContent>
            <extension base="tns:EnumeratedValues">
                <attribute name="preference" use="optional">
                    <simpleType name="orderName">
                        <restriction base="xs:string">
                            <enumeration value="EarlierPreferred"/>
                            <enumeration value="LaterPreferred"/>
                        </restriction>
                    </simpleType>
                </attribute>
            </extension>
        </complexContent>
    </complexType>
    <!--For giving the endpoints for elements like Start/End. The type is currently set to string beacuse XML spy does not
seem to validate dateTime entries correctly, but should be changed to dateTime later -->
    <complexType name="DateEndPointsType">
        <sequence>
            <element name="EarliestStart" type="xs:dateTime"/>
            <element name="LatestEnd" type="xs:dateTime"/>
        </sequence>
    </complexType>
    <!--For giving a type of preference function. Currently, the preference function can be one of two: (a) a piecewise linear
function,  (b) an arbtrary function expressed as a string (such as x*x+ y*y) .-->
    <complexType name="PreferenceFunctionType">
        <sequence>
            <choice>
                <element name="PiecewiseLinearPiece" maxOccurs="unbounded">
                    <complexType>
                        <sequence>
                            <element name="x-coordinate1" type="xs:string"/>
                            <element name="y-coordinate1" type="xs:NMTOKEN"/>
                            <element name="x-coordinate2" type="xs:string"/>
                            <element name="y-coordinate2" type="xs:NMTOKEN"/>
                            <!--TODO: Make dateTime/NMTOKEN for generality?-->
                        </sequence>
                    </complexType>
                </element>
                <element name="FunctionDefinedByEquation" type="xs:string"/>
                <!--<xs:element name="NoPreference"/> -->
            </choice>
        </sequence>
    </complexType>
    <!--For giving a value, and associating with it some preferrence function -->
    <complexType name="ValueWithPreferenceMeasureType">
        <sequence>
            <element name="EndPoints" type="tns:DateEndPointsType"/>
            <element name="PreferenceFunction" type="tns:PreferenceFunctionType" minOccurs="0"/>
        </sequence>
    </complexType>
    <!--For specifying a preferrence for whether an entry must be present or not -->
    <complexType name="PresentOrNotType">
        <attribute name="value" use="required">
            <simpleType>
                <restriction base="xs:string">
                    <enumeration value="MustBePresent"/>
                    <enumeration value="MustBeAbsent"/>
                    <enumeration value="PreferredPresent"/>
                    <enumeration value="PreferredAbsent"/>
                    <enumeration value="Agnostic"/>
                </restriction>
            </simpleType>
        </attribute>
    </complexType>
    <!--For giving the integer endpoints for elements like retries.  -->
    <complexType name="IntegerEndPointsType">
        <sequence>
            <element name="SmallestValue" type="xs:integer"/>
            <element name="LatestValue" type="xs:integer"/>
        </sequence>
    </complexType>
    <!--This associates a preference order to the integer end points or a preferrence such as Smaller is Preferred-->
    <complexType name="IntegerValuesWithPreferenceMeasureType">
        <sequence>
            <element name="EndPoints" type="tns:IntegerEndPointsType"/>
            <element name="PreferenceFunction" type="tns:PreferenceFunctionType" minOccurs="0"/>
        </sequence>
        <attribute name="preferenceOrder">
            <simpleType name="orderName">
                <restriction base="xs:string">
                    <enumeration value="SmallerPreferred"/>
                    <enumeration value="LargerPreferred"/>
                </restriction>
            </simpleType>
        </attribute>
    </complexType>
    <!--This type is for integer values whose (a) presence can be potentially negotiated (b) the values themselves can be
negotiated -->
    <complexType name="IntegerValuesType">
        <sequence>
            <element name="PresentOrNot" type="tns:PresentOrNotType" minOccurs="0"/>
            <element name="RangeInfo" type="tns:IntegerValuesWithPreferenceMeasureType"/>
        </sequence>
    </complexType>
    <!--For specifying a preferrence for whether an entry must be present or not -->
    <complexType name="BooleanValuesType">
        <sequence>
            <element name="PresentOrNot" type="tns:PresentOrNotType" minOccurs="0"/>
        </sequence>
        <attribute name="preference" use="required">
            <simpleType>
                <restriction base="xs:string">
                    <enumeration value="TruePreferred"/>
                    <enumeration value="FalsePreferred"/>
                    <enumeration value="Agnostic"/>
                </restriction>
            </simpleType>
        </attribute>
    </complexType>
    <!--For specifying the minimum and maximum allowable durations-->
    <complexType name="DurationType">
        <sequence>
            <element name="PresentOrNot" type="tns:PresentOrNotType" minOccurs="0"/>
            <element name="MinimumDuration" type="xs:duration"/>
            <element name="MaximumDuration" type="xs:duration"/>
        </sequence>
    </complexType>
    <!--For also stating that the durations have some preference  associated with them-->
    <complexType name="DurationWithPreferenceType">
        <complexContent>
            <extension base="tns:DurationType">
                <attribute name="preferenceOrder" use="optional">
                    <simpleType name="preferenceName">
                        <restriction base="xs:string">
```

```
                            <enumeration value="SmallerPreferred"/>
                            <enumeration value="LargerPreferred"/>
                        </restriction>
                    </simpleType>
                </attribute>
            </extension>
        </complexContent>
    </complexType>
</schema>
```

Appendix B XML Schema for Negotiation Messages (Normative)

The XML Schema for the negotiation *Messages* is available in text form at:

THIS SECTION HAS BEEN REPLACED IN FULL.

```
<?xml version="1.0" encoding="UTF-8"?>
<!--  This is the schema that corresponds to the version 1.0 CPP/A Negotiation spec
  -->
<schema targetNamespace="http://www.oasis-open.org/committees/ebxml-cppa/schema/cpa-negot-1_0.xsd"
xmlns:tns="http://www.oasis-open.org/committees/ebxml-cppa/schema/cpa-negot-1_0.xsd" xmlns:cppa="http://www.oasis-
open.org/committees/ebxml-cppa/schema/cpp-cpa-2_0.xsd" xmlns="http://www.w3.org/2001/XMLSchema">
    <import namespace="http://www.oasis-open.org/committees/ebxml-cppa/schema/cpp-cpa-2_0.xsd"
schemaLocation="http://www.oasis-open.org/committees/ebxml-cppa/schema/cpp-cpa-2_0.xsd"/>
    <attributeGroup name="id.grp">
        <attribute name="id" type="cppa:non-empty-string" use="required"/>
        <attribute ref="cppa:version" use="required"/>
    </attributeGroup>
    <element name="NegotiationMessage">
        <complexType>
            <sequence>
                <element ref="tns:NCPA"/>
                <element ref="tns:CPATemplateId"/>
                <element ref="cppa:SecurityDetails"/>
                <element ref="tns:InitiatingParty"/>
                <element ref="tns:RespondingParty"/>
                <element ref="tns:BPSSBusinessDocumentName"/>
                <element name="ExpirationDate" type="dateTime"/>
                <element ref="tns:BusinessDocuments"/>
                <element ref="tns:NegotiationContent"/>
                <element name="ResponseToURL" type="anyURI" minOccurs="0"/>
                <element name="Offer-CounterOffer-Acceptance-Time" type="dateTime" minOccurs="0"/>
                <element name="Comment" type="string" minOccurs="0"/>
            </sequence>
            <attribute name="businessMsgId" type="ID" use="required"/>
            <attribute name="negotiationDialogId" type="cppa:non-empty-string" use="required"/>
            <attribute name="offerId" type="cppa:non-empty-string"/>
            <attribute name="inresponseTo" type="cppa:non-empty-string"/>
            <attribute name="binding" type="boolean" use="required"/>
            <attribute name="messageType" type="tns:messageTypeValue.type" use="required"/>
            <attribute name="error" type="tns:errorValue.type"/>
        </complexType>
    </element>
    <element name="CPATemplateId">
        <complexType>
            <attributeGroup ref="tns:id.grp"/>
        </complexType>
    </element>
    <element name="NCPA">
        <complexType>
            <attribute name="uri" type="anyURI" use="required"/>
        </complexType>
    </element>
    <element name="BPSSBusinessDocumentName">
        <complexType>
            <attribute name="name" type="tns:bpssBusinessDocumentName.type" use="required"/>
        </complexType>
    </element>
    <element name="BusinessDocuments">
        <complexType>
            <choice>
```

```
                <element name="CPATemplateDoc" type="tns:CPATemplateDoc.type"/>
            </choice>
        </complexType>
    </element>
    <element name="NegotiationContent">
        <complexType>
            <sequence>
                <element ref="tns:AcceptedItem" maxOccurs="unbounded"/>
                <element ref="tns:DeletedItem" minOccurs="0" maxOccurs="unbounded"/>
                <element ref="tns:UpdatedItem" minOccurs="0" maxOccurs="unbounded"/>
                <element ref="tns:InsertedItem" minOccurs="0" maxOccurs="unbounded"/>
            </sequence>
        </complexType>
    </element>
    <element name="AcceptedItem" type="tns:simpleItem.type"/>
    <element name="DeletedItem" type="tns:simpleItem.type"/>
    <element name="UpdatedItem" type="tns:updatedItem.type"/>
    <element name="InsertedItem" type="tns:insertedItem.type"/>
    <element name="InitiatingParty" type="tns:partySummary.type"/>
    <element name="RespondingParty" type="tns:partySummary.type"/>
    <complexType name="simpleItem.type">
        <attribute name="xpath" type="tns:xpath.type" use="required"/>
    </complexType>
    <complexType name="updatedItem.type">
        <attribute name="xpath" type="tns:xpath.type" use="required"/>
        <attribute name="originalValue" type="cppa:non-empty-string" use="required"/>
        <attribute name="proposedValue" type="cppa:non-empty-string" use="required"/>
        <attribute name="itemstatus" type="tns:itemstatus.type" use="required"/>
    </complexType>
    <complexType name="insertedItem.type">
        <attribute name="xpath" type="tns:xpath.type" use="required"/>
        <attribute name="proposedValue" type="cppa:non-empty-string" use="required"/>
        <attribute name="itemstatus" type="tns:itemstatus.type" use="required"/>
    </complexType>
    <complexType name="doc.type">
        <choice>
            <element name="BinaryDoc" type="base64Binary"/>
            <element name="Uri" type="anyURI"/>
        </choice>
    </complexType>
    <complexType name="CPATemplateDoc.type">
        <sequence>
            <element name="NDD" type="tns:doc.type"/>
            <element name="CPATemplate" type="tns:doc.type"/>
        </sequence>
    </complexType>
    <complexType name="partySummary.type">
        <sequence>
            <element ref="cppa:PartyId"/>
            <element name="CPPId">
                <complexType>
                    <attributeGroup ref="tns:id.grp"/>
                </complexType>
            </element>
            <element name="CPPNDD" type="tns:doc.type" minOccurs="0"/>
        </sequence>
    </complexType>
    <simpleType name="xpath.type">
        <restriction base="string"/>
    </simpleType>
    <simpleType name="itemstatus.type">
        <restriction base="NMTOKEN">
            <enumeration value="Required"/>
            <enumeration value="Preferred"/>
        </restriction>
    </simpleType>
    <simpleType name="messageTypeValue.type">
        <restriction base="NMTOKEN">
            <enumeration value="Offer"/>
            <enumeration value="CounterOffer"/>
            <enumeration value="CounterOfferPending"/>
            <enumeration value="Rejected"/>
            <enumeration value="Accepted"/>
            <enumeration value="Expired"/>
            <enumeration value="SinglePartySigned"/>
            <enumeration value="Signed"/>
            <enumeration value="Unsigned"/>
        </restriction>
    </simpleType>
    <simpleType name="errorValue.type">
        <restriction base="NMTOKEN">
            <enumeration value="ExpiredCPP"/>
            <enumeration value="UnableToFullfillSecurityRequirements"/>
            <enumeration value="ProposedSecurityPolicyInadequate"/>
            <enumeration value="OutOfSequenceCounterOffer"/>
            <enumeration value="FaildSignatureValidation_CPATemplate"/>
            <enumeration value="FailedSignatureValidation_CPA"/>
            <enumeration value="UnsupportedBusinessRelationship"/>
            <enumeration value="UnsupportedPackaging"/>
            <enumeration value="UnsupportedSignal"/>
            <enumeration value="FailedToConverge"/>
            <enumeration value="PreviouslyRejectedCPA"/>
            <enumeration value="ExpiredOffer"/>
            <enumeration value="FormatError"/>
            <enumeration value="UnknownSystemError"/>
        </restriction>
    </simpleType>
    <simpleType name="bpssBusinessDocumentName.type">
        <restriction base="NMTOKEN">
            <enumeration value="CPA_Offer_Doc"/>
            <enumeration value="CPA_Accept_Offer_Doc"/>
            <enumeration value="CPA_Counter_Pending_Offer_Doc"/>
            <enumeration value="CPA_Counter_Offer_Doc"/>
            <enumeration value="CPA_Reject_Offer_Doc"/>
            <enumeration value="CPA_Final_Doc"/>
            <enumeration value="CPA_Final_Response_Doc"/>
            <enumeration value="CPA_Final_Response_Doc_Signed"/>
            <enumeration value="CPA_Final_Response_Reject_Doc"/>
        </restriction>
    </simpleType>
</schema>
```

Appendix C Negotiation CPA Example (Non-Normative)

The text file for this *NCPA* example is available at:

THE NCPA'S PACKAGING DEFINITIONS HAVE TO BE COMPLETED NOW THAT, THE NDD AND MESSAGE SCHEMA ARE COMPLETED. OF PARTICULAR CONCERN ARE THE PACKAGING DEFINITION AND CHANGES THAT MAY BE NEEDED TO MATCH THE CHANGES IN THE NEGOTIATION BPSS WITH REGARD TO THE CONCLUSION OF NEGOTIATION.

Deleted: ***AFTER***

```
<?xml version="1.0"?>
<!-- edited with XML Spy v4.4 U (http://www.xmlspy.com) by Hima Mukkamala (Web Services Architecture WG) -->
<tp:CollaborationProtocolAgreement xmlns:tp="http://www.oasis-open.org/committees/ebxml-cppa/schema/cpp-cpa-
2_0.xsd" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xlink="http://www.w3.org/1999/xlink"
xmlns:ds="http://www.w3.org/2000/09/xmldsig#" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xsi:schemaLocation="http://www.oasis-open.org/committees/ebxml-cppa/schema/cpp-cpa-2_0.xsd
              cpp-cpa-2_0.xsd" tp:cpaid="uri:NegoInit-and-NegoResp-cpa" tp:version="2_0a">
    <tp:Status tp:value="proposed"/>
    <tp:Start>2001-05-20T07:21:00Z</tp:Start>
    <tp:End>2003-05-20T07:21:00Z</tp:End>
    <tp:ConversationConstraints tp:invocationLimit="100" tp:concurrentConversations="10"/>
    <!-- Party info for Negotiation Initiator -->
    <tp:PartyInfo tp:partyName="NegotiationInitiator" tp:defaultMshChannelId="asyncChannelA1"
tp:defaultMshPackageId="NegoInit_MshSignalPackage">
        <tp:PartyId tp:type="urn:oasis:names:tc:ebxml-cppa:partyid-type:duns">123456789</tp:PartyId>
        <tp:PartyRef xlink:href="http://NegoInit.com/about.html"/>
        <!-- This role is for Negotiation Initiator performing the role of Negotiation Initiator -->
        <tp:CollaborationRole>
            <tp:ProcessSpecification tp:version="2.0" tp:name="CPPA-Negotiation" xlink:type="simple"
xlink:href="http://www.oasis-open.org/committees/ebxml-cppa-negot/CPA_Negotiation_BPSS.xml"
tp:uuid="bpid:ebXML:CPPA-Negotiation"/>
            <tp:Role tp:name="CPA Negotiation Initiator" xlink:type="simple" xlink:href="http://www.oasis-
open.org/committees/ebxml-cppa-negot/CPA_Negotiation_BPSS.xm#CPA Negotiation Initiator"/>
            <tp:ServiceBinding>
                <tp:Service>bpid:ebXML:CPPA-Negotiation</tp:Service>
                <!-- This send is for sending the Negotiation Offer -->
                <tp:CanSend>
                    <tp:ThisPartyActionBinding tp:id="NegoInit_ABID1" tp:action="CPA_Offer_BT_ReqBA"
tp:packageId="NegoInit_OfferRequestPackage">
                        <tp:BusinessTransactionCharacteristics tp:isNonRepudiationRequired="false"
tp:isNonRepudiationReceiptRequired="false" tp:isConfidential="none" tp:isAuthenticated="none" tp:isTamperProof="none"
tp:isAuthorizationRequired="false" tp:timeToAcknowledgeReceipt="PT2H" tp:timeToPerform="P1D"/>
                        <tp:ActionContext tp:binaryCollaboration="CPA Negotiation BC"
tp:businessTransactionActivity="CPA Offer BTA" tp:requestOrResponseAction="CPA_Offer_BT_ReqBA"/>
                        <tp:ChannelId>asyncChannelA1</tp:ChannelId>
                    </tp:ThisPartyActionBinding>
                    <tp:OtherPartyActionBinding>NegoResp_ABID1</tp:OtherPartyActionBinding>
                </tp:CanSend>
                <!-- This send is for sending the Receipt Acknowledgment -->
                <tp:CanSend>
                    <tp:ThisPartyActionBinding tp:id="NegoInit_ABID2" tp:action="ReceiptAcknowledgement"
tp:packageId="NegoInit_ReceiptAcknowledgmentPackage">
                        <tp:BusinessTransactionCharacteristics tp:isNonRepudiationRequired="true"
tp:isNonRepudiationReceiptRequired="true" tp:isConfidential="transient" tp:isAuthenticated="persistent"
tp:isTamperProof="persistent" tp:isAuthorizationRequired="true"/>
                        <tp:ChannelId>asyncChannelA1</tp:ChannelId>
                    </tp:ThisPartyActionBinding>
                    <tp:OtherPartyActionBinding>NegoResp_ABID2</tp:OtherPartyActionBinding>
                </tp:CanSend>
                <!-- This send is for send the Final message in the collaboration. This would be the double signed CPA
```

```
document or acceptance or reject of the CPA  in the final Response document-->
        <tp:CanSend>
          <tp:ThisPartyActionBinding tp:id="NegoInit_FinalResponseMessageA"
tp:action="Final_CPA_BT_RespBA" tp:packageId="NegoInit_FinalMessage">
            <tp:BusinessTransactionCharacteristics tp:isNonRepudiationRequired="false"
tp:isNonRepudiationReceiptRequired="false" tp:isConfidential="none" tp:isAuthenticated="none" tp:isTamperProof="none"
tp:isAuthorizationRequired="false" tp:timeToAcknowledgeReceipt="PT2H" tp:timeToPerform="P1D"/>
            <tp:ActionContext tp:binaryCollaboration="CPA Negotiation BC"
tp:businessTransactionActivity="CPA Final BTA" tp:requestOrResponseAction="Final_CPA_BT_RespBA"/>
            <tp:ChannelId>asyncChannelA1</tp:ChannelId>
          </tp:ThisPartyActionBinding>
          <tp:OtherPartyActionBinding>NegoResp_FinalResponseMessageB</tp:OtherPartyActionBinding>
        </tp:CanSend>
        <!-- This receive is for receving the response for Negotiation Offer, could be acceptance, reject or counter
offer-->
        <tp:CanReceive>
          <tp:ThisPartyActionBinding tp:id="NegoInit_ABID9" tp:action="CPA_Offer_BT_RespBA"
tp:packageId="NegoInit_OfferResponsePackage">
            <tp:BusinessTransactionCharacteristics tp:isNonRepudiationRequired="false"
tp:isNonRepudiationReceiptRequired="false" tp:isConfidential="none" tp:isAuthenticated="none" tp:isTamperProof="none"
tp:isAuthorizationRequired="false" tp:timeToAcknowledgeReceipt="PT2H" tp:timeToPerform="P1D"/>
            <tp:ActionContext tp:binaryCollaboration="CPA Negotiation BC"
tp:businessTransactionActivity="CPA Offer BTA" tp:requestOrResponseAction="CPA_Offer_BT_RespBA"/>
            <tp:ChannelId>asyncChannelA1</tp:ChannelId>
          </tp:ThisPartyActionBinding>
          <tp:OtherPartyActionBinding>NegoResp_ABID9</tp:OtherPartyActionBinding>
        </tp:CanReceive>
        <!-- This receive is for receving the Final Response document in the final BTA -->
        <tp:CanReceive>
          <tp:ThisPartyActionBinding tp:id="NegoInit_FinalResponseA" tp:action="Final_CPA_BT_ReqBA"
tp:packageId="NegoInit_FinalMessage">
            <tp:BusinessTransactionCharacteristics tp:isNonRepudiationRequired="false"
tp:isNonRepudiationReceiptRequired="false" tp:isConfidential="none" tp:isAuthenticated="none" tp:isTamperProof="none"
tp:isAuthorizationRequired="false" tp:timeToAcknowledgeReceipt="PT2H" tp:timeToPerform="P1D"/>
            <tp:ActionContext tp:binaryCollaboration="CPA Negotiation BC"
tp:businessTransactionActivity="CPA Final BTA" tp:requestOrResponseAction="Final_CPA_BT_ReqBA"/>
            <tp:ChannelId>asyncChannelA1</tp:ChannelId>
          </tp:ThisPartyActionBinding>
          <tp:OtherPartyActionBinding>NegoResp_FinalResponseB</tp:OtherPartyActionBinding>
        </tp:CanReceive>
        <!-- This Receive is for receiving the Receipt Acknowledgment -->
        <tp:CanReceive>
          <tp:ThisPartyActionBinding tp:id="NegoInit_ABID3" tp:action="ReceiptAcknowledgment"
tp:packageId="NegoInit_ReceiptAcknowledgmentPackage">
            <tp:BusinessTransactionCharacteristics tp:isNonRepudiationRequired="true"
tp:isNonRepudiationReceiptRequired="true" tp:isConfidential="transient" tp:isAuthenticated="persistent"
tp:isTamperProof="persistent" tp:isAuthorizationRequired="true"/>
            <tp:ChannelId>asyncChannelA1</tp:ChannelId>
          </tp:ThisPartyActionBinding>
          <tp:OtherPartyActionBinding>NegoResp_ABID3</tp:OtherPartyActionBinding>
        </tp:CanReceive>
        <!-- This Receive is for receiving the Exception -->
        <tp:CanReceive>
          <tp:ThisPartyActionBinding tp:id="NegoInit_ABID4" tp:action="Exception"
tp:packageId="NegoInit_ExceptionPackage">
            <tp:BusinessTransactionCharacteristics tp:isNonRepudiationRequired="true"
tp:isNonRepudiationReceiptRequired="true" tp:isConfidential="transient" tp:isAuthenticated="persistent"
tp:isTamperProof="persistent" tp:isAuthorizationRequired="true"/>
            <tp:ChannelId>asyncChannelA1</tp:ChannelId>
          </tp:ThisPartyActionBinding>
          <tp:OtherPartyActionBinding>NegoResp_ABID4</tp:OtherPartyActionBinding>
        </tp:CanReceive>
      </tp:ServiceBinding>
    </tp:CollaborationRole>
    <!-- This role is for Negotiation Initiator company performing the role of Negotiation Counter offer responder -->
    <tp:CollaborationRole>
```

```
      <tp:ProcessSpecification tp:version="2.0" tp:name="CPPA-Negotiation" xlink:type="simple"
xlink:href="http://www.oasis-open.org/committees/ebxml-cppa-negot/CPA_Negotiation_BPSS.xml"
tp:uuid="bpid:ebXML:CPPA-Negotiation"/>
      <tp:Role tp:name="CPA Negotiation Counter Offer Responder" xlink:type="simple" xlink:href="http://www.oasis-
open.org/committees/ebxml-cppa-negot/CPA_Negotiation_BPSS.xm#CPA Negotiation Counter Offer Responder"/>
      <tp:ServiceBinding>
        <tp:Service>bpid:ebXML:CPPA-Negotiation</tp:Service>
        <!-- This send is for sending the Negotiation Counter Offer in "CPA Counter Offer 2 BTA"-->
        <tp:CanSend>
          <tp:ThisPartyActionBinding tp:id="NegoInit_ABID5" tp:action="CPA_Counter_Offer_BT_ReqBA"
tp:packageId="NegoInit_CounterOfferRequestPackage">
            <tp:BusinessTransactionCharacteristics tp:isNonRepudiationRequired="false"
tp:isNonRepudiationReceiptRequired="false" tp:isConfidential="none" tp:isAuthenticated="none" tp:isTamperProof="none"
tp:isAuthorizationRequired="false" tp:timeToAcknowledgeReceipt="PT2H" tp:timeToPerform="P1D"/>
            <tp:ActionContext tp:binaryCollaboration="CPA Negotiation BC"
tp:businessTransactionActivity="CPA Counter Offer 2 BTA"
tp:requestOrResponseAction="CPA_Counter_Offer_BT_ReqBA">
              <tp:CollaborationActivity tp:name="CPA Counter Offer CA"/>
            </tp:ActionContext>
            <tp:ChannelId>asyncChannelA1</tp:ChannelId>
          </tp:ThisPartyActionBinding>
          <tp:OtherPartyActionBinding>NegoResp_ABID5</tp:OtherPartyActionBinding>
        </tp:CanSend>
        <!-- This send is for sending the Negotiation Counter Offer Response  in "CPA Counter Offer 1 BTA"-->
        <tp:CanSend>
          <tp:ThisPartyActionBinding tp:id="NegoInit_ABID6" tp:action="CPA_Counter_Offer_BT_ReqBA"
tp:packageId="NegoInit_CounterOfferResponsePackage">
            <tp:BusinessTransactionCharacteristics tp:isNonRepudiationRequired="false"
tp:isNonRepudiationReceiptRequired="false" tp:isConfidential="none" tp:isAuthenticated="none" tp:isTamperProof="none"
tp:isAuthorizationRequired="false" tp:timeToAcknowledgeReceipt="PT2H" tp:timeToPerform="P1D"/>
            <tp:ActionContext tp:binaryCollaboration="CPA Negotiation BC"
tp:businessTransactionActivity="CPA Counter Offer 1 BTA"
tp:requestOrResponseAction="CPA_Counter_Offer_BT_RespBA">
              <tp:CollaborationActivity tp:name="CPA Counter Offer CA"/>
            </tp:ActionContext>
            <tp:ChannelId>asyncChannelA1</tp:ChannelId>
          </tp:ThisPartyActionBinding>
          <tp:OtherPartyActionBinding>NegoResp_ABID6</tp:OtherPartyActionBinding>
        </tp:CanSend>
        <!-- This send is for sending the Final CPA Response message in CPA_Final_BTA_init_Responder"-->
        <tp:CanSend>
          <tp:ThisPartyActionBinding tp:id="NegoCOR_FinalMessageA" tp:action="Final_CPA_BT_ReqBA"
tp:packageId="NegoInit_FinalMessage">
            <tp:BusinessTransactionCharacteristics tp:isNonRepudiationRequired="false"
tp:isNonRepudiationReceiptRequired="false" tp:isConfidential="none" tp:isAuthenticated="none" tp:isTamperProof="none"
tp:isAuthorizationRequired="false" tp:timeToAcknowledgeReceipt="PT2H" tp:timeToPerform="P1D"/>
            <tp:ActionContext tp:binaryCollaboration="CPA Negotiation BC"
tp:businessTransactionActivity="CPA_Final_BTA_init_Responder" tp:requestOrResponseAction="Final_CPA_BT_ReqBA">
              <tp:CollaborationActivity tp:name="CPA Counter Offer CA"/>
            </tp:ActionContext>
            <tp:ChannelId>asyncChannelA1</tp:ChannelId>
          </tp:ThisPartyActionBinding>
          <tp:OtherPartyActionBinding>NegoCOResp_FinalMessageB</tp:OtherPartyActionBinding>
        </tp:CanSend>
        <!-- This send is for sending the response to the Final CPA Response message in
CPA_Final_BTA_init_Initiator"-->
        <tp:CanSend>
          <tp:ThisPartyActionBinding tp:id="NegoCOR_FinalMessageResponseA"
tp:action="Final_CPA_BT_RespBA" tp:packageId="NegoInit_FinalMessage">
            <tp:BusinessTransactionCharacteristics tp:isNonRepudiationRequired="false"
tp:isNonRepudiationReceiptRequired="false" tp:isConfidential="none" tp:isAuthenticated="none" tp:isTamperProof="none"
tp:isAuthorizationRequired="false" tp:timeToAcknowledgeReceipt="PT2H" tp:timeToPerform="P1D"/>
            <tp:ActionContext tp:binaryCollaboration="CPA Negotiation BC"
tp:businessTransactionActivity="CPA_Final_BTA_init_Initiator" tp:requestOrResponseAction="Final_CPA_BT_RespBA">
              <tp:CollaborationActivity tp:name="CPA Counter Offer CA"/>
            </tp:ActionContext>
            <tp:ChannelId>asyncChannelA1</tp:ChannelId>
          </tp:ThisPartyActionBinding>
          <tp:OtherPartyActionBinding>NegoCOResp_FinalMessageResponseB</tp:OtherPartyActionBinding>
        </tp:CanSend>
        <!-- This send is for sending the Receipt Acknowledgment -->
        <tp:CanSend>
          <tp:ThisPartyActionBinding tp:id="NegoInit_ABID13" tp:action="ReceiptAcknowledgement"
tp:packageId="NegoInit_ReceiptAcknowledgmentPackage">
            <tp:BusinessTransactionCharacteristics tp:isNonRepudiationRequired="true"
tp:isNonRepudiationReceiptRequired="true" tp:isConfidential="transient" tp:isAuthenticated="persistent"
tp:isTamperProof="persistent" tp:isAuthorizationRequired="true"/>
            <tp:ChannelId>asyncChannelA1</tp:ChannelId>
          </tp:ThisPartyActionBinding>
          <tp:OtherPartyActionBinding>NegoResp_ABID13</tp:OtherPartyActionBinding>
        </tp:CanSend>
        <!-- This receive is for receiving the Final CPA message in CPA_Final_BTA_init_Initiator"-->
        <tp:CanReceive>
          <tp:ThisPartyActionBinding tp:id="NegoCOR_FinalMessageA1" tp:action="Final_CPA_BT_ReqBA"
tp:packageId="NegoInit_FinalMessage">
            <tp:BusinessTransactionCharacteristics tp:isNonRepudiationRequired="false"
tp:isNonRepudiationReceiptRequired="false" tp:isConfidential="none" tp:isAuthenticated="none" tp:isTamperProof="none"
tp:isAuthorizationRequired="false" tp:timeToAcknowledgeReceipt="PT2H" tp:timeToPerform="P1D"/>
            <tp:ActionContext tp:binaryCollaboration="CPA Negotiation BC"
tp:businessTransactionActivity="CPA_Final_BTA_init_Initiator" tp:requestOrResponseAction="Final_CPA_BT_ReqBA">
              <tp:CollaborationActivity tp:name="CPA Counter Offer CA"/>
            </tp:ActionContext>
            <tp:ChannelId>asyncChannelA1</tp:ChannelId>
          </tp:ThisPartyActionBinding>
          <tp:OtherPartyActionBinding>NegoCOResp_FinalMessageB1</tp:OtherPartyActionBinding>
        </tp:CanReceive>
        <!-- This receive is for receiving the response to the Final CPA message in
CPA_Final_BTA_init_Responder"-->
        <tp:CanReceive>
          <tp:ThisPartyActionBinding tp:id="NegoCOR_FinalMessageResponseA2"
tp:action="Final_CPA_BT_RespBA" tp:packageId="NegoInit_FinalMessage">
            <tp:BusinessTransactionCharacteristics tp:isNonRepudiationRequired="false"
tp:isNonRepudiationReceiptRequired="false" tp:isConfidential="none" tp:isAuthenticated="none" tp:isTamperProof="none"
tp:isAuthorizationRequired="false" tp:timeToAcknowledgeReceipt="PT2H" tp:timeToPerform="P1D"/>
            <tp:ActionContext tp:binaryCollaboration="CPA Negotiation BC"
tp:businessTransactionActivity="CPA_Final_BTA_init_Responder"
tp:requestOrResponseAction="Final_CPA_BT_RespBA">
              <tp:CollaborationActivity tp:name="CPA Counter Offer CA"/>
            </tp:ActionContext>
            <tp:ChannelId>asyncChannelA1</tp:ChannelId>
          </tp:ThisPartyActionBinding>
          <tp:OtherPartyActionBinding>NegoCOResp_FinalMessageResponseB2</tp:OtherPartyActionBinding>
        </tp:CanReceive>
        <!-- This receive is for receiving the response forNegotiation Counter Offer, could be accept, reject or again
send a counter offer This happens in "CPA Counter Offer 2 BTA"-->
        <tp:CanReceive>
          <tp:ThisPartyActionBinding tp:id="NegoInit_ABID10" tp:action="CPA_Counter_Offer_BT_RespBA"
tp:packageId="NegoInit_CounterOfferResponsePackage">
            <tp:BusinessTransactionCharacteristics tp:isNonRepudiationRequired="false"
tp:isNonRepudiationReceiptRequired="false" tp:isConfidential="none" tp:isAuthenticated="none" tp:isTamperProof="none"
tp:isAuthorizationRequired="false" tp:timeToAcknowledgeReceipt="PT2H" tp:timeToPerform="P1D"/>
            <tp:ActionContext tp:binaryCollaboration="CPA Negotiation BC"
tp:businessTransactionActivity="CPA Counter Offer 2 BTA"
tp:requestOrResponseAction="CPA_Counter_Offer_BT_RespBA">
              <tp:CollaborationActivity tp:name="CPA Counter Offer CA"/>
            </tp:ActionContext>
            <tp:ChannelId>asyncChannelA1</tp:ChannelId>
          </tp:ThisPartyActionBinding>
          <tp:OtherPartyActionBinding>NegoResp_ABID10</tp:OtherPartyActionBinding>
        </tp:CanReceive>
        <!-- This receive is for receiving the Negotiation Counter Offer. This happens in "CPA Counter Offer 1 BTA"--
>
```

```
                <tp:CanReceive>
                    <tp:ThisPartyActionBinding tp:id="NegoInit_ABID12" tp:action="CPA_Counter_Offer_BT_RespBA"
tp:packageId="NegoInit_CounterOfferRequestPackage">
                        <tp:BusinessTransactionCharacteristics tp:isNonRepudiationRequired="false"
tp:isNonRepudiationReceiptRequired="false" tp:isConfidential="none" tp:isAuthenticated="none" tp:isTamperProof="none"
tp:isAuthorizationRequired="false" tp:timeToAcknowledgeReceipt="PT2H" tp:timeToPerform="P1D"/>
                        <tp:ActionContext tp:binaryCollaboration="CPA Negotiation BC"
tp:businessTransactionActivity="CPA Counter Offer 1 BTA"
tp:requestOrResponseAction="CPA_Counter_Offer_BT_ReqBA">
                            <tp:CollaborationActivity tp:name="CPA Counter Offer CA"/>
                        </tp:ActionContext>
                        <tp:ChannelId>asyncChannelA1</tp:ChannelId>
                    </tp:ThisPartyActionBinding>
                    <tp:OtherPartyActionBinding>NegoResp_ABID12</tp:OtherPartyActionBinding>
                </tp:CanReceive>
                <!-- This Receive is for receiving the Receipt Acknowledgment -->
                <tp:CanReceive>
                    <tp:ThisPartyActionBinding tp:id="NegoInit_ABID7" tp:action="ReceiptAcknowledgment"
tp:packageId="NegoInit_ReceiptAcknowledgmentPackage">
                        <tp:BusinessTransactionCharacteristics tp:isNonRepudiationRequired="true"
tp:isNonRepudiationReceiptRequired="true" tp:isConfidential="transient" tp:isAuthenticated="persistent"
tp:isTamperProof="persistent" tp:isAuthorizationRequired="true"/>
                        <tp:ChannelId>asyncChannelA1</tp:ChannelId>
                    </tp:ThisPartyActionBinding>
                    <tp:OtherPartyActionBinding>NegoResp_ABID7</tp:OtherPartyActionBinding>
                </tp:CanReceive>
                <!-- This Receive is for receiving the Exception -->
                <tp:CanReceive>
                    <tp:ThisPartyActionBinding tp:id="NegoInit_ABID8" tp:action="Exception"
tp:packageId="NegoInit_ExceptionPackage">
                        <tp:BusinessTransactionCharacteristics tp:isNonRepudiationRequired="true"
tp:isNonRepudiationReceiptRequired="true" tp:isConfidential="transient" tp:isAuthenticated="persistent"
tp:isTamperProof="persistent" tp:isAuthorizationRequired="true"/>
                        <tp:ChannelId>asyncChannelA1</tp:ChannelId>
                    </tp:ThisPartyActionBinding>
                    <tp:OtherPartyActionBinding>NegoResp_ABID8</tp:OtherPartyActionBinding>
                </tp:CanReceive>
            </tp:ServiceBinding>
        </tp:CollaborationRole>
        <!-- Certificates used by the "Negotiation Initiator" company -->
        <tp:Certificate tp:certId="NegoInit_AppCert">
            <ds:KeyInfo>
                <ds:KeyName>NegoInit_AppCert_Key</ds:KeyName>
            </ds:KeyInfo>
        </tp:Certificate>
        <tp:SecurityDetails tp:securityId="NegoInit_MessageSecurity">
            <tp:TrustAnchors>
                <tp:AnchorCertificateRef tp:certId="NegoInit_AppCert"/>
            </tp:TrustAnchors>
        </tp:SecurityDetails>
        <tp:DeliveryChannel tp:channelId="asyncChannelA1" tp:transportId="transportA1"
tp:docExchangeId="docExchangeA1">
            <tp:MessagingCharacteristics tp:syncReplyMode="none" tp:ackRequested="always"
tp:ackSignatureRequested="always" tp:duplicateElimination="always"/>
        </tp:DeliveryChannel>
        <tp:Transport tp:transportId="transportA1">
            <tp:TransportSender>
                <tp:TransportProtocol tp:version="1.1">HTTP</tp:TransportProtocol>
                <tp:AccessAuthentication>basic</tp:AccessAuthentication>
            </tp:TransportSender>
            <tp:TransportReceiver>
                <tp:TransportProtocol tp:version="1.1">HTTP</tp:TransportProtocol>
                <tp:AccessAuthentication>basic</tp:AccessAuthentication>
                <tp:Endpoint tp:uri="https://www.NegoInit.com/servlets/ebxmlhandler/async" tp:type="allPurpose"/>
            </tp:TransportReceiver>
        </tp:Transport>
        <tp:DocExchange tp:docExchangeId="docExchangeA1">
            <tp:ebXMLSenderBinding tp:version="2.0"/>
            <tp:ebXMLReceiverBinding tp:version="2.0"/>
        </tp:DocExchange>
    </tp:PartyInfo>
    <!-- Party info for Negotiation Responder -->
    <tp:PartyInfo tp:partyName="NegotiationResponder" tp:defaultMshChannelId="asyncChannelB1"
tp:defaultMshPackageId="NegoInit_MshSignalPackage">
        <tp:PartyId tp:type="urn:oasis:names:tc:ebxml-cppa:partyid-type:duns">123456789</tp:PartyId>
        <tp:PartyRef xlink:href="http://NegoResp.com/about.html"/>
        <!-- This role is for Negotiation Responder performing the role of Negotiation Responder -->
        <tp:CollaborationRole>
            <tp:ProcessSpecification tp:version="2.0" tp:name="CPPA-Negotiation" xlink:type="simple"
xlink:href="http://www.oasis-open.org/committees/ebxml-cppa-negot/CPA_Negotiation_BPSS.xml"
tp:uuid="bpid:ebXML:CPPA-Negotiation"/>
            <tp:Role tp:name="CPA Negotiation Responder" xlink:type="simple" xlink:href="http://www.oasis-
open.org/committees/ebxml-cppa-negot/CPA_Negotiation_BPSS.xm#CPA Negotiation Responder"/>
            <tp:ServiceBinding>
                <tp:Service>bpid:ebXML:CPPA-Negotiation</tp:Service>
                <!-- This send is for sending the Negotiation Offer Response, this could be accept, pending, response-->
                <tp:CanSend>
                    <tp:ThisPartyActionBinding tp:id="NegoResp_ABID9" tp:action="CPA_Offer_BT_RespBA"
tp:packageId="NegoInit_OfferResponsePackage">
                        <tp:BusinessTransactionCharacteristics tp:isNonRepudiationRequired="false"
tp:isNonRepudiationReceiptRequired="false" tp:isConfidential="none" tp:isAuthenticated="none" tp:isTamperProof="none"
tp:isAuthorizationRequired="false" tp:timeToAcknowledgeReceipt="PT2H" tp:timeToPerform="P1D"/>
                        <tp:ActionContext tp:binaryCollaboration="CPA Negotiation BC"
tp:businessTransactionActivity="CPA Offer BTA" tp:requestOrResponseAction="CPA_Offer_BT_RespBA"/>
                        <tp:ChannelId>asyncChannelB1</tp:ChannelId>
                    </tp:ThisPartyActionBinding>
                    <tp:OtherPartyActionBinding>NegoInit_ABID9</tp:OtherPartyActionBinding>
                </tp:CanSend>
                <!-- This send is for sending the Final Response document in the final BTA -->
                <tp:CanSend>
                    <tp:ThisPartyActionBinding tp:id="NegoResp_FinalResponseB" tp:action="Final_CPA_BT_ReqBA"
tp:packageId="NegoInit_FinalMessage">
                        <tp:BusinessTransactionCharacteristics tp:isNonRepudiationRequired="false"
tp:isNonRepudiationReceiptRequired="false" tp:isConfidential="none" tp:isAuthenticated="none" tp:isTamperProof="none"
tp:isAuthorizationRequired="false" tp:timeToAcknowledgeReceipt="PT2H" tp:timeToPerform="P1D"/>
                        <tp:ActionContext tp:binaryCollaboration="CPA Negotiation BC"
tp:businessTransactionActivity="CPA Final BTA" tp:requestOrResponseAction="Final_CPA_BT_ReqBA"/>
                        <tp:ChannelId>asyncChannelA1</tp:ChannelId>
                    </tp:ThisPartyActionBinding>
                    <tp:OtherPartyActionBinding>NegoInit_FinalResponseA</tp:OtherPartyActionBinding>
                </tp:CanSend>
                <!-- This send is for sending the Receipt Acknowledgment -->
                <tp:CanSend>
                    <tp:ThisPartyActionBinding tp:id="NegoResp_ABID3" tp:action="ReceiptAcknowledgement"
tp:packageId="NegoInit_ReceiptAcknowledgmentPackage">
                        <tp:BusinessTransactionCharacteristics tp:isNonRepudiationRequired="true"
tp:isNonRepudiationReceiptRequired="true" tp:isConfidential="transient" tp:isAuthenticated="persistent"
tp:isTamperProof="persistent" tp:isAuthorizationRequired="true"/>
                        <tp:ChannelId>asyncChannelB1</tp:ChannelId>
                    </tp:ThisPartyActionBinding>
                    <tp:OtherPartyActionBinding>NegoInit_ABID3</tp:OtherPartyActionBinding>
                </tp:CanSend>
                <!-- This receive is for receiving the Final message in the collaboration. This would be the double signed
CPA document or acceptance or reject of the CPA  in the final Response document-->
                <tp:CanReceive>
                    <tp:ThisPartyActionBinding tp:id="NegoResp_FinalResponseMessageB"
tp:action="Final_CPA_BT_RespBA" tp:packageId="NegoInit_FinalMessage">
                        <tp:BusinessTransactionCharacteristics tp:isNonRepudiationRequired="false"
tp:isNonRepudiationReceiptRequired="false" tp:isConfidential="none" tp:isAuthenticated="none" tp:isTamperProof="none"
tp:isAuthorizationRequired="false" tp:timeToAcknowledgeReceipt="PT2H" tp:timeToPerform="P1D"/>
                        <tp:ActionContext tp:binaryCollaboration="CPA Negotiation BC"
tp:businessTransactionActivity="CPA Final BTA" tp:requestOrResponseAction="Final_CPA_BT_RespBA"/>
```

```
                        <tp:ChannelId>asyncChannelA1</tp:ChannelId>
                    </tp:ThisPartyActionBinding>
                    <tp:OtherPartyActionBinding>NegoInit_FinalResponseMessageA</tp:OtherPartyActionBinding>
                </tp:CanReceive>
                <!-- This receive is for receiving theoffer in the first place -->
                <tp:CanReceive>
                    <tp:ThisPartyActionBinding tp:id="NegoResp_ABID1" tp:action="CPA_Offer_BT_ReqBA"
tp:packageId="NegoInit_OfferRequestPackage">
                        <tp:BusinessTransactionCharacteristics tp:isNonRepudiationRequired="false"
tp:isNonRepudiationReceiptRequired="false" tp:isConfidential="none" tp:isAuthenticated="none" tp:isTamperProof="none"
tp:isAuthorizationRequired="false" tp:timeToAcknowledgeReceipt="PT2H" tp:timeToPerform="P1D"/>
                        <tp:ActionContext tp:binaryCollaboration="CPA Negotiation BC"
tp:businessTransactionActivity="CPA Offer BTA" tp:requestOrResponseAction="CPA_Offer_BT_ReqBA"/>
                        <tp:ChannelId>asyncChannelB1</tp:ChannelId>
                    </tp:ThisPartyActionBinding>
                    <tp:OtherPartyActionBinding>NegoInit_ABID1</tp:OtherPartyActionBinding>
                </tp:CanReceive>
                <!-- This Receive is for receiving the Receipt Acknowledgment -->
                <tp:CanReceive>
                    <tp:ThisPartyActionBinding tp:id="NegoResp_ABID2" tp:action="ReceiptAcknowledgment"
tp:packageId="NegoInit_ReceiptAcknowledgmentPackage">
                        <tp:BusinessTransactionCharacteristics tp:isNonRepudiationRequired="true"
tp:isNonRepudiationReceiptRequired="true" tp:isConfidential="transient" tp:isAuthenticated="persistent"
tp:isTamperProof="persistent" tp:isAuthorizationRequired="true"/>
                        <tp:ChannelId>asyncChannelB1</tp:ChannelId>
                    </tp:ThisPartyActionBinding>
                    <tp:OtherPartyActionBinding>NegoInit_ABID2</tp:OtherPartyActionBinding>
                </tp:CanReceive>
                <!-- This Receive is for receiving the Exception -->
                <tp:CanReceive>
                    <tp:ThisPartyActionBinding tp:id="NegoResp_ABID4" tp:action="Exception"
tp:packageId="NegoInit_ExceptionPackage">
                        <tp:BusinessTransactionCharacteristics tp:isNonRepudiationRequired="true"
tp:isNonRepudiationReceiptRequired="true" tp:isConfidential="transient" tp:isAuthenticated="persistent"
tp:isTamperProof="persistent" tp:isAuthorizationRequired="true"/>
                        <tp:ChannelId>asyncChannelB1</tp:ChannelId>
                    </tp:ThisPartyActionBinding>
                    <tp:OtherPartyActionBinding>NegoInit_ABID4</tp:OtherPartyActionBinding>
                </tp:CanReceive>
            </tp:ServiceBinding>
        </tp:CollaborationRole>
        <!-- This role is for Negotiation Responder company performing the role of Negotiation Counter offer initiator -->
        <tp:CollaborationRole>
            <tp:ProcessSpecification tp:version="2.0" tp:name="CPPA-Negotiation" xlink:type="simple"
xlink:href="http://www.oasis-open.org/committees/ebxml-cppa-negot/CPA_Negotiation_BPSS.xml"
tp:uuid="bpid:ebXML:CPPA-Negotiation"/>
            <tp:Role tp:name="CPA Negotiation Counter Offer Initiator" xlink:type="simple" xlink:href="http://www.oasis-
open.org/committees/ebxml-cppa-negot/CPA_Negotiation_BPSS.xm#CPA Negotiation Counter Offer Initiator"/>
            <tp:ServiceBinding>
                <tp:Service>bpid:ebXML:CPPA-Negotiation</tp:Service>
                <!-- This send is for sending the Negotiation Counter Offer.This happens in "CPA Counter Offer 1 BTA" -->
                <tp:CanSend>
                    <tp:ThisPartyActionBinding tp:id="NegoResp_ABID12" tp:action="CPA_Counter_Offer_BT_ReqBA"
tp:packageId="NegoInit_CounterOfferRequestPackage">
                        <tp:BusinessTransactionCharacteristics tp:isNonRepudiationRequired="false"
tp:isNonRepudiationReceiptRequired="false" tp:isConfidential="none" tp:isAuthenticated="none" tp:isTamperProof="none"
tp:isAuthorizationRequired="false" tp:timeToAcknowledgeReceipt="PT2H" tp:timeToPerform="P1D"/>
                        <tp:ActionContext tp:binaryCollaboration="CPA Negotiation BC"
tp:businessTransactionActivity="CPA Counter Offer 1 BTA"
tp:requestOrResponseAction="CPA_Counter_Offer_BT_ReqBA">
                            <tp:CollaborationActivity tp:name="CPA Counter Offer CA"/>
                        </tp:ActionContext>
                        <tp:ChannelId>asyncChannelB1</tp:ChannelId>
                    </tp:ThisPartyActionBinding>
                    <tp:OtherPartyActionBinding>NegoInit_ABID12</tp:OtherPartyActionBinding>
                </tp:CanSend>
                <!-- This send is for sending the Negotiation Counter Offer response.This happens in "CPA Counter Offer 2
BTA" -->
                <tp:CanSend>
                    <tp:ThisPartyActionBinding tp:id="NegoResp_ABID10" tp:action="CPA_Counter_Offer_BT_ReqBA"
tp:packageId="NegoInit_CounterOfferResponsePackage">
                        <tp:BusinessTransactionCharacteristics tp:isNonRepudiationRequired="false"
tp:isNonRepudiationReceiptRequired="false" tp:isConfidential="none" tp:isAuthenticated="none" tp:isTamperProof="none"
tp:isAuthorizationRequired="false" tp:timeToAcknowledgeReceipt="PT2H" tp:timeToPerform="P1D"/>
                        <tp:ActionContext tp:binaryCollaboration="CPA Negotiation BC"
tp:businessTransactionActivity="CPA Counter Offer 2 BTA"
tp:requestOrResponseAction="CPA_Counter_Offer_BT_RespBA">
                            <tp:CollaborationActivity tp:name="CPA Counter Offer CA"/>
                        </tp:ActionContext>
                        <tp:ChannelId>asyncChannelB1</tp:ChannelId>
                    </tp:ThisPartyActionBinding>
                    <tp:OtherPartyActionBinding>NegoInit_ABID10</tp:OtherPartyActionBinding>
                </tp:CanSend>
                <!-- This send is for sending the Receipt Acknowledgment -->
                <tp:CanSend>
                    <tp:ThisPartyActionBinding tp:id="NegoResp_ABID7" tp:action="ReceiptAcknowledgement"
tp:packageId="NegoInit_ReceiptAcknowledgmentPackage">
                        <tp:BusinessTransactionCharacteristics tp:isNonRepudiationRequired="true"
tp:isNonRepudiationReceiptRequired="true" tp:isConfidential="transient" tp:isAuthenticated="persistent"
tp:isTamperProof="persistent" tp:isAuthorizationRequired="true"/>
                        <tp:ChannelId>asyncChannelB1</tp:ChannelId>
                    </tp:ThisPartyActionBinding>
                    <tp:OtherPartyActionBinding>NegoInit_ABID7</tp:OtherPartyActionBinding>
                </tp:CanSend>
                <!-- This send is for sending the Final CPA message in CPA_Final_BTA_init_Initiator"-->
                <tp:CanSend>
                    <tp:ThisPartyActionBinding tp:id="NegoCOResp_FinalMessageB1" tp:action="Final_CPA_BT_ReqBA"
tp:packageId="NegoInit_FinalMessage">
                        <tp:BusinessTransactionCharacteristics tp:isNonRepudiationRequired="false"
tp:isNonRepudiationReceiptRequired="false" tp:isConfidential="none" tp:isAuthenticated="none" tp:isTamperProof="none"
tp:isAuthorizationRequired="false" tp:timeToAcknowledgeReceipt="PT2H" tp:timeToPerform="P1D"/>
                        <tp:ActionContext tp:binaryCollaboration="CPA Negotiation BC"
tp:businessTransactionActivity="CPA_Final_BTA_init_Initiator" tp:requestOrResponseAction="Final_CPA_BT_ReqBA">
                            <tp:CollaborationActivity tp:name="CPA Counter Offer CA"/>
                        </tp:ActionContext>
                        <tp:ChannelId>asyncChannelA1</tp:ChannelId>
                    </tp:ThisPartyActionBinding>
                    <tp:OtherPartyActionBinding>NegoCOR_FinalMessageA1</tp:OtherPartyActionBinding>
                </tp:CanSend>
                <!-- This send is for sending the response to the Final CPA message in CPA_Final_BTA_init_Responder"-->
                <tp:CanSend>
                    <tp:ThisPartyActionBinding tp:id="NegoCOResp_FinalMessageResponseB2"
tp:action="Final_CPA_BT_RespBA" tp:packageId="NegoInit_FinalMessage">
                        <tp:BusinessTransactionCharacteristics tp:isNonRepudiationRequired="false"
tp:isNonRepudiationReceiptRequired="false" tp:isConfidential="none" tp:isAuthenticated="none" tp:isTamperProof="none"
tp:isAuthorizationRequired="false" tp:timeToAcknowledgeReceipt="PT2H" tp:timeToPerform="P1D"/>
                        <tp:ActionContext tp:binaryCollaboration="CPA Negotiation BC"
tp:businessTransactionActivity="CPA_Final_BTA_init_Responder"
tp:requestOrResponseAction="Final_CPA_BT_RespBA">
                            <tp:CollaborationActivity tp:name="CPA Counter Offer CA"/>
                        </tp:ActionContext>
                        <tp:ChannelId>asyncChannelA1</tp:ChannelId>
                    </tp:ThisPartyActionBinding>
                    <tp:OtherPartyActionBinding>NegoCOR_FinalMessageResponseA2</tp:OtherPartyActionBinding>
                </tp:CanSend>
                <!-- This receive is for receiving the response forNegotiation Counter Offer, could be accept, reject or again
send a counter offer This happens in "CPA Counter Offer 1 BTA"-->
                <tp:CanReceive>
                    <tp:ThisPartyActionBinding tp:id="NegoResp_ABID6" tp:action="CPA_Counter_Offer_BT_RespBA"
tp:packageId="NegoInit_CounterOfferResponsePackage">
                        <tp:BusinessTransactionCharacteristics tp:isNonRepudiationRequired="false"
tp:isNonRepudiationReceiptRequired="false" tp:isConfidential="none" tp:isAuthenticated="none" tp:isTamperProof="none"
```

```
tp:isAuthorizationRequired="false" tp:timeToAcknowledgeReceipt="PT2H" tp:timeToPerform="P1D"/>
            <tp:ActionContext tp:binaryCollaboration="CPA Negotiation BC"
tp:businessTransactionActivity="CPA Counter Offer 1 BTA"
tp:requestOrResponseAction="CPA_Counter_Offer_BT_RespBA">
              <tp:CollaborationActivity tp:name="CPA Counter Offer CA"/>
            </tp:ActionContext>
            <tp:ChannelId>asyncChannelB1</tp:ChannelId>
          </tp:ThisPartyActionBinding>
          <tp:OtherPartyActionBinding>NegoInit_ABID6</tp:OtherPartyActionBinding>
        </tp:CanReceive>
        <!-- This receive is for receiving Negotiation Counter Offer.This happens in "CPA Counter Offer 2 BTA"-->
        <tp:CanReceive>
          <tp:ThisPartyActionBinding tp:id="NegoResp_ABID5" tp:action="CPA_Counter_Offer_BT_ReqBA"
tp:packageId="NegoInit_CounterOfferRequestPackage">
            <tp:BusinessTransactionCharacteristics tp:isNonRepudiationRequired="false"
tp:isNonRepudiationReceiptRequired="false" tp:isConfidential="none" tp:isAuthenticated="none" tp:isTamperProof="none"
tp:isAuthorizationRequired="false" tp:timeToAcknowledgeReceipt="PT2H" tp:timeToPerform="P1D"/>
            <tp:ActionContext tp:binaryCollaboration="CPA Negotiation BC"
tp:businessTransactionActivity="CPA Counter Offer 2 BTA"
tp:requestOrResponseAction="CPA_Counter_Offer_BT_ReqBA">
              <tp:CollaborationActivity tp:name="CPA Counter Offer CA"/>
            </tp:ActionContext>
            <tp:ChannelId>asyncChannelB1</tp:ChannelId>
          </tp:ThisPartyActionBinding>
          <tp:OtherPartyActionBinding>NegoInit_ABID5</tp:OtherPartyActionBinding>
        </tp:CanReceive>
        <!-- This receive is for receiving the Final CPA message in CPA_Final_BTA_init_Responder"-->
        <tp:CanReceive>
          <tp:ThisPartyActionBinding tp:id="NegoCOResp_FinalMessageB" tp:action="Final_CPA_BT_ReqBA"
tp:packageId="NegoInit_FinalMessage">
            <tp:BusinessTransactionCharacteristics tp:isNonRepudiationRequired="false"
tp:isNonRepudiationReceiptRequired="false" tp:isConfidential="none" tp:isAuthenticated="none" tp:isTamperProof="none"
tp:isAuthorizationRequired="false" tp:timeToAcknowledgeReceipt="PT2H" tp:timeToPerform="P1D"/>
            <tp:ActionContext tp:binaryCollaboration="CPA Negotiation BC"
tp:businessTransactionActivity="CPA_Final_BTA_init_Responder" tp:requestOrResponseAction="Final_CPA_BT_ReqBA">
              <tp:CollaborationActivity tp:name="CPA Counter Offer CA"/>
            </tp:ActionContext>
            <tp:ChannelId>asyncChannelA1</tp:ChannelId>
          </tp:ThisPartyActionBinding>
          <tp:OtherPartyActionBinding>NegoCOR_FinalMessageA</tp:OtherPartyActionBinding>
        </tp:CanReceive>
        <!-- This receive is for receiving the response to the Final CPA message in CPA_Final_BTA_init_Initiator"-->
        <tp:CanReceive>
          <tp:ThisPartyActionBinding tp:id="NegoCOResp_FinalMessageResponseB"
tp:action="Final_CPA_BT_RespBA" tp:packageId="NegoInit_FinalMessage">
            <tp:BusinessTransactionCharacteristics tp:isNonRepudiationRequired="false"
tp:isNonRepudiationReceiptRequired="false" tp:isConfidential="none" tp:isAuthenticated="none" tp:isTamperProof="none"
tp:isAuthorizationRequired="false" tp:timeToAcknowledgeReceipt="PT2H" tp:timeToPerform="P1D"/>
            <tp:ActionContext tp:binaryCollaboration="CPA Negotiation BC"
tp:businessTransactionActivity="CPA_Final_BTA_init_Initiator" tp:requestOrResponseAction="Final_CPA_BT_RespBA">
              <tp:CollaborationActivity tp:name="CPA Counter Offer CA"/>
            </tp:ActionContext>
            <tp:ChannelId>asyncChannelA1</tp:ChannelId>
          </tp:ThisPartyActionBinding>
          <tp:OtherPartyActionBinding>NegoCOR_FinalMessageResponseA</tp:OtherPartyActionBinding>
        </tp:CanReceive>
        <!-- This Receive is for receiving the Receipt Acknowledgment -->
        <tp:CanReceive>
          <tp:ThisPartyActionBinding tp:id="NegoResp_ABID13" tp:action="ReceiptAcknowledgment"
tp:packageId="NegoInit_ReceiptAcknowledgmentPackage">
            <tp:BusinessTransactionCharacteristics tp:isNonRepudiationRequired="true"
tp:isNonRepudiationReceiptRequired="true" tp:isConfidential="transient" tp:isAuthenticated="persistent"
tp:isTamperProof="persistent" tp:isAuthorizationRequired="true"/>
            <tp:ChannelId>asyncChannelB1</tp:ChannelId>
          </tp:ThisPartyActionBinding>
          <tp:OtherPartyActionBinding>NegoInit_ABID13</tp:OtherPartyActionBinding>
```

```
                <tp:CanReceive>
                <!-- This Receive is for receiving the Exception -->
                <tp:CanReceive>
                    <tp:ThisPartyActionBinding tp:id="NegoResp_ABID8" tp:action="Exception"
tp:packageId="NegoInit_ExceptionPackage">
                        <tp:BusinessTransactionCharacteristics tp:isNonRepudiationRequired="true"
tp:isNonRepudiationReceiptRequired="true" tp:isConfidential="transient" tp:isAuthenticated="persistent"
tp:isTamperProof="persistent" tp:isAuthorizationRequired="true"/>
                        <tp:ChannelId>asyncChannelB1</tp:ChannelId>
                    </tp:ThisPartyActionBinding>
                    <tp:OtherPartyActionBinding>NegoInit_ABID8</tp:OtherPartyActionBinding>
                </tp:CanReceive>
            </tp:ServiceBinding>
        </tp:CollaborationRole>
        <!-- Certificates used by the "Negotiation Initiator" company -->
        <tp:Certificate tp:certId="NegoResp_AppCert">
            <ds:KeyInfo>
                <ds:KeyName>NegoResp_AppCert_Key</ds:KeyName>
            </ds:KeyInfo>
        </tp:Certificate>
        <tp:SecurityDetails tp:securityId="NegoResp_MessageSecurity">
            <tp:TrustAnchors>
                <tp:AnchorCertificateRef tp:certId="NegoResp_AppCert"/>
            </tp:TrustAnchors>
        </tp:SecurityDetails>
        <tp:DeliveryChannel tp:channelId="asyncChannelB1" tp:transportId="transportB1"
tp:docExchangeId="docExchangeB1">
            <tp:MessagingCharacteristics tp:syncReplyMode="none" tp:ackRequested="always"
tp:ackSignatureRequested="always" tp:duplicateElimination="always"/>
        </tp:DeliveryChannel>
        <tp:Transport tp:transportId="transportB1">
            <tp:TransportSender>
                <tp:TransportProtocol tp:version="1.1">HTTP</tp:TransportProtocol>
                <tp:AccessAuthentication>basic</tp:AccessAuthentication>
            </tp:TransportSender>
            <tp:TransportReceiver>
                <tp:TransportProtocol tp:version="1.1">HTTP</tp:TransportProtocol>
                <tp:AccessAuthentication>basic</tp:AccessAuthentication>
                <tp:Endpoint tp:uri="https://www.NegoResp.com/servlets/ebxmlhandler/async" tp:type="allPurpose"/>
            </tp:TransportReceiver>
        </tp:Transport>
        <tp:DocExchange tp:docExchangeId="docExchangeB1">
            <tp:ebXMLSenderBinding tp:version="2.0"/>
            <tp:ebXMLReceiverBinding tp:version="2.0"/>
        </tp:DocExchange>
    </tp:PartyInfo>
    <!-- SimplePart corresponding to the SOAP Envelope -->
    <tp:SimplePart tp:id="NegoInit_MsgHdr" tp:mimetype="text/xml">
        <tp:NamespaceSupported tp:location="http://www.oasis-open.org/committees/ebxml-msg/schema/msg-header-
2_0.xsd" tp:version="2.0">
    http://www.oasis-open.org/committees/ebxml-msg/schema/msg-header-2_0.xsd
  </tp:NamespaceSupported>
    </tp:SimplePart>
    <tp:SimplePart tp:id="NegoResp_MsgHdr" tp:mimetype="text/xml">
        <tp:NamespaceSupported tp:location="http://www.oasis-open.org/committees/ebxml-msg/schema/msg-header-
2_0.xsd" tp:version="2.0">
    http://www.oasis-open.org/committees/ebxml-msg/schema/msg-header-2_0.xsd
  </tp:NamespaceSupported>
    </tp:SimplePart>
    <!-- SimplePart corresponding to a Receipt Acknowledgment business signal -->
    <tp:SimplePart tp:id="NegoInit_ReceiptAcknowledgment" tp:mimetype="application/xml">
        <tp:NamespaceSupported tp:location="http://www.ebxml.org/bpss/ReceiptAcknowledgment.xsd"
tp:version="2.0">http://www.ebxml.org/bpss/ReceiptAcknowledgment.xsd
   </tp:NamespaceSupported>
    </tp:SimplePart>
    <tp:SimplePart tp:id="NegoResp_ReceiptAcknowledgment" tp:mimetype="application/xml">
```

```
      <tp:NamespaceSupported tp:location="http://www.ebxml.org/bpss/ReceiptAcknowledgment.xsd" tp:version="2.0">
   http://www.ebxml.org/bpss/ReceiptAcknowledgment.xsd
  </tp:NamespaceSupported>
   </tp:SimplePart>
   <!-- SimplePart corresponding to an Exception business signal -->
   <tp:SimplePart tp:id="NegoInit_Exception" tp:mimetype="application/xml">
      <tp:NamespaceSupported tp:location="http://www.oasis-open.org/committees/ebxml-msg/schema/msg-header-
2_0.xsd" tp:version="2.0">
   http://www.ebxml.org/bpss/Exception.xsd
  </tp:NamespaceSupported>
   </tp:SimplePart>
   <tp:SimplePart tp:id="NegoResp_Exception" tp:mimetype="application/xml">
      <tp:NamespaceSupported tp:location="http://www.oasis-open.org/committees/ebxml-msg/schema/msg-header-
2_0.xsd" tp:version="2.0">
   http://www.ebxml.org/bpss/Exception.xsd
  </tp:NamespaceSupported>
   </tp:SimplePart>
   <!-- SimplePart corresponding to a negotiation offer request action -->
   <tp:SimplePart tp:id="NegoInit_OfferRequest" tp:mimetype="application/xml">
      <tp:NamespaceSupported tp:location="http://www.ebxml.org/schemas/NegotiationOffer.xsd" tp:version="1.0">
   http://www.ebxml.org/schemas/NegotiationOffer.xsd
  </tp:NamespaceSupported>
   </tp:SimplePart>
   <!-- SimplePart corresponding to a Negotiation offer response action (accept) -->
   <tp:SimplePart tp:id="NegoInit_OfferAccept" tp:mimetype="application/xml">
      <tp:NamespaceSupported tp:location="http://www.ebxml.org/schemas/OfferAccept.xsd" tp:version="1.0">
   http://www.ebxml.org/schemas/OfferAccept.xsd
  </tp:NamespaceSupported>
   </tp:SimplePart>
   <!-- SimplePart corresponding to a Negotiation offer response action (reject) -->
   <tp:SimplePart tp:id="NegoInit_OfferReject" tp:mimetype="application/xml">
      <tp:NamespaceSupported tp:location="http://www.ebxml.org/schemas/OfferReject.xsd" tp:version="1.0">
   http://www.ebxml.org/schemas/OfferReject.xsd
  </tp:NamespaceSupported>
   </tp:SimplePart>
   <!-- SimplePart corresponding to a Negotiation offer response action (counter pending) -->
   <tp:SimplePart tp:id="NegoInit_OfferCounterPending" tp:mimetype="application/xml">
      <tp:NamespaceSupported tp:location="http://www.ebxml.org/schemas/OfferCounterPending.xsd" tp:version="1.0">
   http://www.ebxml.org/schemas/OfferCounterPending.xsd
  </tp:NamespaceSupported>
   </tp:SimplePart>
   <!-- SimplePart corresponding to a Negotiation Counter  offer request action  -->
   <tp:SimplePart tp:id="NegoInit_CounterOfferRequest" tp:mimetype="application/xml">
      <tp:NamespaceSupported tp:location="http://www.ebxml.org/schemas/CounterOffer.xsd" tp:version="1.0">
   http://www.ebxml.org/schemas/CounterOfferRequest.xsd
  </tp:NamespaceSupported>
   </tp:SimplePart>
   <!-- SimplePart corresponding to a Negotiation Final document being sent in the negotiation process  -->
   <tp:SimplePart tp:id="NegoInit_FinalMessage" tp:mimetype="application/xml">
      <tp:NamespaceSupported tp:location="http://www.ebxml.org/schemas/FinalMessage.xsd" tp:version="1.0">
   http://www.ebxml.org/schemas/FinalMessage.xsd
  </tp:NamespaceSupported>
   </tp:SimplePart>
   <!-- SimplePart corresponding to a Negotiation Counter  offer request action  -->
   <tp:SimplePart tp:id="NegoInit_FinalMessageResponse" tp:mimetype="application/xml">
      <tp:NamespaceSupported tp:location="http://www.ebxml.org/schemas/FinalMessageResponse.xsd"
tp:version="1.0">
   http://www.ebxml.org/schemas/FinalMessageResponse.xsd
  </tp:NamespaceSupported>
   </tp:SimplePart>
   <!-- An ebXML message with a SOAP Envelope only -->
   <tp:Packaging tp:id="NegoInit_MshSignalPackage">
      <tp:ProcessingCapabilities tp:parse="true" tp:generate="true"/>
      <tp:CompositeList>
         <tp:Composite tp:id="NegoInit_MshSignal" tp:mimetype="multipart/related" tp:mimeparameters="type=text/xml">
            <tp:Constituent tp:idref="NegoInit_MsgHdr"/>
            </tp:Composite>
        </tp:CompositeList>
    </tp:Packaging>
    <!-- An ebXML message with a SOAP Envelope plus a Offer action payload -->
    <tp:Packaging tp:id="NegoInit_OfferRequestPackage">
        <tp:ProcessingCapabilities tp:parse="true" tp:generate="true"/>
        <tp:CompositeList>
            <tp:Composite tp:id="NegoInit_OfferRequestMsgId" tp:mimetype="multipart/related"
tp:mimeparameters="type=text/xml">
                <tp:Constituent tp:idref="NegoInit_MsgHdr"/>
                <tp:Constituent tp:idref="NegoInit_OfferRequest"/>
            </tp:Composite>
        </tp:CompositeList>
    </tp:Packaging>
    <!-- An ebXML message with a SOAP Envelope plus a offer response action payload -->
    <tp:Packaging tp:id="NegoInit_OfferResponsePackage">
        <tp:ProcessingCapabilities tp:parse="true" tp:generate="true"/>
        <tp:CompositeList>
            <tp:Composite tp:id="NegoInit_OfferResponseAcceptMsgId" tp:mimetype="multipart/related"
tp:mimeparameters="type=text/xml">
                <tp:Constituent tp:idref="NegoInit_MsgHdr"/>
                <tp:Constituent tp:idref="NegoInit_OfferAccept"/>
            </tp:Composite>
        </tp:CompositeList>
        <tp:CompositeList>
            <tp:Composite tp:id="NegoInit_OfferResponseRejectMsgId" tp:mimetype="multipart/related"
tp:mimeparameters="type=text/xml">
                <tp:Constituent tp:idref="NegoInit_MsgHdr"/>
                <tp:Constituent tp:idref="NegoInit_OfferReject"/>
            </tp:Composite>
        </tp:CompositeList>
        <tp:CompositeList>
            <tp:Composite tp:id="NegoInit_OfferResponsePendingMsgId" tp:mimetype="multipart/related"
tp:mimeparameters="type=text/xml">
                <tp:Constituent tp:idref="NegoInit_MsgHdr"/>
                <tp:Constituent tp:idref="NegoInit_OfferCounterPending"/>
            </tp:Composite>
        </tp:CompositeList>
    </tp:Packaging>
    <!-- An ebXML message with a SOAP Envelope plus a counter offer request action payload -->
    <tp:Packaging tp:id="NegoInit_CounterOfferRequestPackage">
        <tp:ProcessingCapabilities tp:parse="true" tp:generate="true"/>
        <tp:CompositeList>
            <tp:Composite tp:id="NegoInit_CounterOfferRequestMsgId" tp:mimetype="multipart/related"
tp:mimeparameters="type=text/xml">
                <tp:Constituent tp:idref="NegoInit_MsgHdr"/>
                <tp:Constituent tp:idref="NegoInit_CounterOfferRequest"/>
            </tp:Composite>
        </tp:CompositeList>
    </tp:Packaging>
    <!-- An ebXML message with a SOAP Envelope plus a counter offer response action payload -->
    <tp:Packaging tp:id="NegoInit_CounterOfferResponsePackage">
        <tp:ProcessingCapabilities tp:parse="true" tp:generate="true"/>
        <tp:CompositeList>
            <tp:Composite tp:id="NegoInit_CounterOfferResponseAcceptMsgId" tp:mimetype="multipart/related"
tp:mimeparameters="type=text/xml">
                <tp:Constituent tp:idref="NegoInit_MsgHdr"/>
                <tp:Constituent tp:idref="NegoInit_OfferAccept"/>
            </tp:Composite>
        </tp:CompositeList>
        <tp:CompositeList>
            <tp:Composite tp:id="NegoInit_CounterOfferResponseRejectMsgId" tp:mimetype="multipart/related"
tp:mimeparameters="type=text/xml">
                <tp:Constituent tp:idref="NegoInit_MsgHdr"/>
                <tp:Constituent tp:idref="NegoInit_OfferReject"/>
            </tp:Composite>
        </tp:CompositeList>
        <tp:CompositeList>
            <tp:Composite tp:id="NegoInit_CounterOfferResponsePendingMsgId" tp:mimetype="multipart/related"
tp:mimeparameters="type=text/xml">
                <tp:Constituent tp:idref="NegoInit_MsgHdr"/>
                <tp:Constituent tp:idref="NegoInit_OfferCounterPending"/>
            </tp:Composite>
        </tp:CompositeList>
    </tp:Packaging>
    <!-- An ebXML message with a SOAP Envelope plus a Receipt Acknowledgment payload -->
    <tp:Packaging tp:id="NegoInit_ReceiptAcknowledgmentPackage">
        <tp:ProcessingCapabilities tp:parse="true" tp:generate="true"/>
        <tp:CompositeList>
            <tp:Composite tp:id="NegoInit_ReceiptAcknowledgmentMsg" tp:mimetype="multipart/related"
tp:mimeparameters="type=text/xml">
                <tp:Constituent tp:idref="NegoInit_MsgHdr"/>
                <tp:Constituent tp:idref="NegoInit_ReceiptAcknowledgment"/>
            </tp:Composite>
        </tp:CompositeList>
    </tp:Packaging>
    <tp:Packaging tp:id="NegoResp_ReceiptAcknowledgmentPackage">
        <tp:ProcessingCapabilities tp:parse="true" tp:generate="true"/>
        <tp:CompositeList>
            <tp:Composite tp:id="NegoResp_ReceiptAcknowledgmentMsg" tp:mimetype="multipart/related"
tp:mimeparameters="type=text/xml">
                <tp:Constituent tp:idref="NegoResp_MsgHdr"/>
                <tp:Constituent tp:idref="NegoResp_ReceiptAcknowledgment"/>
            </tp:Composite>
        </tp:CompositeList>
    </tp:Packaging>
    <!-- An ebXML message with a SOAP Envelope plus an Exception payload -->
    <tp:Packaging tp:id="NegoInit_ExceptionPackage">
        <tp:ProcessingCapabilities tp:parse="true" tp:generate="true"/>
        <tp:CompositeList>
            <tp:Composite tp:id="NegoInit_ExceptionMsg" tp:mimetype="multipart/related"
tp:mimeparameters="type=text/xml">
                <tp:Constituent tp:idref="NegoInit_MsgHdr"/>
                <tp:Constituent tp:idref="NegoInit_Exception"/>
            </tp:Composite>
        </tp:CompositeList>
    </tp:Packaging>
    <tp:Packaging tp:id="NegoResp_ExceptionPackage">
        <tp:ProcessingCapabilities tp:parse="true" tp:generate="true"/>
        <tp:CompositeList>
            <tp:Composite tp:id="NegoResp_ExceptionMsg" tp:mimetype="multipart/related"
tp:mimeparameters="type=text/xml">
                <tp:Constituent tp:idref="NegoResp_MsgHdr"/>
                <tp:Constituent tp:idref="NegoResp_Exception"/>
            </tp:Composite>
        </tp:CompositeList>
    </tp:Packaging>
    <tp:Comment xml:lang="en-US">CPPA negotiation between NegoInit.com and NegoResp.com</tp:Comment>
</tp:CollaborationProtocolAgreement>
```

Appendix D Negotiation BPSS Instance Document (Normative)

The text file for this example of the BPSS instance document for automated negotiation is available at:

THIS APPENDIX HAS BEEN REPLACED IN FULL.

THIS SHOULD BE REPLACED BY A "PRETTY PRINTING" VERSION.

```
<?xml version="1.0" encoding="UTF-8"?>
<ProcessSpecification xmlns="http://www.ebxml.org/BusinessProcess"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.ebxml.org/BusinessProcess ebBPSS1.04.xsd"
name="CPPA-Negotiation" uuid="bpid:ebXML:CPPA-Negotiation" version="R02.00">
      <Documentation>This business process describes CPPA negotiation
process</Documentation>
      <!--CPA Offer Document-->
      <BusinessDocument name="CPA Offer Doc" nameID="CPA_Offer_Doc"/>
      <!--CPA Acceptt Offer Document-->
      <BusinessDocument name="CPA Accept Offer Doc"
nameID="CPA_Accept_Offer_Doc"/>
      <!--CPA Counter Pending Offer Document-->
      <BusinessDocument name="CPA Counter Pending Offer Doc"
nameID="CPA_Counter_Pending_Offer_Doc"/>
      <!--CPA Counter Offer Document-->
      <BusinessDocument name="CPA Counter Offer Doc"
nameID="CPA_Counter_Offer_Doc"/>
      <!--CPA Reject Offer Document-->
      <BusinessDocument name="CPA Reject Offer Doc"
nameID="CPA_Reject_Offer_Doc"/>
      <!--Changed 09/16 CPA Document. This will probably come from the CPA
specification-->
      <BusinessDocument name="CPA Final Doc" nameID="CPA_Final_Doc"/>
      <!--Changed 09/16 . Response to final CPA Document. This will probably
come from the CPA specification
            This is used when the CPA is not signed just to show acceptance
or denial of final CPA-->
      <BusinessDocument name="CPA Final Response Doc"
nameID="CPA_Final_Response_Doc"/>
      <!--Changed 09/16 . Response to final CPA Document which is signed and
agreed to create a double singed CPA. Receiving party will create a Signature
over the signed CPA and send that. This will probably come from the CPA
specification-->
      <BusinessDocument name="CPA Final Response Doc Signed"
nameID="CPA_Final_Response_Doc_Signed"/>
      <!--Changed 02/28. Response to final CPA Document Could be rejecting
the final CPA cause it's different
from the agreed upon CPA or signature does not verify-->
      <BusinessDocument name="CPA Final Response Reject Doc"
nameID="CPA_Final_Response_Reject_Doc"/>
```

```
      <!-- Changed 09/16. Business Transaction for sending the CPA. This CPA
is sent by the party finally accepting the offer-->
      <BusinessTransaction name="CPA Final BT" nameID="CPA_Final_BT">
            <RequestingBusinessActivity name="Final_CPA_BT_ReqBA"
nameID="Final_CPA_BT_ReqBA" isAuthorizationRequired="false"
isIntelligibleCheckRequired="false" isNonRepudiationReceiptRequired="false"
isNonRepudiationRequired="false">
                  <DocumentEnvelope businessDocument="CPA Final Doc"
businessDocumentIDRef="CPA_Final_Doc" isAuthenticated="none"
isConfidential="none" isTamperProof="none"/>
            </RequestingBusinessActivity>
            <RespondingBusinessActivity name="Final_CPA_BT_RespBA"
nameID="Final_CPA_BT_RespBA" isAuthorizationRequired="false"
isIntelligibleCheckRequired="false" isNonRepudiationRequired="false">
                  <DocumentEnvelope businessDocument="CPA Final Response Doc"
businessDocumentIDRef="CPA_Final_Response_Doc" isAuthenticated="none"
isConfidential="none" isPositiveResponse="true" isTamperProof="none"/>
                  <DocumentEnvelope businessDocument="CPA Final Response Doc
Signed" businessDocumentIDRef="CPA_Final_Response_Doc_Signed"
isAuthenticated="none" isConfidential="none" isPositiveResponse="true"
isTamperProof="none"/>
                  <DocumentEnvelope businessDocument="CPA Final Response
Reject Doc" businessDocumentIDRef="CPA_Final_Response_Reject_Doc"
isAuthenticated="none" isConfidential="none" isPositiveResponse="false"
isTamperProof="none"/>

            </RespondingBusinessActivity>
      </BusinessTransaction>
      <!-- Business Transaction for the original negotiation cppa -->
      <BusinessTransaction name="CPA Offer BT" nameID="CPA_Offer_BT">
            <RequestingBusinessActivity name="CPA_Offer_BT_ReqBA"
nameID="CPA_Offer_BT_ReqBA" isAuthorizationRequired="false"
isIntelligibleCheckRequired="false" isNonRepudiationReceiptRequired="false"
isNonRepudiationRequired="false">
                  <DocumentEnvelope businessDocument="CPA Offer Doc"
businessDocumentIDRef="CPA_Offer_Doc" isAuthenticated="none"
isConfidential="none" isTamperProof="none"/>
            </RequestingBusinessActivity>
            <RespondingBusinessActivity name="CPA_Offer_BT_RespBA"
nameID="CPA_Offer_BT_RespBA" isAuthorizationRequired="false"
isIntelligibleCheckRequired="false" isNonRepudiationRequired="false">
                  <DocumentEnvelope businessDocument="CPA Accept Offer Doc"
businessDocumentIDRef="CPA_Accept_Offer_Doc" isAuthenticated="none"
isConfidential="none" isPositiveResponse="true" isTamperProof="none"/>
                  <DocumentEnvelope businessDocument="CPA Reject Offer Doc"
businessDocumentIDRef="CPA_Reject_Offer_Doc" isAuthenticated="none"
isConfidential="none" isPositiveResponse="false" isTamperProof="none"/>
                  <DocumentEnvelope businessDocument="CPA Counter Pending
Offer Doc" businessDocumentIDRef="CPA_Counter_Pending_Offer_Doc"
isAuthenticated="none" isConfidential="none" isPositiveResponse="true"
isTamperProof="none"/>
            </RespondingBusinessActivity>
      </BusinessTransaction>
      <!-- Business Transaction for sending the counter offer -->
      <BusinessTransaction name="CPA Counter Offer BT"
nameID="CPA_Counter_Offer_BT">
```

```
            <RequestingBusinessActivity name="CPA_Counter_Offer_BT_ReqBA"
nameID="CPA_Counter_Offer_BT_ReqBA" isAuthorizationRequired="false"
isIntelligibleCheckRequired="false" isNonRepudiationReceiptRequired="false"
isNonRepudiationRequired="false">
                  <DocumentEnvelope businessDocument="CPA Counter Offer Doc"
businessDocumentIDRef="CPA_Counter_Offer_Doc" isAuthenticated="none"
isConfidential="none" isTamperProof="none"/>
            </RequestingBusinessActivity>
            <RespondingBusinessActivity name="CPA_Counter_Offer_BT_RespBA"
nameID="CPA_Counter_Offer_BT_RespBA" isAuthorizationRequired="false"
isIntelligibleCheckRequired="false" isNonRepudiationRequired="false">
                  <DocumentEnvelope businessDocument="CPA Accept Offer Doc"
businessDocumentIDRef="CPA_Accept_Offer_Doc" isAuthenticated="none"
isConfidential="none" isPositiveResponse="true" isTamperProof="none"/>
                  <DocumentEnvelope businessDocument="CPA Reject Offer Doc"
businessDocumentIDRef="CPA_Reject_Offer_Doc" isAuthenticated="none"
isConfidential="none" isPositiveResponse="false" isTamperProof="none"/>
                  <DocumentEnvelope businessDocument="CPA Counter Pending
Offer Doc" businessDocumentIDRef="CPA_Counter_Pending_Offer_Doc"
isAuthenticated="none" isConfidential="none" isPositiveResponse="true"
isTamperProof="none"/>
            </RespondingBusinessActivity>
      </BusinessTransaction>
      <!-- Main collaboration for negotiation business process -->
      <BinaryCollaboration name="CPA Negotiation BC"
nameID="CPA_Negotiation_BC" initiatingRole="CPA_Negotiation_Initiator_Role">
            <!-- Role for initiator for negotiation process -->
            <Role name="CPA Negotiation Initiator"
nameID="CPA_Negotiation_Initiator_Role"/>
            <!-- Role for initial responder of business collaboration -->
            <Role name="CPA Negotiation Responder"
nameID="CPA_Negotiation_Responder_Role"/>
            <Start toBusinessState="CPA_Offer_BTA"/>
            <BusinessTransactionActivity name="CPA Offer BTA"
nameID="CPA_Offer_BTA" businessTransaction="CPA Offer BT"
businessTransactionIDRef="CPA_Offer_BT" fromRole="CPA Negotiation Initiator"
fromRoleIDRef="CPA_Negotiation_Initiator_Role" toRole="CPA Negotiation
Responder" toRoleIDRef="CPA_Negotiation_Responder_Role"
isLegallyBinding="false" isConcurrent="false"/>
            <CollaborationActivity name="CPA Counter Offer CA"
binaryCollaboration="CPA Negotiation Counter Offer BC"
binaryCollaborationIDRef="CPA_Negotiation_CounterOfferBC" fromRole="CPA
Negotiation Counter Offer Initiator"
fromRoleIDRef="CPA_Negotiation_CounterOfferInitiator_Role" toRole="CPA
Negotiation Counter Offer Responder"
toRoleIDRef="CPA_Negotiation_CounterOfferResponder_Role"
preCondition="Initiating Role for this activity corresponds to Responding
Role in CPA Offer BTA"/>
            <BusinessTransactionActivity name="CPA Final BTA"
nameID="CPA_Final_BTA" businessTransaction="CPA Final BT"
businessTransactionIDRef="CPA_Final_BT" fromRole="CPA Negotiation Responder"
fromRoleIDRef="CPA_Negotiation_Responder_Role" toRole="CPA Negotiation
Initiator" toRoleIDRef="CPA_Negotiation_Initiator_Role"
isLegallyBinding="false" isConcurrent="false"/>
            <!-- If final CPA BTA goes through fine, then overall
collaboration is marked success -->
```

```
            <Success fromBusinessState="CPA Final BTA"
conditionGuard="Success"/>
            <!-- If inner collaboration goes through fine, then overall
collaboration is marked success. Inner collaboration
                  Would have gone through the transaction that ends up with
either the final CPA (Signed if needed) -->
            <Success fromBusinessState="CPA Counter Offer CA"
conditionGuard="Success"/>
            <!-- If Reject offer document is sent for offer bta collaboration
is marked as failure-->
            <Failure fromBusinessState="CPA Offer BTA"
conditionGuard="BusinessFailure">
                  <ConditionExpression
expressionLanguage="DocumentEnvelopeLanguage" expression="CPA Reject Offer
Doc"/>
            </Failure>
            <!-- Changed 02/28. If final CPA is rejected then its a failure
too. -->
            <Failure fromBusinessState="CPA Final BTA"
conditionGuard="Failure">
                  <ConditionExpression
expressionLanguage="DocumentEnvelopeLanguage" expression="CPA Final Response
Reject Doc"/>
            </Failure>
            <!-- If Final CPA BTA fails for some reason, then collaboration
is marked as failure -->
            <Failure fromBusinessState="CPA Final BTA"
conditionGuard="Failure"/>
            <Failure fromBusinessState="CPA Counter Offer CA"
conditionGuard="Failure"/>
            <!-- Transition to Final CPA offer  binary Transaction if the
responder for main transaction accepts the initial offer -->
            <Transition fromBusinessState="CPA Offer BTA"
toBusinessState="CPA Final BTA">
                  <ConditionExpression
expressionLanguage="DocumentEnvelopeLanguage" expression="CPA Accept Offer
Doc"/>
            </Transition>
            <!-- Transition to counter offer  binary collaboration if the
responder for main transaction returns a counter offer pending message -->
            <Transition fromBusinessState="CPA Offer BTA"
toBusinessState="CPA Counter Offer CA">
                  <ConditionExpression
expressionLanguage="DocumentEnvelopeLanguage" expression="CPA Counter Pending
Offer Doc"/>
            </Transition>
      </BinaryCollaboration>
      <BinaryCollaboration name="CPA Negotiation Counter Offer BC"
nameID="CPA_Negotiation_CounterOfferBC"
initiatingRole="CPA_Negotiation_CounterOfferInitiator_Role">
            <Role name="CPA Negotiation Counter Offer Initiator"
nameID="CPA_Negotiation_CounterOfferInitiator_Role"/>
            <Role name="CPA Negotiation Counter Offer Responder"
nameID="CPA_Negotiation_CounterOfferResponder_Role"/>
            <!-- This collaboration starts with the negotiation process
responder sending the counter offer -->
            <Start toBusinessState="CPA Counter Offer 1 BTA"/>
            <!-- This transaction activity is for negotiation process
responder sending the counter offer -->
            <BusinessTransactionActivity name="CPA Counter Offer 1 BTA"
nameID="CPA_Counter_Offer_1_BTA" businessTransaction="CPA Counter Offer BT"
businessTransactionIDRef="CPA_Counter_Offer_BT" fromRole="CPA Negotiation
Counter Offer Initiator"
fromRoleIDRef="CPA_Negotiation_CounterOfferInitiator_Role" toRole="CPA
Negotiation Counter Offer Non Initiator"
toRoleIDRef="CPA_Negotiation_CounterOfferResponder_Role"
isLegallyBinding="false" isConcurrent="false" postCondition="Parties reverse
roles they play"/>
            <!-- This transaction acitvity is for negotiation process
initiator sending the counter offer -->
            <BusinessTransactionActivity name="CPA Counter Offer 2 BTA"
nameID="CPA_Counter_Offer_2_BTA" businessTransaction="CPA Counter Offer BT"
businessTransactionIDRef="CPA_Counter_Offer_BT" fromRole="CPA Negotiation
Counter Offer Responder"
fromRoleIDRef="CPA_Negotiation_CounterOfferResponder_Role" toRole="CPA
Negotiation Counter Offer Initiator"
toRoleIDRef="CPA_Negotiation_CounterOfferInitiator_Role"
isLegallyBinding="false" isConcurrent="false" postCondition="Parties reverse
roles they play"/>
            <BusinessTransactionActivity name="CPA Final BTA Init Initiator"
nameID="CPA_Final_BTA_init_Initiator" businessTransaction="CPA Final BT"
businessTransactionIDRef="CPA_Final_BT" fromRole="CPA Negotiation Counter
Offer Initiator" fromRoleIDRef="CPA_Negotiation_CounterOfferInitiator_Role"
toRole="CPA Negotiation Counter Offer Responder"
toRoleIDRef="CPA_Negotiation_CounterOfferResponder_Role"
isLegallyBinding="false" isConcurrent="false"/>
            <BusinessTransactionActivity name="CPA Final BTA Init Responder"
nameID="CPA_Final_BTA_init_Responder" businessTransaction="CPA Final BT"
businessTransactionIDRef="CPA_Final_BT" fromRole="CPA Negotiation Counter
Offer Responder" fromRoleIDRef="CPA_Negotiation_CounterOfferResponder_Role"
toRole="CPA Negotiation Counter Offer Initiator"
toRoleIDRef="CPA_Negotiation_CounterOfferInitiator_Role"
isLegallyBinding="false" isConcurrent="false"/>
            <!-- Inner collaboration succeeds if the final BTA which involves
sending final CPA succeeds -->
            <Success fromBusinessState="CPA Final BTA Init Initiator"
conditionGuard="Success"/>
            <!-- Inner collaboration succeeds if the final BTA which involves
sending final CPA succeeds. This is
                  the same as above but the difference is this initiated by a
different party -->
            <Success fromBusinessState="CPA Final BTA Init Responder"
conditionGuard="Success"/>
            <!-- Changed 02/28. If final CPA is rejected then its a failure
too. -->
            <Failure fromBusinessState="CPA Final BTA Init Initiator"
conditionGuard="BusinessFailure">
                  <ConditionExpression
expressionLanguage="DocumentEnvelopeLanguage" expression="CPA Final Response
Reject Doc"/>
            </Failure>
            <Failure fromBusinessState="CPA Final BTA Init Responder"
conditionGuard="BusinessFailure">
```

```
                  <ConditionExpression
expressionLanguage="DocumentEnvelopeLanguage" expression="CPA Final Response
Reject Doc"/>
            </Failure>
            <!-- Inner collaboration fails if the final BTA which involves
sending final CPA fails -->
            <Failure fromBusinessState="CPA Final BTA Init Initiator"
conditionGuard="Failure"/>
            <!-- Inner collaboration fails if the final BTA which involves
sending final CPA fails. This is
                  the same as above but the difference is this initiated by a
different party -->
            <Failure fromBusinessState="CPA Final BTA Init Responder"
conditionGuard="Failure"/>
            <Failure fromBusinessState="CPA Counter Offer 1 BTA"
conditionGuard="BusinessFailure">
                  <ConditionExpression
expressionLanguage="DocumentEnvelopeLanguage" expression="CPA Reject Offer
Doc"/>
            </Failure>
            <Failure fromBusinessState="CPA Counter Offer 2 BTA"
conditionGuard="BusinessFailure">
                  <ConditionExpression
expressionLanguage="DocumentEnvelopeLanguage" expression="CPA Reject Offer
Doc"/>
            </Failure>
            <!-- If the negotiation process responder (initiator in this
innercollaboration) sends  an acceptace offer, negotiation process responder
sends the final CPA -->
            <Transition fromBusinessState="CPA Counter Offer 2 BTA"
toBusinessState="CPA Final BTA Init Initiator">
                  <ConditionExpression
expressionLanguage="DocumentEnvelopeLanguage" expression="CPA Accept Offer
Doc"/>
            </Transition>
            <!-- If the negotiation process initiator (responder in this
inner collaboration) sends an acceptace offer, negotiation process initiator
sends the final CPA -->
            <Transition fromBusinessState="CPA Counter Offer 1 BTA"
toBusinessState="CPA Final BTA Init Responder">
                  <ConditionExpression
expressionLanguage="DocumentEnvelopeLanguage" expression="CPA Accept Offer
Doc"/>
            </Transition>
            <!-- If the negotiation process responder sends counter offer and
negotiation process initiator sends a counter offer, negotiation process
initiator sends the counter offer next time -->
            <Transition fromBusinessState="CPA Counter Offer 1 BTA"
toBusinessState="CPA Counter Offer 2 BTA">
                  <ConditionExpression
expressionLanguage="DocumentEnvelopeLanguage" expression="CPA Counter Pending
Offer Doc"/>
            </Transition>
            <!-- If the negotiation process initiator sends a counter offer
and negotiation process responds sends a counter offer, negotiation process
responder sends the counter offer next time, hence the transition back to
original BTA-->
```

```
            <Transition fromBusinessState="CPA Counter Offer 2 BTA" 
toBusinessState="CPA Counter Offer 1 BTA">
                  <ConditionExpression 
expressionLanguage="DocumentEnvelopeLanguage" expression="CPA Counter Pending 
Offer Doc"/>
            </Transition>
      </BinaryCollaboration>
</ProcessSpecification>
```

Appendix E Instance Documents for Business Signals

The XML Schemas of the business signals are defined in [ebBPSS].

Acceptance Acknowledgment

The instance document for the AcceptanceAcknowledgment business signal is available as a text file at:

Exception

The instance document for the Exception business signal is available as a text file at:

Appendix F Example of NDD Instance Document (Non-Normative)

The text file for this example of an *NDD* instance document for automated negotiation is available at:

THIS APPENDIX HAS BEEN COMPLETELY REPLACED.

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- edited with XML Spy v4.4 U (http://www.xmlspy.com) by neelakantan kartha (Sterling Commerce) -->
<!-- edited  by neelakantan kartha (Sterling Commerce) -->
<NegotiationDescriptor xmlns="http://www.oasis-open.org/committees/ebxml-cppa/schema/cpp-cpa-negot-2_0.xsd"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://www.oasis-
open.org/committees/ebxml-cppa/schema/cpp-cpa-2_0.xsd
                    NDD11.xsd" xmlns:xsd="http://www.w3.org/2001/XMLSchema" documentLocation="C:\Documents and
Settings\nkartha\My Documents\ebxml\negotiation\cpa-example-2_0a.xml">
    <!--The value of cpaid might be negotiable, since a party might require that the cpaid conform to a particular format.
However, automatic negotiation on the values is difficult. For instance, how does a party convey to the other party the kinds
of cpaids that it deems o.k? Without this information being conveyed somehow, it wil be difficult to come to agreement
automatically. Fortunately, this issue pertains to the negotiation algorithm and not the NDD. For version 1, we can stipulate
that the cpaid must be URI. The mustBeFilledIn attribute signals to the other party that this value must be filled in. This
attribute, which is optional, is often useful in situations (maybe other than cpaid), where the dominant partner fills out most of
the CPA and the only thing that is required of the other partner is to fill in some information. -->
    <NegotiableInformationItem xpath="/CollaborationProtocolAgreement/@cpaid">
        <Value mustBeFilledIn="Yes"/>
    </NegotiableInformationItem>
    <!-- Versions might be negotiable, since one party might have a product that conforms to an earlier version of the spec.
By an OrderedValue, it is implied that there is a preference to the earliervalues or latervalues, as given b the attribute
preferenceOrder-->
    <NegotiableInformationItem xpath="/CollaborationProtocolAgreement/@version">
        <OrderedValue preference="EarlierPreferred">
            <Value> 1.0 </Value>
            <Value> 2_0.a</Value>
        </OrderedValue>
    </NegotiableInformationItem>
    <!--The value can be one of the following three: proposed, agreed and signed. However, I do not think that this attribute
is negotiated-rather one party might set the value of this attribute to agreed, once it is satisfied that a satisfactory CPA has
been reached. Again, how and when this attrribute is set seems part of the negotiation algorithm.
Also, the possible values of this attribute are part of the CPPA schema-hence there is no need to repeat them here -->
    <NegotiableInformationItem xpath="/CollaborationProtocolAgreement/Status/@value">
        <Value/>
    </NegotiableInformationItem>
    <!--Here, I am taking the (simplistic) assumption that each party specfies the earliest time for starting and the latest time
for ending the Start element (that species the  Starting Date and Time for the CPA). No preference function is given. See the
next entry  for an example of how one would encode a piecewise linear preference function.-->
    <NegotiableInformationItem xpath="/CollaborationProtocolAgreement/Start">
        <ValueWithPreferenceMeasure>
            <EndPoints>
                <EarliestStart>1998-01-31T14:20:00.011Z</EarliestStart>
                <LatestEnd>2002-11-21T14:20:00.000-05:00</LatestEnd>
            </EndPoints>
        </ValueWithPreferenceMeasure>
    </NegotiableInformationItem>
    <!--Nothing new here, when compared to the previous NegotiableInformationItem except that this gives an example of a
piecewise linear preference function. I suspect that this might be over engineering at this poin  1998-04-07T18:39:09Z t -->
    <NegotiableInformationItem xpath="/CollaborationProtocolAgreement/End">
        <ValueWithPreferenceMeasure>
            <EndPoints>
                <EarliestStart>1998-04-07T00:18:39.090Z</EarliestStart>
                <LatestEnd>2002-11-31T13:20:00.000-05:00</LatestEnd>
            </EndPoints>
            <PreferenceFunction>
                <PiecewiseLinearPiece>
                    <x-coordinate1>1998-04-07T18:39:09Z </x-coordinate1>
                    <y-coordinate1>15</y-coordinate1>
                    <x-coordinate2>2000-11-31T13:20:00.000-05:00 </x-coordinate2>
                    <y-coordinate2>30</y-coordinate2>
                </PiecewiseLinearPiece>
                <PiecewiseLinearPiece>
                    <x-coordinate1>2000-11-31T13:20:00.000-05:00 </x-coordinate1>
                    <y-coordinate1>30</y-coordinate1>
                    <x-coordinate2>2002-11-31T13:20:00.000-05:00 </x-coordinate2>
                    <y-coordinate2>60</y-coordinate2>
                </PiecewiseLinearPiece>
            </PreferenceFunction>
        </ValueWithPreferenceMeasure>
    </NegotiableInformationItem>
    <!--Note the ConversationConstraints is an element that might be present or absent, and hence a party might negotiate
the presence or absence of this element. There are four cases to consider A party (a) insists that an element must be
present (b) insists that an element is absent (c) is ok with the element being present or absent, but has a preference for one
or the other (d) is o.k with the element being present or absent, and is completely agnostic.-->
    <NegotiableInformationItem xpath="/CollaborationProtocolAgreement/ConversationConstraints">
        <PresentOrNot value="MustBePresent"/>
    </NegotiableInformationItem>
    <!-- Note that invocationLimit is an attribute of ConversationConstraints that may or may not be present. So first of all,
the presence or absence of this attribute may be negotiable. Then, the value of this attribute may also be negotiable -->
    <NegotiableInformationItem xpath="/CollaborationProtocolAgreement/ConversationConstraints/@invocationLimit">
        <IntegerValues>
            <RangeInfo preferenceOrder="SmallerPreferred">
                <EndPoints>
                    <SmallestValue>1</SmallestValue>
                    <LatestValue>5</LatestValue>
                </EndPoints>
            </RangeInfo>
        </IntegerValues>
    </NegotiableInformationItem>
    <!--concurrentConversations is similar to invocationLimit. I am including this just for the sake of illustrating another use of
the schema -->
    <NegotiableInformationItem
xpath="/CollaborationProtocolAgreement/ConversationConstraints/@concurrentConversations">
        <IntegerValues>
            <PresentOrNot value="MustBePresent"/>
            <RangeInfo>
                <EndPoints>
                    <SmallestValue>2</SmallestValue>
                    <LatestValue>8</LatestValue>
                </EndPoints>
                <PreferenceFunction>
                    <FunctionDefinedByEquation> x**2-2*x+3</FunctionDefinedByEquation>
                </PreferenceFunction>
            </RangeInfo>
        </IntegerValues>
    </NegotiableInformationItem>
    <!--The partyInfo element raises a number of interesting issues. In the CPA, there can be exactly two partyInfo
elements, hence there is no negotiation on these once a CPA has been formed. Since this NDD refers to
a CPA, there can be no negotiation on this element. We will forget this for the time being and see what would be the case if
the document referred to by this NDD were a CPP. A CPP can have multiple PartyInfo elements and one among these must
be chosen to form the CPA. Thus the issue here is to associate a preference order between several elements at the same
level. A simple way of doing this in a CPP is as follows: -->
    <NegotiableInformationItem xpath="/CollaborationProtocolProfile/PartyInfo[2]">
        <Preference value="1"/>
    </NegotiableInformationItem>
    <NegotiableInformationItem xpath="/CollaborationProtocolProfile/PartyInfo[1]">
        <Preference value="3"/>
    </NegotiableInformationItem>
    <NegotiableInformationItem xpath="/CollaborationProtocolProfile/PartyInfo[3]">
```

```
            <Preference value="2"/>
        </NegotiableInformationItem>
        <!--This is included to provide an example where the cardinality of an element may be negotiable. It might be the case
that the number of PartyId elements within a partyInfo element is negotiable (because, say of limitations the underlying
system has of handling a large number of partyIds) -->
        <NegotiableInformationItem xpath="/CollaborationProtocolAgreement/PartyInfo/PartyId">
            <Cardinality>
                <RangeInfo>
                    <EndPoints>
                        <SmallestValue>1</SmallestValue>
                        <LatestValue>5</LatestValue>
                    </EndPoints>
                </RangeInfo>
            </Cardinality>
        </NegotiableInformationItem>
        <!--Example of a boolean value-->
        <NegotiableInformationItem
xpath="/CollaborationProtocolAgreement/PartyInfo/CollaborationRole/ServiceBinding/Service/CanSend/ThisPartyActionBind
ing/BusinessTransactionCharacteristics/@isNonRepudiationRequired">
            <BooleanValue preference="TruePreferred">
                <PresentOrNot value="MustBePresent"/>
            </BooleanValue>
        </NegotiableInformationItem>
        <!--Example of negotiating a duration-->
        <NegotiableInformationItem
xpath="/CollaborationProtocolAgreement/PartyInfo/CollaborationRole/ServiceBinding/Service/CanSend/ThisPartyActionBind
ing/BusinessTransactionCharacteristics/@timeToAcknowledgeReceipt">
            <DurationWithPreference preferenceOrder="SmallerPreferred">
                <MinimumDuration>PT5M</MinimumDuration>
                <MaximumDuration>PT6M</MaximumDuration>
            </DurationWithPreference>
        </NegotiableInformationItem>
        <!--This is how would express that one element of an enumeration must be present. Note that the possible values of the
enumeration is defined in the cpp-cpa schema and need not be repeated here-->
        <NegotiableInformationItem
xpath="/CollaborationProtocolAgreement/PartyInfo/DeliveryChannel/MessagingCharactersitcs/@syncReplyMode">
            <OrderedValue>
                <PresentOrNot value="MustBePresent"/>
                <Value> signalsOnly</Value>
            </OrderedValue>
        </NegotiableInformationItem>
</NegotiationDescriptor>
<!--Notes
0. The top element of an NDD document is named NegotiationDescriptor.  The NegotiationDescriptor element contains
   NegotiationInformationItem elements for each item that is negotiable. (
1. The documentLocation attribute of NegotiationDescriptor element is a uri that  points to the document for which this >
   NDD document pertains to. In particular, the xpath attribute of a NegotiablInformationItem element is an xpath of
   this document. The documentLocation attribute is a required attribute.




Non-Negotiable elements and Attributes
================================
1.  CollaborationProtocolAgreement
2.  CollaborationProtocolAgreement/@schemaLocation
3.  CollaborationProtocolAgreement/Status
4.  CollaborationProtocolAgreement/PartyInfo/@partyName (Since this is set by each party, it is difficult to see how this
would be negotiable. If it is, it would be similar to /CollaborationProtocolAgreement/@cpaid)
5.  CollaborationProtocolAgreement/PartyInfo/@defaultMshChannelId and
CollaborationProtocolAgreement/PartyInfo/@defaultMsPackageId  (Again, if these are negotiable, it would be as a result of
the negotiation algorithm recognizing that the default values are not reasonable. Again, only the value can be negotiated, as
in /CollaborationProtocolAgreement/@cpaid)
6.  CollaborationProtocolAgreement/PartyInfo/PartyRef/@xlink:type(always  simple)
7. CollaborationProtocolAgreement/PartyInfo/CollaborationRole/ProcessSpecification
8. CollaborationProtocolAgreement/PartyInfo/CollaborationRole/ProcessSpecification@name
```

```
9. CollaborationProtocolAgreement/PartyInfo/CollaborationRole/ProcessSpecification@xlink:type (always simple)
10.
CollaborationProtocolAgreement/PartyInfo/CollaborationRole/ProcessSpecification/ds:Reference/ds:Trasforms/ds:Transform
/@ds:Algorithm (fixed by the spec)

Elements and attributes similar to others in the sample
=============================================
0. CollaborationProtocolAgreement/PartyInfo/PartyId/@type (similar to /CollaborationProtocolAgreement/@version", with an
enumeration that enumerates the range of understood naming systems)
1.   CollaborationProtocolAgreement/PartyInfo/PartyRef (similar to  CollaborationProtocolAgreement/PartyInfo/PartyId)
2. CollaborationProtocolAgreement/PartyInfo/PartyRef/@xlink:href (similar to /CollaborationProtocolAgreement/@cpaid)
3. CollaborationProtocolAgreement/PartyInfo/PartyRef/@type (similar to CollaborationProtocolAgreement/@version)
4. CollaborationProtocolAgreement/PartyInfo/PartyRef/@schemaLocation
(similar to /CollaborationProtocolAgreement/@cpaid)

5.  CollaborationProtocolAgreement/PartyInfo/CollaborationRole (similar to negotiating the cardinality of
/CollaborationProtocolAgreement/PartyInfo/PartyId)

6.  CollaborationProtocolAgreement/PartyInfo/CollaborationRole/ProcessSpecification@version (similar to
/CollaborationProtocolAgreement/@version)

7.  CollaborationProtocolAgreement/PartyInfo/CollaborationRole/ProcessSpecification@xlink:href and uuid
((similar to /CollaborationProtocolAgreement/@cpaid)


 !-->
```

Appendix G Examples of Negotiation-Message Instance Documents (Non-Normative)

Example of Offer Message Instance Document

THIS APPENDIX HAS BEEN REPLACED IN FULL.

Negotiation Message Offer

```
<?xml version="1.0" encoding="UTF-8"?>
<tp:NegotiationMessage xmlns:tp="http://www.oasis-open.org/committees/ebxml-cppa/schema/cpa-negot-1_0.xsd"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:cppa="http://www.oasis-open.org/committees/ebxml-
cppa/schema/cpp-cpa-2_0.xsd" xsi:schemaLocation="http://www.oasis-open.org/committees/ebxml-cppa/schema/cpa-
negot-1_0.xsd negotiationMsg.xsd" businessMsgId="busMsg001" binding="false" negotiationDialogId="negotDialog001"
offerId="offer001" messageType="Offer">
    <tp:NCPA uri="http://www.companya.com/ncpa/myncpa.xml"/>
    <tp:CPATemplateId id="uri:companyA-and-companyB-CPA1" cppa:version="1.0"/>
    <cppa:SecurityDetails cppa:securityId="ID">
        <cppa:SecurityPolicy/>
    </cppa:SecurityDetails>
    <tp:InitiatingParty>
        <cppa:PartyId cppa:type="urn:oasis:names:tc:ebxml-cppa:partyid-type:duns">123456789</cppa:PartyId>
        <CPPId id="companya-cpp123456789" cppa:version="1.0"/>
    </tp:InitiatingParty>
    <tp:RespondingParty>
        <cppa:PartyId cppa:type="urn:oasis:names:tc:ebxml-cppa:partyid-type:duns">987654321</cppa:PartyId>
        <CPPId id="companyb-cpp987654321" cppa:version="1.0"/>
    </tp:RespondingParty>
    <tp:BPSSBusinessDocumentName name="CPA_Offer_Doc"/>
    <ExpirationDate>2002-12-20T00:00:00Z</ExpirationDate>
    <tp:BusinessDocuments>
        <CPATemplateDoc>
            <NDD>
                <Uri>http://www.companya.com/proposedncpa/ncpa.xml</Uri>
            </NDD>
            <CPATemplate>
                <Uri>http://www.companya.com/proposedcpa/companya-companyb-cpa1234.xml</Uri>
            </CPATemplate>
        </CPATemplateDoc>
    </tp:BusinessDocuments>
    <tp:NegotiationContent>
        <tp:AcceptedItem xpath="/"/>
    </tp:NegotiationContent>
</tp:NegotiationMessage>
```

Example of Counter-Offer Message Instance Document

The text file for the example of the counter-offer *Message* instance document is available at:

THIS APPENDIX HAS BEEN REPLACED IN FULL.

Negotiation Message Counter Offer

```
<?xml version="1.0" encoding="UTF-8"?>
<tp:NegotiationMessage xmlns:tp="http://www.oasis-open.org/committees/ebxml-cppa/schema/cpa-negot-1_0.xsd"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:cppa="http://www.oasis-open.org/committees/ebxml-
cppa/schema/cpp-cpa-2_0.xsd" xsi:schemaLocation="http://www.oasis-open.org/committees/ebxml-cppa/schema/cpa-
```

```
negot-1_0.xsd negotiationMsg.xsd" businessMsgId="busMsg002" binding="false" inresponseTo="busMsg001"
negotiationDialogId="negotDialog001" offerId="offer001" messageType="CounterOffer">
    <tp:NCPA uri="http://www.companya.com/ncpa/myncpa.xml"/>
    <tp:CPATemplateId id="uri:companyA-and-companyB-CPA1" cppa:version="1.0"/>
    <cppa:SecurityDetails cppa:securityId="ID">
        <cppa:SecurityPolicy/>
    </cppa:SecurityDetails>
    <tp:InitiatingParty>
        <cppa:PartyId cppa:type="urn:oasis:names:tc:ebxml-cppa:partyid-type:duns">123456789</cppa:PartyId>
        <CPPId id="companya-cpp123456789" cppa:version="1.0"/>
    </tp:InitiatingParty>
    <tp:RespondingParty>
        <cppa:PartyId cppa:type="urn:oasis:names:tc:ebxml-cppa:partyid-type:duns">987654321</cppa:PartyId>
        <CPPId id="companyb-cpp987654321" cppa:version="1.0"/>
    </tp:RespondingParty>
    <tp:BPSSBusinessDocumentName name="CPA_Counter_Offer_Doc"/>
    <ExpirationDate>2003-05-20T00:00:00Z</ExpirationDate>
    <tp:BusinessDocuments>
        <CPATemplateDoc>
            <NDD>
                <Uri>http://www.companyb.com/proposedncpa/ncpa.xml</Uri>
            </NDD>
            <CPATemplate>
                <Uri>http://www.companyb.com/proposedcpa/companya-companyb-cpa1234.xml</Uri>
            </CPATemplate>
        </CPATemplateDoc>
    </tp:BusinessDocuments>
    <tp:Offer-CounterOffer-Acceptance-Time>2003-03-20T00:00:00Z</tp:Offer-CounterOffer-Acceptance-Time>
    <tp:NegotiationContent>
        <tp:AcceptedItem xpath="/CollaborationProtocolAgreement/PartyInfo[0]"/>
        <tp:AcceptedItem xpath="/CollaborationProtocolAgreement/PartyInfo[1]/Certificate"/>
        <tp:AcceptedItem xpath="/CollaborationProtocolAgreement/PartyInfo[1]/SecurityDetails"/>
        <tp:AcceptedItem xpath="/CollaborationProtocolAgreement/PartyInfo[1]/DeliveryChannel"/>
        <tp:AcceptedItem xpath="/CollaborationProtocolAgreement/PartyInfo[1]/Transport"/>
        <tp:AcceptedItem xpath="/CollaborationProtocolAgreement/PartyInfo[1]/DocExchange"/>
        <tp:UpdatedItem
xpath="/CollaborationProtocolAggreement/PartyInfo[1]/CollaborationRole/ServiceBinding/Cansend[0]/ThisPartyActionBindin
g/BusinessTransactionCharacteristics@isNonRepudiationRequired" originalValue="true" proposedValue="false"
itemStatus="Preferred"/>
    </tp:NegotiationContent>
</tp:NegotiationMessage>
```

Appendix H Glossary of Terms

This appendix contains definitions of terms created by this specification. For definitions of terms created by the CPPA Specification[ebCPP] and related terms that are part of the general ebXML vocabulary, see [ebCPP].

CPA **Negotiation Process:** The process by which a *Collaboration Protocol Agreement (CPA)* is formed based on information provided by two *Parties* interested doing *Business*. The *Negotiation Process* includes the *Negotiation Protocol*, defined in this specification, and the private negotiation process at each *Party*.

CPA **Template:** A *CPA Template* is a *CPA* with open fields. The schema for a *CPA Template* is the normal *CPP-CPA* schema. The means of identifying open fields in the *CPA Template* is defined in this specification.

Negotiation BPSS Instance Document: The XML instance document that defines the *Negotiation-Protocol* choreography. This XML instance document conforms to the ebXML *Business Process Specification Schema* specification[ebBPSS].

Negotiation *CPA* (*NCPA*): The *CPA* that governs the *Negotiation Protocol*.

Negotiation Descriptor Document (*NDD*): A *Negotiation Descriptor Document (NDD)* describes what is negotiable in a *CPP* or a *CPA Template*.

Negotiation Dialog: A single instance of the *Negotiation Protocol* that negotiates one *CPA* from the initial proposal until the *CPA* is successfully completed or the negotiation terminates without success.

Negotiation-Dialog Identifier: A unique identifier that distinguishes each *Negotiation Dialog* from all others that may be in progress between two *Parties*.

Negotiation *Message*: The *Negotiation Protocol* consists of exchanges of *Messages* that contain the details of offers and counter offers. This specification defines the schema and semantics of each *Message*.

Negotiation Protocol: The *Negotiation Protocol* defines the exchange of data between both parties in the negotiation (and perhaps with a negotiation service). The format of these *Messages* and the choreography of their exchanges are defined by a *Negotiation CPA* and its corresponding BPSS instance document.

Offer Identifier: The *Offer Identifier* is a unique identifier associated with each offer and counter offer.