ebXML　**解説書**

第 5 部

ebXML 交換協定

平成 14 年 3 月

(財)日本情報処理開発協会 電子商取引推進センター

この解説書は、経済産業省委託「平成 13 年度 精算・調達・運用支援統合情報システムに関する調査研究」事業の成果です。

はじめに

1999 年 11 月に、国際標準 EDI（UN／EDIFACT）の利用グループの支援を受けた国連 CEFACT（Center for Trade Facilitation and eBusiness）と、先進的 IT ベンダーのコンソーシアムである OASIS (Organisation for Advanced Structured Information Standards)の協業で始められた ebXML イニシャチブは、2001 年 5 月、今後の電子ビジネスコラボレーション実現のフレームワークとなる一連の仕様（ebXML 仕様）の第 1 版を完成し公表した。

ebXML 仕様は、従来のレガシーEDI や WEB－EDI を XML 化するに留まらず、取引企業同士のそれぞれのアプリケーションが、情報交換により合意されたビジネスプロセスを遂行してビジネス目標を達成する、すなわち電子ビジネスコラボレーションを実現させるために必要な標準仕様を定めている。

今後、当該標準仕様は、IT ベンダーの戦略的製品やサービスに取り入れられるとともに、ユーザー業界においてはビジネスプロセス改善の仕組みに採用されて行くことが期待されている。

(財)日本情報処理開発協会では経済産業省の委託を受けて、2001 年 5 月に公表された ebXML 仕様を中心に、電子商取引推進協議会の平成 13 年度 XML／EDI 標準化専門委員会の討議結果を反映し、次の 6 部からなる解説書を作成した。

第１部 ebXML 概要
第２部 ebXML ビジネスプロセス
第３部 ebXML 情報構成要素
第４部 ebXML レジストリ・リポジトリ
第５部 ebXML 交換協定
第６部 ebXML 通信仕様

なお、ebXML 仕様は、2001 年 5 月以降、第 2 章－第 3 章関連は UN／CEFACT、第 4 章－第 6 章関連は OASIS が仕様の改訂・保守を継続しており、ebXML 仕様の実装においては該当組織より発表されている最新版の仕様を参照されることを推奨する。

平成 14 年 3 月
財団法人日本情報処理開発協会
電子商取引推進センター

第 5 部 ebXML 交換協定　序文

「ebXML 交換協定」は、ebXML イニシャチブにより作成されたコラボレーションプロトコルプロファイルおよびコラボレーションプロトコル合意仕様を掲載する。

<参照 ebXML 仕様書>
Collaboration Protocol Profile and Collaboration Protocol Agreement Specification V1.0
2001 年 5 月 10 日

コラボレーションプロトコルプロファイル
および
コラボレーションプロトコル合意仕様
改訂 **1.0**

ebXML TPA チーム

2001 年 5 月 10 日

技術検証 ECOM XML/EDI 標準化専門委員会

本書の位置付け

本書では、電子ビジネスコミュニティ向けの ebXML 仕様を規定している。

本書は自由に配布可能である。

本書はインターネット・ソサイアティーの標準 RFC 形式に準拠している。

本バージョン

http://www.ebxml.org/specs/ebCCP.pdf

最新のバージョン

http://www.ebxml.org/specs/ebCCP.pdf

ebXML 参加者

以下の各氏には、本書の開発に際して多大なるご協力を賜った。ここに感謝の意を表す。

David Burdett, CommerceOne
Tim Chiou, United World Chinese Commercial Bank
Chris Ferris, Sun
Scott Hinkelman, IBM
Maryann Hondo, IBM
Sam Hunting, ECOM XML
John Ibbotson, IBM
Kenji Itoh, JASTPRO
Ravi Kacker, eXcelon Corp.
Thomas Limanek, iPlanet
Daniel Ling, VCHEQ
Henry Lowe, OMG
Dale Moberg, Cyclone Commerce
Duane Nickull, XMLGlobal Technologies
Stefano Pogliani, Sun
Rebecca Reed, Mercator
Karsten Riemer, Sun
Marty Sachs, IBM
Yukinori Saito, ECOM
Tony Weida, Edifecs

目次

はじめに

1.1 本書の総括

『ebXML ビジネスプロセス仕様スキーマ』[ebBPSS]で定義されているように、*取引当事者*は他の*取引当事者*との間で*取引トランザクション*を実行するエンティティである。他の*当事者*との間で電子的に*メッセージ*交換する各*当事者*の能力（商業・*ビジネス*的な能力と技術的な能力の両方）は、*取引当事者プロファイル（TPP; Trading-Partner Profile*）という文書に記述する。*当事者*双方の間で合意された相互作用は、*取引当事者合意書（TPA; Trading-Partner Agreement*）という文書に記述しする。*TPA* は、*当事者*双方の *TPP* の共通部分を抽出して作成してもよい。

*当事者*の*メッセージ*交換能力は、*TPP* 内の*コラボレーションプロトコルプロファイル（CPP; Collaboration-Protocol Profile*）に記述する。*当事者*双方の間の*メッセージ*交換についての合意は、*TPA* 内の*コラボレーションプロトコル合意書（CPA; Collaboration-Protocol Agreement*）に記述する。*CPP* および *CPA* に含まれる内容は、対象となっている電子*取引コラボレーション*の実行時に用いられる、転送、メッセージング、セキュリティの拘束条件、および*当事者*間のやりとりを定義している*プロセス仕様*文書の参照情報である。

本書は、*コラボレーションプロトコルプロファイル（CPP*）および*コラボレーションプロトコル合意書（CPA*）の詳細な定義を含む。

本書は、ebXML の仕様一式の 1 つである。ebXML 仕様および仕様相互の関係については、『ebXML テクニカルアーキテクチャ仕様』[ebTA]に概要が示されている。

本書の構成は、次の通りである。

- 5 節で、本書の目的を定義する。
- 6 節で、システムの概要を示す。
- 7 節で、*CPP* の定義を示し、全体の構造およびすべての必要なフィールドを特定する。
- 8 節で、CPA の定義を示す。
- 付録で、XML *CPP* および *CPA* 文書（非規範的）、DTD（規範的）、DTD と同等な XML スキーマ文書、*CPP* および *CPA*（規範的）内の情報の形式、および 2 つの *CPP* から *CPA* を作成する場合の例を示す。

1.2 文書規約

*斜体の*用語は、ebXML 用語集[ebGLOSS]で定義されているものである。***太字の斜体***の用語は、XML CPP または CPA の定義内容の要素、属性、またはその両方を表現している。

本書で「注意:」と書かれた部分は、非規範的な説明または提案である。この部分は仕様の要件ではない。

外部文書の参照は、角括弧で囲まれた大文字の英数字（たとえば、[RFC2396]）で示す。これらの関連文書は、9 節「関連文書」で列挙する。

しなければならない (MUST)、してはならない (MUST NOT)、要求される (REQUIRED)、することになる (SHALL)、することはない (SHALL NOT)、する必要がある (SHOULD)、しないほうがよい (SHOULD NOT)、推奨される (RECOMMENDED)、場合がある (MAY)、選択できる (OPTIONAL)といったキーワードが使用された場合は、RFC 2119 [Bra97] における定義に沿って解釈されるものとする。

> 注意: ベンダは、多重度付きの要素のサポートについて十分に考慮する必要がある。そのような要素をサポートするということは、要素を、定義された機能に従って適切に処理することを意味する。したがって、認識するだけで無視することは許されない。*当事者*によっては、これらの要素を一部の *CPP* または *CPA* でのみ使用し、他の文書では使用しない場合もある。すべてのベンダが実装する必要のあるパラメータまたは運用モードを定義するために用いられる要素もある。主な実行時の機能を表現するオプションの要素（さまざまな代替の通信プロトコルまたはセキュリティ機能など）を、プラグインを用いて実装する方が適切な場合もある。したがって、当事者によっては、すべての機能ではなく必要な機能だけを利用してもよい。

1.3 XML スキーマの利用

CPP および *CPA* のスキーマは、『XML スキーマ仕様』[XMLSCHEMA-1,XMLSCHEMA-2]のプロポーザル案に基づいている。XML スキーマがプロポーザル案に位置付けられる際、同仕様およびそのスキーマの変更がいくつか必要となる。これらの変更は、付録 A で示すように、最新のスキーマ文書において XML コメントで表される。

1.4 本書のバージョン

本書を修正する場合は、新しいバージョン番号をつけることとする。XML スキーマ文書の ***Schema*** 要素の ***version*** 属性の値は、仕様のバージョンと同じとする。

1.5 定義

本書中の技術用語は、『ebXML 用語集』[ebGLOSS]で定義されている。

1.6 対象読者

本書の対象となる読者は、ebXML サービスの実装者、および電子*ビジネス*で使用されるミドルウェア、アプリケーションソフトウェアの設計者や開発者である。さらに、個々の企業における *CPP* および *CPA* の作成担当者も対象となる。

1.7 前提条件

読者は、XML を理解し、電子ビジネス（eBusiness）の概念にも精通していることが期待される。

1.8 関連文書

関連文書としては、次の話題に関する ebXML 仕様書がある。

- ebXML テクニカルアーキテクチャ仕様[ebTA]
- ebXML メッセージ取扱サービス仕様[ebMS]
- ebXML ビジネスプロセス仕様スキーマ[ebBPSS]
- ebXML 用語集 [ebGLOSS]
- ebXML コア構成要素および取引文書の概要[ccOVER]
- ebXML レジストリサービス仕様[ebRS]

関連文書の一覧全体は、9 節を参照のこと。

設計目的

本書の目的は、*当事者*双方の相互運用性を保証することである。ただし、さまざまなベンダのアプリケーションソフトウェアおよび実行時支援ソフトウェアを*当事者*双方が調達する場合もある。CPP により、当事者のメッセージ交換機能および CPP がサポートする*取引コラボレーション*が定義される。*CPA* では、*当事者*双方が*取引コラボレーション*の実行時に相互作用する方法を定義する。*当事者*双方は、同じ CPA を用いて、実行時システムを構成することとする。同じベンダ製の実行時システムを取得しているかどうかに関わらず、*メッセージ*交換ができるように互換性を保って構成されていることが前提になる。この設定プロセスは、*CPA* を読み取り設定プロセスを実行するツールを用いて自動化してもよい。

*当事者*間のやりとりを直接サポートする場合だけでなく、ポータル、ブローカなどの仲介者を通して*当事者*双方の間のやりとりをサポートする場合に本書を使用してもよい。この初期バージョンの仕様では、取引*当事者*間の *CPA* だけでなく、*当事者*と仲介者の間の *CPA* を作成することで実現してもよい。*当事者*と仲介者の間のやりとりに必要な機能は、*当事者*と仲介者の間の *CPA* に記述する。また、取引*当事者*間のやりとりに必要な機能は、*当事者*間の *CPA* に記述する。

取引*当事者*の *CPP* の共通部分から *CPA* を合成できるようにすることも本書の目的の 1 つである。その際、合成される *CPA* は、*当事者*双方の間で共通または互換性のある要素のみを含むこととする。一方、可変な量（エラーの再試行回数など）は*当事者*双方の間で交渉する。*CPP* および *CPA* のスキーマ設計によって、この合成および交渉のプロセスが促進される。ただし、合成プロセスおよび交渉プロセス自体は、本書の範囲外である。付録 F に、この点に関する非規範的な議論を示す。

既存の EDI ベースのアプリケーションや他のレガシーアプリケーションの、ebXML 仕様に基づくプラットフォームへの移行を促進することもまた、本書の目的である。特に、*CPP* および *CPA* は、『X12 838 取引当事者プロファイル』に基づくアプリケーションを、取引関係を構築しその上で取引を実行するためのより自動化された手段に移行するプロセスの一部として用いられる。

システムの概要

1.9 本書の対象範囲

*当事者*双方の間の情報交換の際には、個々の*当事者*が相手がサポートしている*取引*/*コラボレーション*、その取引コラボレーションにおける相手の役割、および相手が*メッセージ*を送受信する方法に関する技術的な詳細を知る必要がある。場合によっては、この詳細について*当事者*双方が合意に達する必要がある。

取引/*コラボレーション*のコンテキストにおいて個々の*当事者*が情報を交換する方法は、*コラボレーションプロトコルプロファイル（CPP）*に記述できる。また、当事者の間の合意は、*コラボレーションプロトコル合意書（CPA）*として表現できる。

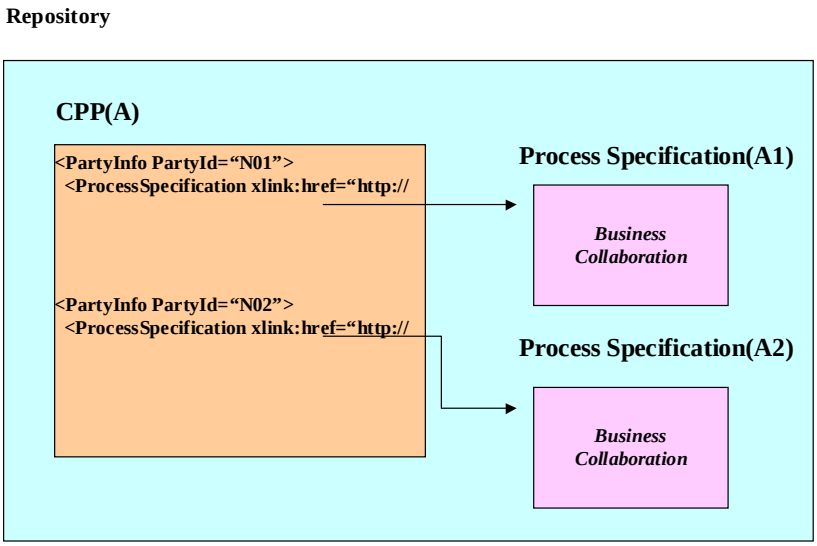
*当事者*は、1 つの *CPP* において記述される場合もあり、また、例えば *CPP* がサポートする別の*取引*/*コラボレーション*、世界の異なる地域での運用、あるいは組織の異なった部署などを記述する複数の *CPP* を作成する場合もある。

*取引*を希望する*当事者*が適切な取引当事者となり得る相手を見つけることができるようにするために、*CPP* を、ebXML レジストリ[ebRS]に含まれているようなリポジトリに格納してもよい。*当事者*は、リポジトリの仕様の一部として提供されている発見プロセスを用いて、リポジトリを使用して*取引当事者*の存在を発見してもよい。

2 つの当事者間のやりとりを定義する文書は、『ebXML ビジネスプロセス仕様スキーマ』[ebBPSS]に準拠する*プロセス仕様*文書である。
CPP および *CPA* には、この*プロセス仕様*文書の参照が含まれる。*プロセス仕様*文書は、ebXML レジストリなどのリポジトリに格納してもよい。取引コラボレーションの別の記述に関しては、該当する節の注意事項を参照のこと。

図 1 は、ebXML レジストリにおける、*CPP* と、A1 および A2 という 2 つの*プロセス仕様*文書の関係を示している。左側の「A」という *CPP* には、異なる*当事者*として表わされる 1 つの企業内の 2 つの部門に関する情報が含まれている。右側の 2 つの四角は、*プロセス仕様*文書である。*CPP* 内の ***PartyInfo*** 要素はそれぞれ、これらのプロセス仕様文書のうちの 1 つへの参照を含んでいる。この情報によって、*当事者*が実行できる*取引*/*コラボレーション*が特定される。

図1: ebXMLレジストリのCPPとビジネスプロセス仕様の構造



本書では、*CPP* および *CPA* を電子的に作成するための、マークアップ言語の語彙が定義される。*CPP* および *CPA* は、XML 文書である。本書の付録で、サンプルの *CPP*、サンプルの *CPA*、DTD、およびそれらに対応する XML スキーマ文書を示す。

CPP で、個々の*当事者*の能力を記述する。CPA で、特定の*取引コラボレーション*の実行のために使用することが*当事者*双方によって合意された機能を記述する。これらの *CPA* によって、*取引文書*を電子的に当事者間で交換するための、「情報技術契約条件」が定義される。*CPA* で示される情報の内容は、電子データ交換（EDI）の『*取引当事者合意書*』（TPA）に含まれる可能性のある情報技術仕様と類似している。ただし、これらの *CPA* は紙の文書ではない。むしろ、これらの文書は、*当事者*のサイトにあるコンピュータによって処理し、必要な*ビジネス*情報の交換を設定し実行できる電子文書である。取引合意書の契約条件についての「法律的な」記述は、本書の範囲外であり、したがって、*CPP* および *CPA* には含まれない。

1 つの企業を複数の*当事者*として表現してもよい。たとえば、本社の調達部門と製造現場の調達部門を別の*当事者*として表現することもできる。次に、別々の*当事者*として表現された部門をすべて包含する *CPP* を構築してもよい。その *CPP* の中で、これらの部門をそれぞれ異なる ***PartyInfo*** 要素で表現することになる。

一般に、*CPA* の*当事者*は、クライアントとサーバの両方の特徴を持つ可能性がある。クライアントはサービスを依頼し、サーバは依頼元の*当事者*にサービスを提供する。アプリケーションによっては、一方の*当事者*はサービスを依頼するのみであり、他方の*当事者*はサービスを提供するのみである場合もある。そのようなアプリケーションの場合、伝統的なクライアント/サーバアプリケーションとある程度類似している。しかし、他のアプリケーションでは、両方の*当事者*がそれぞれ相手にサービスを依頼してもよい。その場合、2 *当事者*間の関連は、クライアント/サーバ関係ではなくピアツーピア関係として記述できる。

1.10 2 つの CPP から 1 つの CPA を作成する

本節では、取引相手を発見し、2 *当事者*の *CPP* から *CPA* を作成するプロセスを要約する。一般に、本節は、可能な手順の概要を示しているだけなので、規範的な仕様とは見なさないようにして欲しい。詳細は、付録 F「2 つの CPP から CPA を合成する」を参照のこと。

図 2 に、*CPP* の作成手順を示す。*当事者* A が発見プロセスのリポジトリに配置する情報を表の形にし、その情報を含む *CPP* を構築し、*当事者*に関する追加情報と合わせて ebXML レジストリなどのリポジトリに入力する。この追加情報には、この*当事者*が関わっている取引に関する記述が含まれている場合がある。*当事者* A の情報がリポジトリに格納されると、そのリポジトリの発見サービスを用いて、相手が*当事者* A を発見できるようになる。

図2: コラボレーションプロトコルプロファイル (CPP)の概要

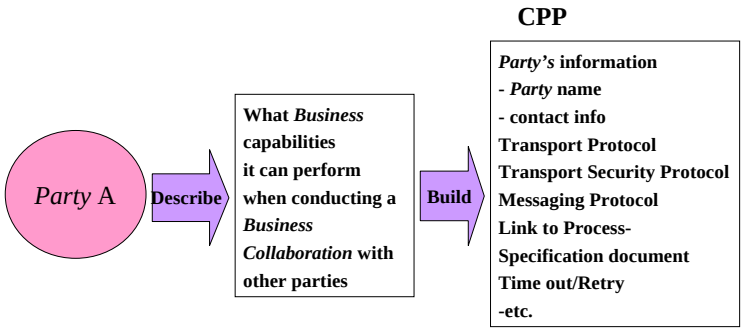


図 3 では、*当事者* A と *当事者* B が、それぞれの *CPP* 内の情報の共通部分を抽出することで
、同一の *CPA* を共同で構築している。構築された *CPA* には、*当事者*双方が*取引コラボレーション*の実行に際してどのように振舞うかが定義されている。

図3: コラボレーションプロトコル合意書 (*CPA*)の概要

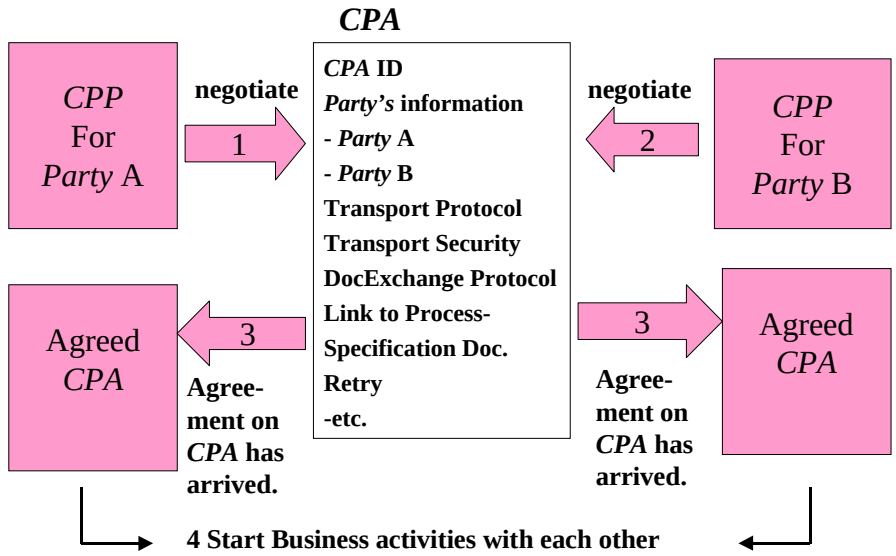


図 4 に、プロセスの全体を示す。各ステップが左側に列挙されている。プロセスの最後には、*当事者*双方が、合意に至った同一の *CPA* から自社のシステムを構成している。この時点で取引を実行する準備が完了する。

図4: ebXMLレジストリのあるCPP/CPAのアーキテクチャの概要

1. Any *Party* may register its CPPs to an ebXML Registry.

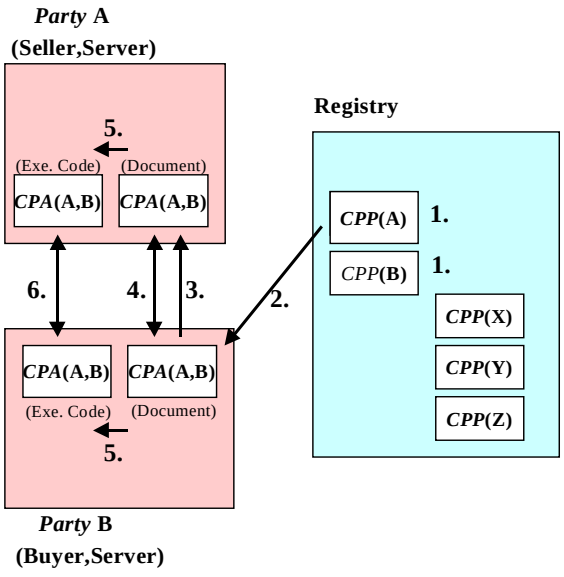
2. *Party* B discovers trading partner A (Seller) by searching in the Registry and downloads *CPP*(A) to *Party* B's server.

3. *Party* B creates *CPA*(A,B) and sends *CPA*(A,B) to *Party* A.

4. *Parties* A and B negotiate and store identical copies of the completed *CPA* as a document in both servers. This process is done manually or automatically.

5. *Parties* A and B configure their run-time systems with the information in the *CPA*.

6. *Parties* A and B do business under the new *CPA*.



注意: 本書では、ebXML または他のレジストリに登録された *CPP* は、レジストリによって割り当てられたグローバルに一意な識別子によって参照されると仮定している。識別子は、同じ*当事者*に属する複数の *CPP* を区別するために使用してもよい。詳細は、7.1 節を参照のこと。

1.11 CPA の動作原理

CPA によって、*当事者*の間の有効で可視（すなわち強行可能）なすべてのやりとり、およびそれらのやりとりの実行方法を記述する。この記述は、個々の*当事者*によって実行される内部プロセスとは独立である。個々の*当事者*はそれぞれの内部プロセスを実行し、*CPA* および *プロセス仕様*文書で記述される *取引/コラボレーション*とのインタフェースを取得する。*CPA* は、*当事者*の内部プロセスについての詳細を相手に開示しない。*CPA* の目的は、人間が容易に理解でき、かつコンピュータによる実現が可能なほど十分精密な、高いレベルの仕様を提供することにある。

CPA 情報を使用して、*当事者*のシステム構成が設定され、その結果、*取引/コラボレーション*の実行における*メッセージ*交換が可能になる。通常、*メッセージ*交換を実行するか、または*当事者*間のやりとりをサポートするためのソフトウェアは、任意の*取引/コラボレーション*をサポートできるミドルウェアである。このミドルウェアのコンポーネントの 1 つとして、ebXML *メッセージ取扱*サービスハンドラ[ebMS]がある。本書では、そのようなミドルウェアを、「実行時システム」または「実行時ソフトウェア」と呼ぶ。

参照されている *CPA* および *プロセス仕様*文書によって、*当事者*間のやり取りが定義される。やり取りは、*プロセス仕様*文書の*バイナリコラボレーション*コンポーネントの定義に従い、*ビジネス*の単一ユニットを表す。やり取りは、一方の*当事者*からの依頼*メッセージ*および相手からの 0 または 1 つの応答*メッセージ*を表す、1 つ以上の*取引/トランザクション*から成る。*プロセス仕様*文書では、特に、各*取引/トランザクション*内の依頼メッセージと応答*メッセージ*、および*取引/トランザクション*の順序が定義される。詳細は、[ebBPSS]を参照のこと。

CPA は、実際には、複数の *プロセス仕様*文書を参照してもよい。*CPA* が *プロセス仕様*文書を参照する際には、それぞれの *プロセス仕様*文書で別々のやり取りの型を定義する。1 つのやり取りには、1 つの *プロセス仕様*文書しか含まれない。

新しい*ビジネス*のユニットが開始されるごとに新しいやり取りが開始される。*取引/コラボレーション*によって、やり取りの終了時点も指定される。*当事者* A と *当事者* B の間の *CPA* の見地から見ると、*当事者* A が最初の依頼メッセージを *当事者* B に送信した時点でやり取りが開始される。一方、*当事者* B においては、*当事者* A から *ビジネス*ユニットの最初の依頼を受信した時点でやり取りが開始される。そして、*当事者*が *ビジネス*のそのユニットを終了した時点でやり取りが終了する。

注意: 実行時システムは、やり取りの開始と終了を依頼するためのインタフェースを提供する必要がある。

1.12 CPA の実装場所

概念的には、各*当事者*サイトの*企業間*（B2B）サーバ上に、*CPA* および *プロセス仕様*文書が実装される。B2B サーバには、実行時ソフトウェア（相手*当事者*との通信をサポートするミドルウェア）、*CPA* で指定された機能の実行、各*当事者*のバックエンドプロセスとのインタフェース、監査や復旧などのための*当事者*間のやりとりのログ記録が含まれる。ミドルウェアは、*当事者*間の *ビジネス*の単一ユニットを具体化するために長期実行やり取りという概念をサポートしている場合がある。2 *当事者*サイトにある企業間処理のシステムを構成するために、個々の*当事者*サイトにある、*CPA* および *プロセス仕様*文書内の情報が実行時システムにインストールされる。静的な情報はローカルデータベースに記録してもよい。また、*CPA* および *プロセス仕様*文書内の他の情報は、*CPA* のサポートに必要なコードの生成またはカスタマイズ用に使用してもよい。

注意: *CPP/CPA* の意味情報と XML の構文の両方を理解するグラフィカルな *CPP/CPA* 編集ツールを提供することもできる。同時に、本書内の定義に従い、*当事者*サイトで、*CPA* の実行、規則の強制、およびバックエンドプロセスとのインタフェースのために必要なコードを自動的に生成できるようにすることが重要である。

1.13 定義および適用範囲

本書では、*CPP* および *CPA* の XML 文書の内容を定義し説明する。適用範囲は、これらの定義に制限される。2 つの *CPP* から *CPA* を合成する方法は定義しない。また、*CPP* および *CPA* の実行時サポートに関連した情報も定義していない。実行時サポートに関する提案および推奨は、非規範的に述べられている。そうした情報は、「注意」として述べられており、*CPP* および *CPA* の定義を明確にするために付記されている。本書への適合性については、10 節を参照のこと。

注意: 本書の範囲は、*CPP* および *CPA* の内容の定義に制限されている。したがって、単に、ここで定義されている DTD および XML スキーマに準拠した *CPP* または *CPA* 文書を生成することにより、本書に準拠することも可能である。ただし、本書の価値は、*CPA* 内の情報をガイドとして*当事者*間の電子商取引をサポートする実行時システムを実現できることにある、という点を理解することは重要である。

CPP の定義

CPP によって、相手*当事者*との間で電子*ビジネス*を行う*当事者*の能力が定義される。定義される能力には、サポートしている通信およびメッセージ取扱プロトコルなどの技術力、およびサポートしている*取引/コラボレーション*で示される*ビジネス*能力の両方が含まれる。

本節では、個々の XML 要素について述べ、*CPP* の詳細を定義し解説する。解説は、XML フラグメントの一部を用いて示される。DTD および XML スキーマの詳細はそれぞれ付録 C および付録 D、*CPP* 文書のサンプルは付録 A を参照して頂きたい。

CPP の、***ProcessSpecification 要素、DeliveryChannel 要素、DocExchange 要素、および Transport 要素***によって、ビジネスの 1 つのユニット（やり取り）の処理が記述される。これらの要素によって、多層通信モデルとよく似た多層構造が作成される。本節の残りの部分では、上記の要素および対応する実行時処理の両方を説明する。

プロセス仕様層 - *プロセス仕様*層では、*当事者*間の*ビジネス*合意の中心部分を定義する。すなわち、*CPA* の*当事者*が互いに依頼できるサービス（*取引/トランザクション*）、依頼の順序を規定する遷移規則が指定される。この層は、*CPP* および *CPA* で参照されている別の*プロセス仕様*文書によって定義される。

配信経路 - 配信経路によって、*当事者のメッセージ*受信特性が記述される。受信特性は、文書交換の定義および転送の定義から構成される。*CPP* に複数の配信経路を定義していてもよい。

文書交換層 - 文書交換層では、*当事者の プロセス仕様*層からの*取引文書*を受け付け、指定がある場合は暗号化したりデジタル署名を追加したりし、トランスポート層に渡し相手*当事者*に送信する。受信*メッセージ*の場合、逆のステップを実行する。文書交換層用に選択されたオプションは、トランスポート層用に選択されたオプションを補完するものである。たとえば、*メッセージ*セキュリティが必要であるが選択された転送プロトコルで*メッセージ*暗号化が提供されていない場合は、*メッセージ*セキュリティを文書交換層で指定する必要がある。*当事者*間で*メッセージ*交換を実現するためのプロトコルは、『ebXML *メッセージ取扱サービス*仕様』[ebMS]や他の類似したメッセージ取扱サービスによって定義される。

トランスポート層 - トランスポート層では、選択した転送プロトコルを用いて*メッセージ*を配信する。選択されたプロトコルは、文書交換層用に選択されたオプションに影響する。たとえば、暗号化および認証機能を提供するトランスポート層プロトコルもあれば、そのような機能を提供しないトランスポート層プロトコルもある。

CPP に包含される機能層では、*取引文書*の最大積載量については扱っていないということに注意すること。

1.14 CPP インスタンス文書のグローバル識別子

CPP を ebXML または他のレジストリに配置する際、レジストリによって、グローバルに一意な識別子（GUID）がメタデータの一部として割り当てられる。GUID は、同じ*当事者*に属する *CPP* を区別するために使用してもよい。

> 注意: レジストリでは、*CPP* に GUID を挿入できない。一般に、レジストリは、文書の内容を変更しない。さらに、*CPP* に署名を入れてもよい。また、署名入りの CPP を変更した場合は、その署名は無効になる。

1.15 SchemaLocation 属性

2000 年 10 月 24 日付けでプロポーザル候補になっていた W3C の『XML スキーマ仕様』[XMLSCHEMA-1, XMLSCHEMA-2]は、2001 年 3 月 30 日付けでプロポーザル案になった。本書の作成時点で入手可能な、スキーマ検証用の XML の構文解析機（パーサ）をサポートするツールの多く（例外も少なくないが）は、その XML スキーマ仕様のプロポーザル候補の内容もサポートしている。

検証用パーサおよびスキーマ検証ツールが、ebXML の CPP および CPA 文書を正しく処理できるようするために、ebXML の TP チームは、W3C の XML スキーマ仕様のプロポーザル候補に準拠したスキーマを生成する必要がある。CPP および CPA 編集ツールの実装においては、文書の検証に用いるスキーマ文書のロケーション URI を検証用パーサに通知するために、文書のルート要素に schemaLocation 属性を付けることが強く推奨される。schemaLocation 属性を指定しない場合は、これらの文書を検証するために必要な他のツールとの間で相互運用性の問題が発生する場合がある。

W3C のプロポーザルとして XML スキーマ仕様が採用される場合などは、そのプロポーザルに準拠するために必要な更新を含む改定版の CPP/CPA スキーマを作成することとする。

schemaLocation 属性の使用例を次に示す。

```
<CollaborationProtocolAgreement
      xmlns="http://www.ebxml.org/namespaces/tradePartner"
      xmlns:xsi="http://www.w3.org/2000/10/XMLSchema-instance"
      xsi:schemaLocation="http://www.ebxml.org/namespaces/tradePartner
            http://ebxml.org/project_teams/trade_partner/cpp-cpa-10.xsd"
      ...
      >
      ...
</CollaborationProtocolAgreement>
```

1.16 CPP の構造

CPP の全体の構造は次の通りである。特に指定がない場合は、*CPP* の要素は、ここで示されている順序に並んでいなければならない。以降の節では、個々の要素について詳細に述べる。

```
<CollaborationProtocolProfile
      xmlns="http://www.ebxml.org/namespaces/tradePartner"
      xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
      xmlns:xlink="http://www.w3.org/1999/xlink"
      version="1.1">
      <PartyInfo>  <!--one or more-->
       ...
      </PartyInfo>
      <Packaging id="ID"> <!--one or more-->
            ...
      <Packaging>
      <ds:Signature>  <!--zero or one-->
      ...
      </ds:Signature>
      <Comment>text</Comment> <!--zero or more-->
      </CollaborationProtocolProfile>
```

1.17 CollaborationProtocolProfile 要素

CollaborationProtocolProfile 要素は、*CPP* XML 文書のルート要素である。

基本的な文書に対する、必須の [XML]名前空間（Namespace）[XMLNS]宣言は、次の通りである。

- The default namespace: xmlns="http://www.ebxml.org/namespaces/tradePartner",
- XML Digital Signature namespace: xmlns:ds="http://www.w3.org/2000/09/xmldsig#",
- and the XLINK namespace: xmlns:xlink="http://www.w3.org/1999/xlink".

さらに、**CollaborationProtocolProfile** 要素には、*CPP* のバージョンを示す暗黙の ***version*** 属性が含まれている。この属性の目的は、企業の *CPP* のインスタンスに、バージョン管理機能を持たせることにある。バージョン属性の値は、「1.0」、「2.3」などの数値を表す文字列表現である必要がある。バージョン文字列の値は、公開済みの *CPP* 文書が変更された際には、必ず変更する必要がある。

注意: バージョン識別子の値の割り当て方法は、実装に委ねられている。

CollaborationProtocolProfile 要素は、次の要素から成ることとする。

- 1 つ以上の必須の ***PartyInfo*** 要素。***PartyInfo*** 要素によって、*CPP* によって能力が記述される組織（または組織の部分）が識別される。
- 必須の ***Packaging*** 要素

- 0 個以上の ***ds:Signature*** 要素。***ds:Signature*** 要素には、*CPP* 文書に署名するデジタル署名が含まれている。
- 0 個以上の ***Comment*** 要素。

CPP 文書には、文書が変更されていないこと（一貫性）を保証する手段、および文書の著者を認証する手段を提供するために、デジタル署名を付与してもよい。デジタル署名入りの *CPP* は、W3C/IETF 共同の『XML デジタル署名仕様』[XMLDSIG]に準拠した技術を用いて署名されることとする。

1.18 PartyInfo 要素

PartyInfo 要素は、*CPP* によって能力を記述する対象となる組織を特定し、その*当事者*についてのすべての詳細情報を包含する。組織をさまざまな特徴を持つ部門の集合として表現したいと考える場合は、1 つの *CPP* に複数の ***PartyInfo*** 要素を付与してもよい。***PartyInfo*** の個々のサブ要素については、以降の節で議論する。***PartyInfo*** 要素の全体構造は、次の通りである。

```
<PartyInfo>
      <PartyId type="...">  <!--one or more-->
            ...
      </PartyId>
      <PartyRef xlink:type="...", xlink:href="..."/>
      <CollaborationRole>   <!--one or more-->
            ...
      </CollaborationRole>
      <Certificate>  <!--one or more-->
            ...
      </Certificate>
      <DeliveryChannel>  <!--one or more-->
      ...
      </DeliveryChannel>
      <Transport>  <!--one or more-->
      ...
      </Transport>
      <DocExchange>  <!--one or more-->
      ...
      </DocExchange>
</PartyInfo>
```

PartyInfo 要素は、次に示す子要素から成る。

- 1 つ以上の必須の ***PartyId*** 要素。***PartyId*** 要素は、組織の論理識別子として使用される。
- 必須の ***PartyRef*** 要素。***PartyRef*** 要素は、その*当事者*についての詳細情報へのポインタとなる。
- 1 つ以上の必須の ***CollaborationRole*** 要素。***CollaborationRole*** 要素は、その*当事者*が*プロセス仕様*のコンテキストにおいて果たすことができる役割を特定する。

- 1 つ以上の必須の ***Certificate*** 要素。***Certificate*** 要素は、セキュリティ機能において*当事者*が使用する証明書を特定する。
- 1 つ以上の必須の ***DeliveryChannel*** 要素。***DeliveryChannel*** 要素は、*当事者*が*メッセージ*を受信するために使用できる個々の配信経路の特性を定義する。その中には、転送レベル（HTTP など）とメッセージ取扱プロトコル（ebXML メッセージ取扱サービスなど）の両方が含まれる。
- 1 つ以上の必須の ***Transport*** 要素。***Transport*** 要素によって、*当事者*側で*メッセージ*の受信用にサポートしている転送プロトコルの特性が定義される。
- 1 つ以上の必須の ***DocExchange*** 要素。***DocExchange*** 要素によって、*当事者*側でサポートしている*メッセージ*交換の機能（*メッセージ*交換プロトコルなど）が定義される。

1.18.1 PartyId 要素

必須の ***PartyId*** 要素によって、*当事者*を論理的に識別するために使用してもよい論理識別子が提供される。その*当事者*を識別する別の論理識別子を指定するために、追加の ***PartyId*** 要素を、同じ ***PartyInfo*** 要素に付与してもよい。論理識別子に優先順位がある場合は、優先順位の高い順に ***PartyId*** 要素を並べる必要がある。

複数の ***PartyInfo*** 要素を含む *CPP* には、***PartyInfo*** 要素ごとに異なる論理識別子を定義するための ***PartyId*** 要素を付与してもよい。たとえば、これにより、大規模な組織の場合、目的ごとに異なる識別子を使用できるようになる。

PartyId 要素の値は、一意な識別子を与える任意の文字列である。この識別子は、*CPA* の両方の*当事者*が理解できるものならば何を使用してもよい。通常は、DUNS などの有名なディレクトリ、または[ISO6523]で指定されている名前付けシステムに列挙されている識別子を用いる。

PartyId要素には、***type***という単一の暗黙の属性が付与される。***type***属性の値は文字列である。

type属性が付与されている場合、***type***属性によって、***PartyId***要素の内容に適用される範囲または名前空間が指定される。

type 属性が付与されていない場合は、***PartyId*** 要素の内容は、[RFC2396]に準拠した URI でなければならない。***type*** 属性の値は、***PartyId*** 要素の値の名前空間を定義する URN であることが推奨される。通常、この URN は、有名な組織識別子のディレクトリとして登録される。

URI の参照方法の 2 つの例を次に示す。

```
<PartyId type = "uriReference">urn:duns:123456789</PartyId>
<PartyId type = "uriReference">urn:www.example.com</PartyId>
```

最初の例は、*当事者*の DUNS 番号の URN である。ただし、Dun and Bradstreet 社によって、Internet Assigned Numbers Authority（IANA）に DUNS 番号の URN が登録されていることが前提である。最後のフィールドは、この組織の DUNS 番号である。

2 番目の例には、任意の URN が示されている。この際、*当事者*自身が直接 IANA に登録した URN を用いてもよい。

1.18.2 PartyRef 要素

PartyRef 要素によって、この*当事者*に関する追加情報へのリンクが URI 形式で指定される。通常この URL は、情報の取得元となる URL である。追加情報は、*当事者*の Web サイトにある場合も、ebXML レジストリ、UDDI リポジトリ、LDAP ディレクトリなどの、公にアクセス可能なリポジトリにある場合もある。URI を介して入手可能な情報には、連絡先、所在地、電話番号、およびその*当事者*がサポートしている*取引コラボレーション*についての詳細が含まれていてもよい。この情報は、ebXML のコア構成要素[ccOVER]の形式で表現されていてもよい。これらの URI における情報の内容または形式について定義することは、本書の範囲外である。

PartyRef 要素は、[XLINK]のシンプルリンクである。次の属性が付与される。

- 必須の ***xlink:type*** 属性。
- 必須の ***xlink:href*** 属性。
- 暗黙の ***type*** 属性

1.18.2.1 xlink:type 属性

必須の ***xlink:type*** 属性は、固定値「simple」を持つこととする。この値によって、要素が[XLINK]のシンプルリンクであることが示される。

1.18.2.2 xlink:href 属性

必須の ***xlink:href*** 属性には、[RFC2396]に準拠した URI の値が付与されることとする。この値によって、その*当事者*についての外部情報が存在する場所が特定される。

1.18.2.3 Type 属性

暗示された ***type*** 属性の値により、*当事者*に関する外部情報の文書型が識別される。この値は、*当事者*に関する情報に関連付けられる名前空間を定義する URI でなければならない。***type*** 属性が省略されている場合、*当事者*に関する情報は HTML ウェブページで表示されなければならない。

PartyRef 要素の例を次に示す。

```
<PartyRef xlink:type="simple"
      xlink:href=http://example2.com/ourInfo.xml
```

```
            type="uri-reference"/>
```

1.18.3 CollaborationRole 要素

CollaborationRole 要素は、*当事者*を、*プロセス仕様*文書[ebBPSS]で定義されている*取引コラボレーション*内の特定の役割に関連付ける。一般に、*プロセス仕様*は、「買い手」、「売り手」などの役割によって定義される。*プロセス仕様*のコンテキスト内で実現される特定の*当事者*と役割の間の関係は、*CPP* 文書と *CPA* 文書の両方で定義される。*CPP* の場合、***CollaborationRole*** 要素は、*CPP* から参照されている各*プロセス仕様*文書において、その*当事者*が果たすことのできる役割を指定する。***CollaborationRole*** 要素の例は次の通りである。

```
<CollaborationRole id="N11" >
    <ProcessSpecification name="BuySell" version="1.0">
     ...
    </ProcessSpecification>
    <Role name="buyer" xlink:href="..."/>
    <CertificateRef certId = "N03"/>
      <!-- primary binding with "preferred" DeliveryChannel -->
    <ServiceBinding name="some process" channelId="N02"  packageId="N06">
        <!-- override "default" deliveryChannel  for selected message(s)-->
        <Override action="OrderAck" channelId="N05" packageId="N09"
            xlink:type="simple"
            xlink:href="..."/>

    </ServiceBinding>
    <!-- the first alternate binding -->
    <ServiceBinding channelId="N04" packageId="N06">
        <Override action="OrderAck" channelId="N05" packageId="N09"
      xlink:type="simple"
            xlink:href="..."/>
    </ServiceBinding>
</CollaborationRole>
```

*当事者*が、複数の*取引コラボレーション*で役割を果たしたり、特定の*取引コラボレーション*で複数の役割を果たしたりできることを示す場合、***PartyInfo*** 要素に複数の ***CollaborationRole*** 要素を付与することとする。個々の ***CollaborationRole*** 要素には、***ProcessSpecification*** 要素と ***Role*** 要素の適切な組み合せを付与することとする。

CollaborationRole 要素は、必須の ***ProcessSpecification*** 要素、必須の ***Role*** 要素、0 個以上の ***CertificateRef*** 要素、および 1 つ以上の ***ServiceBinding*** 要素、という 4 つの子要素から成ることとする。***ProcessSpecification*** 要素は、役割を定義するプロセス仕様文書を指定する。***Role*** 要素は、その当事者がサポートすることのできる役割を指定する。***CertificateRef*** 要素は、使用される証明書を指定する。***ServiceBinding*** 要素は、役割をデフォルトの ***DeliveryChannel*** にバインドする。デフォルトの ***DeliveryChannel*** は、指定されたプロセス仕様文書内で示されている役割のコンテキストにおいて当事者により受信されるすべてのメッセージトラフィックの受信プロパティを記述する。下記のように、代替の ***DeliveryChannel*** を、特定の目的のために、***Override*** 要素を用いて指定してもよい。

CollaborationRole の子要素として複数の ***ServiceBinding*** 要素が存在する場合は、***ServiceBinding*** 要素の順序は、*当事者*にとって優先順位が高いものから指定されることとする。特定の *プロセス仕様*文書に対するデフォルトの配信経路は、特定の *プロセス仕様*文書を参照する配信経路のうち、優先順位が最高の ***ServiceBinding*** 要素によって特定された配信経路である。

> 注意: *CPA* が合成される際には、当事者の間で互換性のある配信経路のうち、***ServiceBinding*** の優先順位を適用して優先順位が最高になる配信経路を選択する。

CPA が合成される際には、*当事者*の間で互換性のある ***ServiceBinding*** 要素のみが保持されることとする。それぞれの*当事者*は、*CPA* から参照されている個々の *プロセス仕様*文書に対して、デフォルトの配信経路を持つこととする。各 *プロセス仕様*文書において、個々の*当事者*に対するデフォルトの配信経路は、その *プロセス仕様*文書を参照している ***ServiceBinding*** 要素のうち、優先順位が最高の ***ServiceBinding*** 要素の ***channelId*** 属性によって示される配信経路である。

> 注意: 実装によっては、*取引コラボレーション*の実行中に*メッセージ*ごとに、ダイナミックに配信経路を割り当てる機能を提供してもよい。配信経路は、現在の条件に基づいて、*メッセージ*の送信元の*取引コラボレーション*を参照している ***ServiceBinding*** 要素で特定された配信経路から選択される。複数の配信経路が適用可能である場合は、優先順位が最高の ***ServiceBinding*** 要素から参照されている配信経路が使用される。

CollaborationRole 要素には、次に示す属性が付与される。

- 必須の ***id*** 属性。

1.18.3.1 id 属性

必須の ***id*** 属性は、[XML]の ID 属性である。この属性によって、***CollaborationRole*** 要素が、*CPP* 文書内の任意の場所から参照できるようになる。

1.18.3.2 CertificateRef 要素

空の ***CertificateRef*** 要素には、***certId*** という暗黙の IDREF 属性が付与される。***certId*** 属性によって、ID 属性の値に合致する（***PartyInfo*** 要素の下の）***Certificate*** 要素を参照するために使用される証明書が特定される。

1.18.3.3 certId 属性

暗黙の ***certId*** 属性は、[XML]の IDREF 属性である。[XML]の IDREF 属性は、***CollaborationRole*** 要素を、合致する ID 属性付きの証明書に関連付ける。

> 注意: 配信経路記述内で特定されている証明書は*メッセージ*交換に、***certId*** 属性

は*プロセス仕様*内の認可の役割に、それぞれ関連付けられる。

1.18.4 ProcessSpecification 要素

ProcessSpecification 要素によって、*当事者*間のやりとりを定義する*プロセス仕様*文書へのリンクが指定される。この*取引コラボレーション*記述を『ebXML ビジネスプロセス仕様スキーマ』[ebBPSS]に準拠して準備することが推奨されている。*プロセス仕様*文書は、ebXML レジストリに登録してもよい。

> 注意：*当事者*は、『ebXML ビジネスプロセス仕様スキーマ』とは別の方法で*取引コラボレーション*を記述することも可能である。*取引コラボレーション*の別の記述を使用する場合、*CPA* を結んだ*当事者*は、この*取引コラボレーション*記述の解釈方法、および*取引コラボレーション*記述内の情報を参照する *CPA* の要素の解釈方法に合意しなければならない。*CPA* の影響を受ける要素は、***Role*** 要素、***Override*** 要素、および ***Characteristics*** 要素のいくつかの属性である。

ProcessSpecification 要素の構文は次の通りである。

```
<ProcessSpecification
      name="BuySell"
      version="1.0"
      xlink:type="simple"
      xlink:href="http://www.ebxml.org/services/purchasing.xml"
      <ds:Reference ds:URI="http://www.ebxml.org/services/purchasing.xml">
            <ds:Transforms>
                  <ds:Transform
            ds:Algorithm="http://www.w3.org/TR/2000/CR-xml-c14n-20001026"/>
            </ds:Transforms>
            <ds:DigestMethod
                  ds:Algorithm="http://www.w3.org/2000/09/xmldsig#dsa-sha1">
                  String
            </ds:DigestMethod>
            <ds:DigestValue>j6lwx3rvEPO0vKtMup4NbeVu8nk=</ds:DigestValue>
      </ds:Reference>
</ProcessSpecification>
```

ProcessSpecification 要素には、***ds:Reference*** という必須の要素 1 つ、および次に示す属性が付与される。

- 必須の ***name*** 属性。タイプ ID が付属する。
- 必須の ***version*** 属性。
- 固定の ***xlink:type*** 属性。
- 必須の ***xlink:href*** 属性。

ds:Reference 要素は、***xlink:type*** 属性および ***xlink:href*** 属性と、次のように関連付けられる。個々の ***ProcessSpecification*** 要素は、***xlink:href*** 属性、および「simple」という値付きの ***xlink:type*** 属性が付与されることとする。また、『XML デジタル署名仕様』[XMLDSIG]

に従って形式化された ***ds:Reference*** 要素を付与してもよい。文書が署名付きである場合は、***ds:Reference*** 要素を使用しなければならない ***ds:Reference*** 要素が付与されている場合は、包含されている ***ProcessSpecification*** 要素の ***xlink:href*** 属性と同じ値を持つ、***ds:URI*** 属性が含まれていなければならない。

1.18.4.1 name 属性

ProcessSpecification 要素は、必須の ***name*** 属性として、*CPP* 文書内の他の場所でこの要素を参照する際に使用できる[XML]の ID を含んでいなければならない。

1.18.4.2 version 属性

ProcessSpecification 要素には、必須の ***version*** 属性が含まれている。***version*** 属性によって、***xlink:href*** 属性で特定された（または ***ds:Reference*** 要素でも特定された）*プロセス仕様*文書のバージョンが指定される。

1.18.4.3 xlink:type 属性

xlink:type 属性には、固定の値として「simple」が付与される。この値によって、要素が[XLINK]のシンプルリンクであることが示される。

1.18.4.4 xlink:href 属性

必須の ***xlink:href*** 属性には、*プロセス仕様*文書、および[RFC2396]に準拠した URI の値が付与されることとする。

1.18.4.5 ds:Reference 要素

ds:Reference 要素によって、包含されている ***ProcessSpecification*** 要素の ***xlink:href*** 属性と同じプロセス仕様文書が特定される。さらに、CPP が作成されてから*プロセス仕様*文書が変更されていないことも検証される。

> 注意: *当事者*は、任意の時点で、*CPP* または *CPA* の有効性をテストしてもよい特に、次の有効性テストが有用である場合がある。
>
> - 作成後に変更されている場合、*CPA* の合成開始時に、*CPP* および参照されている*プロセス仕様*文書の有効性をテストする。
> - *CPA* の*当事者*システムへのインストール時に、*CPA* および参照されている*プロセス仕様*文書の有効性をテストする。
> - *CPA* が*当事者*システムにインストールされた後に間隔を置いて、*CPA* の有効性をテストする。*CPA* および*プロセス仕様*文書は、インストールツールによって特定のミドルウェアに適合する形式に変換してもよい。したがって、*CPA* および参照されている*プロセス仕様*文書の変更は、実行中の実行時処理には必ずしも影響を及ぼさない。そのような変更は、*CPA* および参照されている*プロセス仕様*文書の再インストールが必要になるまで検出されない可能性がある。

ds:Reference 要素およびその子要素の構文および意味情報は、『XML デジタル署名仕様』[XMLDSIG]で定義されている。上記の例に示されている ***ds:DigestMethod*** 属性の文字列の値の替わりに、文字列値を持つ子要素 ***ds:HMACOutputLength*** を使用してもよい。

[XMLDSIG]に従って、***ds:Reference*** 要素は、***ds:Transforms*** という子要素を 1 つ持つ。さらに、***ds:Transforms*** 要素は、***ds:Transform*** という 1 つ以上の子要素の順序付きリストを持つ。ただし、本書では現在、『Canonical XML』[XMLC14N]に示されている変換の使用を要求している。他の変換は許されていない。したがって、次の追加の要件が、***ProcessSpecification*** 要素内の ds:Reference 要素に適用される。

- ***ds:Reference*** 要素は、***ds:Transforms*** という子要素を持たなければならない。
- ***ds:Transforms*** 要素は、***ds:Transform*** という子要素を 1 つだけ持たなければならない。
- ***ds:Transform*** 要素には、必須の ***ds:Algorithm*** 属性の必須の値（http://www.w3.org/TR/2000/CR-xml-c14n-20001026）を介して、『Canonical XML』[XMLC14N]に示されている変換を指定しなければならない。

Canonical XML の実装は、『XML デジタル署名仕様』[XMLDSIG]によって要求されているという点に注意すること。

ProcessSpecification 要素の下の ***ds:Reference*** 要素は、*CPP* の有効性を示唆している。

- ***ProcessSpecification***要素の下の***ds:Reference***要素が『XMLデジタル署名仕様』[XMLDSIG]で定義されている参照の検証に失敗した場合、*CPP*は無効であると見なされなければならない。

- ***ds:Reference*** 要素が参照できない場合は、*CPP* は無効であると見なされなければならない。

ds:Reference 要素の有効性は、***ds:Signature*** 要素の記述によっても規定される。

> 注意: 『XMLデジタル署名仕様』[XMLDSIG]には、「署名アプリケーションは、Reference要素内の署名元によって提供される識別記号（URI）および変換に依存してもよい。あるいは、ローカルキャッシュなどの方法を通して内容を取得してもよい（MAYが強調されている）と書かれている。ただし、ebXML *CPP/CPA*の実装においては、署名または有効化の際にキャッシュの結果を使用しないことが推奨される。
>
> 注意: 『XML デジタル署名仕様』[XMLDSIG]には、XML 文書および外部参照されている文書への署名について記述されている。*CPP* または *CPA* 文書が正しく署名されている場合、その機能は参照されている*プロセス仕様*文書が変更されていないことを確証するためにも使用されることになる。ただし、本書では現在のところ、*CPP* または *CPA* は署名されている必要があるとはしていない。

注意: *CPA* の *当事者*が、既存の *プロセス仕様*文書をカスタマイズしたいと考える場合は、既存の文書をコピーし、修正し、*CPA* で修正したコピーを参照してもよい。明確さ、簡潔さ、および履歴の記録のために、既存の *プロセス仕様*文書を元の形式で参照し、その参照を仕様の合意済みの修正部分に付けることを望むかもしれない。したがって、*CPP* における ***ProcessSpecification*** 要素内の ***ds:Reference*** 要素の ***ds:Transforms*** というサブ要素の使用方法は、『XML デジタル署名仕様』[XMLDSIG]で指定されている他の変換を許す方向に、将来拡大される可能性がある。たとえば、元の文書への修正は、将来、XSLT 変換として表現されるかもしれない。変換の適用後、変換された文書を『ebXML ビジネスプロセス仕様スキーマ』[ebBPSS]と比較して検証する必要がある。

1.18.5 Role 要素

必須の ***Role*** 要素は、*プロセス仕様*内でその *当事者*が、***CollaborationRole*** 要素内の ***ServiceBinding*** 要素を介して、どの役割をサポートできるかを指定する。

Role 要素には、次に示す属性が付与される。

- 必須の ***name*** 属性。
- 固定の ***xlink:type*** 属性。
- 必須の ***xlink:href*** 属性。

1.18.5.1 name 属性

必須の ***name*** 属性は、***役割***の名前を表す文字列である。この属性の値は、***ProcessSpecification*** 要素が参照している *プロセス仕様*[ebBPSS]内の場所から取得される。参照されているプロセスのどの要素がルート（最上部）になるかによって、次のうちのいずれかの場所が取得元となる。

- ***BinaryCollaboration***/***InitiatingRole*** 要素の ***name*** 属性
- ***BinaryCollaboration***/***RespondingRole*** 要素の ***name*** 属性
- ***BusinessTransactionActivity*** 要素の ***fromAuthorizedRole*** 属性
- ***BusinessTransactionActivity*** 要素の ***toAuthorizedRole*** 属性
- ***CollaborationActivity*** 要素の ***fromAuthorizedRole*** 属性
- ***CollaborationActivity*** 要素の ***toAuthorizedRole*** 属性
- ***business-partner-role*** 要素の ***name*** 属性

1.18.5.2 xlink:type 属性

xlink:type 属性には、固定の値として「simple」が付与される。この値によって、要素が[XLINK]のシンプルリンクであることが示される。

1.18.5.3 xlink:href 属性

必須の ***xlink:href*** 属性には、[RFC2396]に準拠した URI の値が付与されることとする。この値によって、*取引*/*コラボレーション*のコンテキストにおける役割を定義する、*プロ*

*セス仕様*文書内の要素または属性の場所が指定される。以下に例を示す。

```
Xlink:href="http://www.ebxml.org/processes/purchasing#N05
```

"N05"は、役割名を定義するプロセス仕様文書内の要素の ID 属性値である。

1.18.6 ServiceBinding 要素

ServiceBinding 要素によって、指定されたプロセス仕様文書のコンテキストにおいて当事者に送信されるすべてのメッセージトラフィックのための DeliveryChannel 要素が指定される。ServiceBinding 要素の例を次に示す。

```
<ServiceBinding name="SomeProcess" channelId="X03" packageId="N06">
            <Override action="OrderAck"
                  channelId="X04"
                  packageId="N09"
                  xlink:type="simple"
                  xlink"href="..."/>  <!--zero or more-->
      </ServiceBinding>
```

ServiceBinding 要素には、1 つの子 ***Service*** 要素、および 0 またはそれ以上の ***Override*** 子要素が含まれる。

ServiceBinding 要素に次に示す属性を付与する。

- 必須の ***channelId*** 属性。
- 必須の ***packageId*** 属性。

1.18.6.1 channelId 属性

必須の ***channelId*** 属性は、***ProcessSpecification*** 要素が参照している*プロセス仕様*のために受信されるすべての*メッセージ*トラフィックのためのデフォルトのテクニカルバインディングを提供する ***DeliveryChannel*** を指定する[XML]の IDREF であるとする。

1.18.6.2 packageId 属性

必須の ***packageId*** 属性は、***ServiceBinding*** 要素と共に使用される ***Packaging*** 要素を指定する[XML]の IDREF である。

1.18.7 Service 要素

Service 要素の値は、ebXML *メッセージヘッダ*[ebMS]内の ***Service*** 要素の値、または別のメッセージ取扱サービスの*メッセージヘッダ*内の類似した要素の値として使用されることになる文字列である。

*プロセス仕様*文書が『ebXML ビジネスプロセス仕様スキーマ』[ebBPSS]により定義される場合、***Service*** 要素の値は、***CollaborationRole*** 親要素内で識別される役割に対応す

る、認定された役割と関連付けられる一連の*取引トランザクション*全体に対する識別子となる。

> 注意：***Service*** 要素の目的は、ebXML *メッセージヘッダ*の経路情報を提供することのみである。***CollaborationRole*** 要素およびその子要素は、*CPP* または *CPA* に関連する ***ProcessSpecification*** 文書内の情報を識別する。

1.18.7.1 type 属性

type 属性が存在する場合、メッセージの送受信をする*当事者ら*が他の方法によって、***Servic*** 要素の値の解釈方法を知っていることを示す。これらの 2 つの*当事者*は、この ***type*** 属性値を使用して解釈を行うことができる。

type 属性が存在しない場合、***Servic*** 要素の値は URI[RFC2396]でなければならない。

1.18.8 Override 要素

Override 要素を使用することで、*当事者*は、選択された*取引トランザクション*のメッセージへの別の ***DeliveryChannel*** をマッピング、またはバインドすることができる。この場合の*取引トランザクション*は、親の ***ServiceBinding*** 要素のコンテキストにおいて*当事者*が受信することになっている。

それぞれの ***Override*** 要素によって、親の ***ServiceBinding*** 要素に関連付けられている*プロセス仕様*のコンテキストにおいて*当事者*が受信することになる*メッセージ*ごとに、異なる ***DeliveryChannel*** 要素が指定される。***Override*** 要素には、次に示す属性が付与される。

- 必須の ***action*** 属性。
- 必須の ***channelId*** 属性。
- 必須の ***packageId*** 属性。
- 暗黙の ***xlink:href*** 属性。
- 固定の ***xlink:type*** 属性。

ServiceBinding 要素において、特定の ***action*** 属性の値を持つ ***Override*** 要素は 1 つだけであるとする。

> 注意: 2 つの *CPP* から *CPA* を合成する際に、一方の *CPP* の配信経路が他方の*当事者*と互換性のない ***Override*** 要素を持つ場合もある。この非互換性は、交渉するか、または互換性のあるデフォルトの配信経路に復元することにより、解決しなければならない。

1.18.8.1 action 属性

必須の ***action*** 属性の値は、***channelId*** 属性によって識別される ***DeliveryChannel*** と関連付けられることになっている*取引トランザクション*を識別する文字列である。*プロセス仕*

*様*文書が『ebXML ビジネスプロセス仕様スキーマ』[ebBPSS]で定義されている場合は、***action*** 属性の値は、***ProcessSpecification*** 要素によって参照される *プロセス仕様*文書内で必要とされる ***BusinessTransaction*** 要素の ***name*** 属性の値と一致しなければならない。

取引コラボレーションの別の記述に関しては、該当する節の注意事項を参照のこと。

1.18.8.2 channelId 属性
必須の ***channelId*** 属性は、***action*** 属性で特定される*メッセージ*に関連付けられる ***DeliveryChannel*** 要素を識別するための[XML]の IDREF である。

1.18.8.3 packageId 属性
必須の ***packageId*** 属性は、***action*** 属性で特定される*メッセージ*に関連付けられた ***Packaging*** 要素を識別するための[XML]の IDREF である。

7.3.7.4 xlink:href 属性
暗黙の ***xlink:href*** 属性が付与されていてもよい。この属性が付与されている場合は、「絶対」[XPOINTER] URI 表現を使用する。この絶対 URI によって、***ProcessSpecification*** 要素で識別される *プロセス仕様*文書[ebBPSS]内の ***BusinessTransaction*** 要素を指定する。

7.3.7.5 xlink:type 属性
暗黙の ***xlink:type*** 属性には、固定の値として「simple」が付与される。この値によって、要素が[XLINK]のシンプルリンクであることが示される。

1.18.9 Certificate 要素

Certificate 要素によって、この *CPP* 内で使用される証明書が定義される。*CPP* 内のさまざまなセキュリティ機能で使用するために、1 つ以上の ***Certificate*** 要素を指定してもよい。***Certificate*** 要素の例を次に示す。

```
<Certificate certId = "N03">
      <ds:KeyInfo>. . .</ds:KeyInfo>
</Certificate>
```

Certificate 要素の属性としては、***certId*** という、必須の属性だけが付与される。
Certificate 要素の子要素としては、***ds:KeyInfo*** という要素だけが付与される。

1.18.9.1 certId 属性
必須の ***certId*** 属性は、ID 属性である。この属性は、IDREF 属性を用いて、***CertificateRef*** 要素内で参照される。一方、証明書は、*CPP* 内の他の場所で指定される。その例を以下に示す。

```
<CertificateRef certId = "N03"/>
```

1.18.9.2 ds:KeyInfo 要素

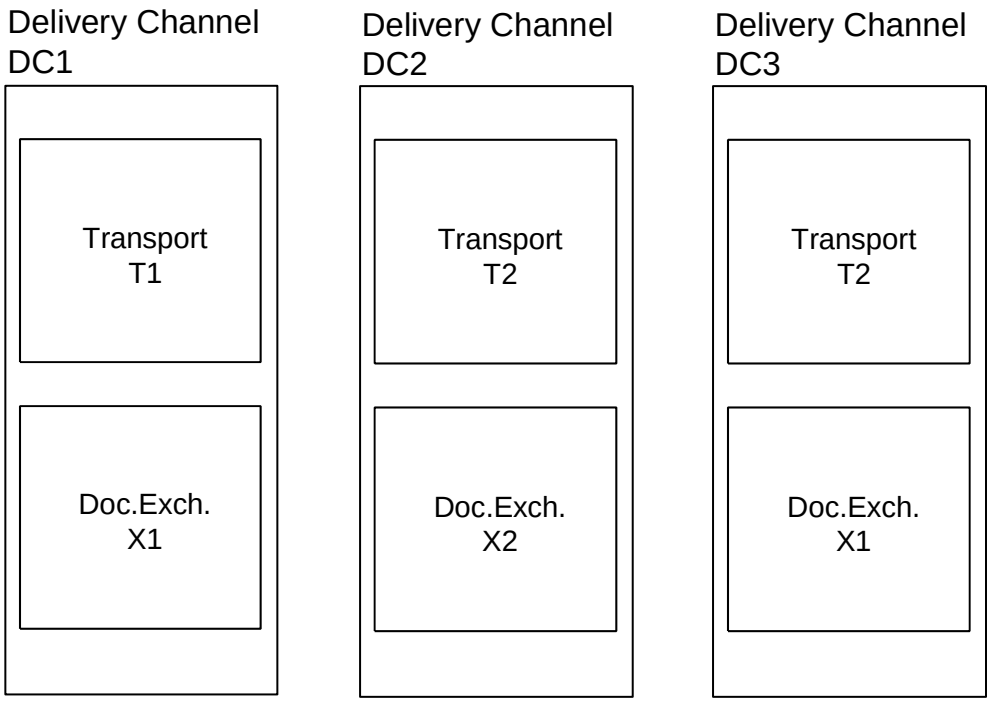
ds:KeyInfo 要素によって、証明書情報が定義される。***ds:KeyInfo*** 要素および ***ds:KeyInfo*** 要素のサブ要素の内容は、『XML デジタル署名仕様』[XMLDSIG]で定義されている。

> 注意: *CPP* および *CPA* を作成するためのソフトウェアは、***ds:KeyInfo*** 要素を認識して、証明書の定義に必要なサブ要素構造を挿入してもよい。

1.18.10 DeliveryChannel 要素

配信経路は、***Transport*** 要素と ***DocExchange*** 要素を組み合せて、*当事者のメッセージ受信の特性を記述したものである。CPP* は、1 つ以上の ***DeliveryChannel*** 要素、1 つ以上の ***Transport*** 要素、および 1 つ以上の ***DocExchange*** 要素を含むこととする。個々の配信経路は、***DocExchange*** 要素と ***Transport*** 要素の任意の組み合せを参照してもよい。また、同じ ***DocExchange*** 要素または同じ ***Transport*** 要素が、複数の配信経路で参照されていてもよい。配信経路は、異なる通信のアドレスまたはセキュリティ定義を指定してさえいれば、同じ転送プロトコルおよび同じ文書交換プロトコルを使用していてもよい。図 5

図5: 3つの配信経路



に配信経路の 3 つの例を示す。

配信経路には、「DC1」、「DC2」、「DC3」という値を持つ ID 属性が付与されている。個々の配信経路には、1 つの転送定義と 1 つの文書交換定義が含まれている。転送定義および文書交換定義には、図に示されているような名前が付けられている。配信経

路 DC3 では、他の配信経路で使用されているのと同じ転送定義および文書交換定義が異なる組み合せで参照されている。図の中では、配信経路 DC3 は、転送定義 T2（配信経路 DC2 からも参照されている）および文書交換定義 X1（配信経路 DC1 からも参照されている）の組み合せになっている。

特定の配信経路は、個々の ***ServiceBinding*** 要素または ***Override*** 要素（***action*** 属性）に関連付けられていることとする。配信経路の構文は次の通りである。

```
<DeliveryChannel channelId="N04" transportId="N05" docExchangeId="N06">
      <Characteristics
            syncReplyMode = "responseOnly"
            nonrepudiationOfOrigin = "true"
            nonrepudiationOfReceipt = "true"
            secureTransport = "true"
            confidentiality = "true"
            authenticated = "true"
            authorized = "true"/>
</DeliveryChannel>
```

個々の ***DeliveryChannel*** 要素には、1 つの ***Transport*** 要素および 1 つの ***DocExchange*** 要素が指定されている。これらを合わせて、1 つの配信経路定義となっている。

DeliveryChannel 要素には、次に示す属性が付与される。

- 必須の ***channelId*** 属性。
- 必須の ***transportId*** 属性。
- 必須の ***docExchangeId*** 属性。

DeliveryChannel 要素には、***Characteristics*** という、1 つの必須の子要素が付与される。

1.18.10.1 channelId 属性

channelId 属性は、*CPP* または *CPA* の他の部分から IDREF 属性を用いて ***DeliveryChannel*** 要素を一意に識別するための[XML]の ID 属性である。

1.18.10.2 transportId 属性

transportId 属性は、配信経路の転送特性を定義するための ***Transport*** 要素を識別する[XML]の IDREF である。*CPP* 文書の他の場所で指定されている ***Transport*** 要素の ***transportId*** 属性の値と同じ値を持たなければならない。

1.18.10.3 docExchangeId 属性

docExchangeId 属性は、配信経路の文書交換特性を定義する ***DocExchange*** 要素を特定するための[XML]の IDREF である。*CPP* 文書の他の場所で指定されている ***DocExchange*** 要素の ***docExchangeId*** 属性の値と同じ値を持たなければならない。

1.18.11 Characteristics 要素

Characteristics 要素によって、配信経路のセキュリティ特性などの属性が記述される。***Characteristics*** 要素の属性（***syncReplyMode*** 以外）を用いて、*プロセス仕様*文書内の対応する属性の値を上書きしてもよい。

取引コラボレーションの別の記述に関しては、該当する節の注意事項を参照のこと。

Characteristics 要素には、次に示す属性が付与される。

- 暗黙の ***syncReplyMode*** 属性。
- 暗黙の ***nonrepudiationOfOrigin*** 属性。
- 暗黙の ***nonrepudiationOfReceipt*** 属性。
- 暗黙の ***secureTransport*** 属性。
- 暗黙の ***confidentiality*** 属性。
- 暗黙の ***authenticated*** 属性。
- 暗黙の ***authorized*** 属性。

1.18.11.1 syncReplyMode 属性

syncReplyMode 属性の値は、次の値のいずれかを列挙したものである。

- "signalsOnly"
- "responseOnly"
- "signalsAndResponse"
- "none"

この属性（存在する場合）は、HTTP などの同期通信プロトコルへの応答として、受信側のアプリケーションが予期する情報の種類を示している。「signalsOnly」という値は、返信された応答（HTTP の場合は HTTP 200）が、*プロセス仕様*文書 [ebBPSS]で定義されている 1 つ以上の*ビジネスシグナル*（*ビジネス*応答*メッセージ*ではない）のみを含むことを示している。「responseOnly」という値は、*ビジネス*応答*メッセージ*のみが返信されることを示している。「signalsAndResponse」という値は、1 つ以上の*ビジネスシグナル*だけでなく、*ビジネス*応答*メッセージ*もアプリケーションによって返信されることを示している。***syncReplyMode*** 属性の暗黙のデフォルト値である「none」という値は、*ビジネス*応答*メッセージ*も*ビジネスシグナル*も同期して返信されないことを示している。この場合、*ビジネス*応答*メッセージ*および*ビジネスシグナル*は、非同期応答として返信される。

syncReplyMode 属性の値が「none」以外である場合、ebXML メッセージ取扱サービスの ***syncReply*** 属性は「true」という値に設定される。

配信経路が同期機能のない転送プロトコル（SMTP など）を指定しており、***Characteristics*** 要素の ***syncReplyMode*** 属性の値が「none」以外である場合、応答には、転送プロトコルが同期応答をサポートしている場合と同じ内容が含まれる。

1.18.11.2 nonrepudiationOfOrigin 属性

nonrepudiationOfOrigin 属性の値は、「true」または「false」（論理値）である。値が「true」である場合、配信経路は、*メッセージ*を送信した*当事者*の証明書によってデジタル署名された*メッセージ*を要求する。

1.18.11.3 nonrepudiationOfReceipt 属性

nonrepudiationOfReceipt 属性の値は、「true」または「false」（論理値）である。値が「true」である場合、配信経路は、*メッセージ*を受信した*当事者*の証明書によってデジタル署名された*メッセージ*により承認された*メッセージ*を要求する。

1.18.11.4 secureTransport 属性

secureTransport 属性の値は、「true」または「false」（論理値）である。値が「true」である場合、配信経路が、[SSL]、[IPSEC]などの安全な転送プロトコルを使用していることを示している。

1.18.11.5 confidentiality 属性

confidentiality 属性の値は、「true」または「false」（論理値）である。値が「true」である場合、配信経路が、持続的な方法で暗号化された*メッセージ*を要求していることを示している。*メッセージ*は、転送レベルより上のレベルで暗号化され、アプリケーションに暗号化された状態で配信されなければならない。

1.18.11.6 authenticated 属性

authenticated 属性の値は、「true」または「false」（論理値）である。値が「true」である場合、アプリケーションに配信される前に*メッセージ*の送信元が認証されることを、配信経路が要求していることを示している。

1.18.11.7 authorized 属性

authorized 属性の値は、「true」または「false」（論理値）である。値が「true」である場合、アプリケーションに配信される前に*メッセージ*の送信元が認可されることを、配信経路が要求していることを示している。

1.18.12 Transport 要素

CPP の ***Transport*** 要素によって、通信プロトコル、エンコード方式、および転送セキュリティに関する*当事者*の能力が定義される。

Transport 要素の全体構造は、次の通りである。

```
<Transport transportId = "N05">
      <!--protocols are HTTP, SMTP, and FTP-->
      <SendingProtocol version = "1.1">HTTP</SendingProtocol>
            <!--one or more SendingProtocol elements-->
      <ReceivingProtocol version = "1.1">HTTP</ReceivingProtocol>
      <!--one or more endpoints-->
```

```
        <Endpoint uri="http://example.com/servlet/ebxmlhandler"
              type = "request"/>
        <TransportSecurity>  <!--0 or 1 times-->
              <Protocol version = "3.0">SSL</Protocol>
              <CertificateRef certId = "N03"/>
        </TransportSecurity>
  </Transport>
```

1.18.12.1 transportId 属性

Transport 要素には、***Transport*** 要素ごとの一意な識別子となる、1 つの必須の ***transportId*** 属性（[XML]の ID 型）が含まれている。この値は、*CPP* または *CPA* 文書内の他の場所で、***DeliveryChannel*** 要素の ***transportId*** IDREF 属性によって参照されていることとする。

1.18.12.2 同期応答

転送プロトコルの大きな特徴の 1 つとなるのは、該当の転送プロトコルが同期応答をサポートしているかどうかという点である。同期応答に関する詳細は、7.5.11.1 を参照のこと。

1.18.13 転送プロトコル

サポートされている通信プロトコルは、HTTP、SMTP、および FTP である。CPP では、*当事者*がサポートしているプロトコルをすべて指定してもよい。

> 注意: 本書の目標は、ここで定義されている語彙を用いて MIME コンテンツを搬送する転送プロトコルをサポートできるようにすることである。

1.18.13.1 SendingProtocol 要素

SendingProtocol 要素によって、*当事者*が、当事者たちに対して取引データを送付するために使用できる（または選択する）プロトコルが指定される。暗黙の ***version*** 属性によって、そのプロトコルの特定のバージョンが指定される。たとえば、*CPP* 内で、SMTP または HTTP を値とする ***SendingProtocol*** 要素を含む ***Transport*** 要素が、***DeliveryChannel*** 要素内で参照されており、その ***DeliveryChannel*** 要素が購買注文プロセスにおける売り手の役割として参照されているとする。その場合、その当事者は、SMTP と HTTP のいずれかによって購買注文を送付できると宣言していることになる。*CPP* の場合は、***SendingProtocol*** 要素が、***Transport*** 要素ごとに 1 回以上現れてもよい。*CPA* の場合は、***SendingProtocol*** 要素は、1 回だけ現れることとする。

1.18.13.2 ReceivingProtocol 要素

ReceivingProtocol 要素によって、*当事者*が相手*当事者*から*取引データ*を受信するために用いるプロトコルが指定される。暗黙の ***version*** 属性によって、そのプロトコルの特定のバージョンが指定される。たとえば、*CPP* 内で、***Transport*** 要素が、HTTP を値とし

て持つ ***ReceivingProtocol*** 要素を含む ***DeliveryChannel*** 要素内で参照されており、その ***DeliveryChannel*** 要素が購買注文*取引コラボレーション*における売り手の役割として参照されているとする。その場合、その当事者は、HTTP によって購買注文に対するビジネス応答を受信できると宣言していることになる。

CPA においては、***SendingProtocol*** 要素および ***ReceivingProtocol*** 要素は、当事者たちの補完的な役割にどの転送プロトコルを使用するかに関して、合意された実際の合意書を示すために使用する。たとえば、上記の例の続きで考えると、この購買注文*取引コラボレーション*の売り手は、受信プロトコルとして SMTP、送信プロトコルとして HTTP を、それぞれ指定できる。これらの当事者たちの能力は、*CPP* の中で示されている買い手の能力と合致している。この能力の合致によって、買い手から購買注文が SMTP で送付され、売り手から買い手に対して購買注文への応答（承認、取消、依頼変更など）が HTTP で送信されるという相互運用可能な転送合意書がサポートされる。

受信転送能力を完全に記述するには、受信プロトコル情報と、エンドポイントを提供する URL を組み合わせる必要がある（下記を参照）。

> 注意: この URL スキームによって、使用されるプロトコルに関する情報が与えられる。しかし、***ReceivingProtocol*** 要素によって明示的に指定する方法も、依然として、すべてのエンドポイントを同じ URL スキーム（たとえば、HTTP エンドポイントはあらゆる転送プロトコルで使用される）で識別するのに役に立つ。同様に、HTTPS は、転送セキュリティプロトコルとして[SSL]を使用している HTTP である。このため、HTTP://という URL スキームも、HTTPS://という URL スキームも、同じ受信プロトコルと見なすことができる。したがって、***ReceivingProtocol*** 要素は、接続の際に必要な情報を示すために用意されているエンドポイント非依存である。

1.18.14 Endpoint 要素

Endpoint 要素の必須の ***uri*** 属性によって、ReceiveProtocol 要素に関連付けられた、*当事者*の通信アドレス情報が指定される。1 つ以上の ***Endpoint*** 要素が、さまざまな目的に対して異なるアドレスを与えるために、***Transport*** 要素ごとに付与されることとする。***uri*** 属性の値は、選択されたプロトコルで要求される形式で表された、*当事者*の電子アドレスを含む URI である。***uri*** 属性の値は、[RFC2396]で定義されている URI の構文に準拠することとする。

type 属性によって、このエンドポイントの目的が指定される。***type*** 属性の値は、「login」、「request」、「response」、「error」、および「allPurpose」のいずれかが列挙されたものである。これらの値すべてを 1 つずつ並べたものが最大の値となる。***type*** 属性は省略してもよい。***type*** 属性を省略した場合は、デフォルト値の「allPurpose」が値となる。「login」エンドポイントは、*当事者*間の初期*メッセージ*のアドレスとして使用してもよい。「request」エンドポイントおよび「response」エンドポイントは、そ

れぞれ、*メッセージ*の依頼および応答のために使用される。「error」エンドポイントは、メッセージ取扱サービスによって発行されるエラー*メッセージ*のアドレスとして使用してもよい。「error」エンドポイントが定義されている場合は、これらのエラー*メッセージ*は、「response」アドレス（定義されている場合）、または「allPurpose」エンドポイントに送信されることとする。エラーメッセージを受信できるようにするには、個々の ***Transport*** 要素が、「error」、「response」、または「allPurpose」のうちの、少なくとも 1 つのエンドポイントを含むこととする。

1.18.15 転送プロトコル

以降の節では、サポートされている転送プロトコルについて個別に議論する。

1.18.15.1 HTTP

HTTP は、ハイパーテキスト転送プロトコル[HTTP]の略である。HTTP で使用されるアドレスは、[RFC2396]に準拠した URI であることとする。アプリケーションによっては、1 つ以上のエンドポイントが存在してもよい。どのエンドポイントが使用されるかは、アプリケーションによって決定される。

HTTP エンドポイントの例を次に示す。

```
<Endpoint uri="http://example.com/servlet/ebxmlhandler"
      type = "request"/>
```

「request」エンドポイントおよび「response」エンドポイントは、特定の依頼または非同期応答の際に、*CPA* の*取引文書*において交換されるアプリケーション指定の URI によって、動的に上書きされる。

同期応答の場合、「response」エンドポイントは、存在していても無視される。同期応答は常に、既存の接続を介して（すなわち、接続元として識別される URI に向けて）返信される。

1.18.15.2 SMTP

SMTP は、Simple Mail Transfer Protocol [SMTP]の略である。この規格を使用する場合は、多目的インターネットメッセージ拡張 [MIME]がサポートされていなければならない。SMTP トランスポート層で使用される MIME メディアタイプは、サブタイプ「octet-stream」の「Application」である。

SMTP の場合、通信アドレスは、[RFC822]で定義されている送信先の完全なメールアドレスである。SMTP エンドポイントの例を次に示す。

```
<Endpoint uri="mailto:ebxmlhandler@example.com"
      type = "request"/>
```

MIME 付きの SMTP は、必要に応じて、パス内のリンクごとに文書のエンコードおよび

デコード、および送信先の文書交換機能へのデコードされた文書の送信を自動的に行う。

> 注意: 上位レベル（メールユーザエージェント）で既にエンコードされていない場合は、SMTP メール転送エージェントによって、バイナリデータ（すなわち、7 ビット ASCII ではないデータ）にエンコードされる。
>
> 注意: SMTP 自身は（認証または暗号化を含まないため）、サービスの拒否を受けたり、未知の*当事者*によって隠蔽されたりする場合がある。トランスポート層に SMTP を選択した場合は、文書交換層またはトランスポート層で適切な暗号化および認証方法（[S/MIME]など）も選択することを強く推奨する。
>
> 注意: SMTP は、特定のサービス品質を保証しない非同期プロトコルである。メール*メッセージ*の到着に対するトランスポート層の承認（すなわち「SMTP 承認」）に続いて、メール*メッセージ*の配信方法を知っている、将来のある時点に*メッセージ*の配信を試みる、という点について、SMTP サーバ側で宣言が発信される。ただし、この*メッセージ*は確固としたものではなく、送付先に配信されない場合もある。さらに、送付元は、トランスポート層の承認を最も近いノードからのみ受信する。この*メッセージ*が中間ノードを介して送信された場合も、SMTP はエンドツーエンドの承認を提供しない。したがって、SMTP 承認を受け取っても*メッセージ*がアプリケーションに配信されたことの保証にはならないし、SMTP 承認を受け取らなかったとしても*メッセージ*が配信されなかったことの証拠とはならない。SMTP と共に、信頼性の高い ebXML *メッセージ取扱サービス*のメッセージ取扱プロトコルを使用することが推奨される。

1.18.15.3 FTP

FTP は、ファイル転送プロトコル[RFC959]の略である。

配信経路によって受信特性が指定されているので、それぞれの*当事者*は、FTP の PUT を使用して*メッセージ*を送信する。エンドポイントでは、（*当事者*に PUT するための）ユーザ ID および入力ディレクトリパスを指定する。FTP エンドポイントの例を次に示す。

```
<Endpoint uri="ftp://userid@server.foo.com"
      type = "request"/>
```

FTP はすべての実装で横断的に互換性がなければならないので、ebXML 用の FTP では、[RFC959]の 5.1 節で指定され[RFC1123]の 4.1.2.13 節で修正されているコマンドおよびパラメータのうちの、最小セットを使用する。このモードは、ストリーム専用であり、タイプは ASCII Non-print（AN）、Image（I）、または Local 8（L8）のいずれかでなければならない。

ストリームモードの場合、ファイルの最後でデータ接続が閉じられる。複数のサードパ

ーティセッションが存在している場合、各転送コマンドがそれぞれのポートを取得する前に、サーバ側の FTP を「PASV」に設定しなければならない。

NOTE: [RFC 959]には、TCP 接続が閉じられてからソケットペアが再利用できるまでに時間がかかるのを避けるために、転送コマンドを送信する前に必ず、ユーザ側の FTP から PORT コマンドを送信してデフォルトでないデータポートを割り当てることによって、1 つの FTP で複数の転送を可能にする必要がある。

注意: PASV コマンドに対する 227 応答の形式は十分標準化されていないので、FTP クライアントは、[RFC959]で示されているように、念のために、括弧を付けておくことにするかもしれない。ユーザ側の FTP プログラムがホストおよびポート番号の最初の桁に対する応答がスキャンされない場合、ユーザ側の FTP で間違ったホストを示すことになる。この応答において、h1、h2、h3、および h4 はサーバホストの IP アドレス、p1 および p2 は PASV が割り当てられているデフォルトでないのデータ転送ポートである。

注意: ファイアウォール透明性のためのプロポーザルとして、[RFC1579]では、クライアント側が PASV コマンドを送信することを提案している。これにより、サーバ側では、ランダムポートで受動的に TCP を開いた後、クライアントにポート番号を通知できるようになる。この結果、クライアントは、接続をアクティブに開いて確立できる。

注意: STREAM モードの場合、ファイルの最後でデータ接続が閉じられるので、制御コード 226 または 250 が送信側から届かない場合、受信側の FTP で異常切断と見なされる可能性がある。

注意: [RFC1579]では、FTP プロトコルで、開始時にクライアントに APSV（すべて受動的）という新しいコマンドを送信させることにより、オプションを実装しているサーバが常に受動的に開かれるようにすることが有効であるかもしれないという観察を示している。151 という新しい応答コードは、PORT コマンドまたは PASV コマンドの前に送信されたすべてのファイル転送要求への応答として発行される。このメッセージには、転送に用いられるポート番号が含まれる。PORT コマンドが、以前に APSV を受信したサーバに送信される場合もある。このコマンドによって、次の転送処理に関わるデフォルトの動作が上書きされるため、サードパーティによる転送が可能になる。

1.18.16 転送セキュリティ

TransportSecurity 要素によって、***ReceivingProtocol*** 要素に関連付けられた、*CPP* のトランスポート層のための当事者のセキュリティ仕様が提供される。この *CPP* から合成された *CPA* に対して、転送セキュリティが使用されない場合、この要素は省略してもよい。以降で特に指定がなければ、転送セキュリティは双方向の*メッセージ*に適用される。

構文は次の通りである。

```
<TransportSecurity>
      <Protocol version = "3.0">SSL</Protocol>
      <CertificateRef certId = "N03"/> <!--zero or one-->
</TransportSecurity>
```

TransportSecurity 要素には、***Protocol*** および ***CertificateRef*** という、2 つの必須の子要素が含まれる。

1.18.16.1 Protocol 要素

Protocol 要素の値によって、*当事者*がサポートを準備している転送セキュリティプロトコルが識別できる。暗黙の ***version*** 属性によって、指定されたプロトコルのバージョンが識別される。

特定のセキュリティプロパティは、指定されたプロトコルによって提供されるサービスに依存する。たとえば、SSL は、証明書ベースの暗号化および認証を実行する。

認証が双方向であるか*メッセージ*送付元から*メッセージ*送付先に向けてであるかは、選択された転送セキュリティプロトコルによって決まる。

1.18.16.2 CertificateRef 要素

空の ***CertificateRef*** 要素には、***certId*** という暗黙の IDREF 属性が付与される。certId 属性によって、ID 属性の値に合致する（PartyInfo 要素の下の）Certificate 要素を参照するために使用される証明書が特定される。転送セキュリティプロトコルが証明書を使用する場合は、CertificateRef 要素が付与されていなければならない。そうでなければ（認証がパスワードによってなされる場合などは）、この要素は省略してもよい。

1.18.16.3 HTTP 特有の問題

HTTP における暗号化のために、プロトコルは、公開鍵暗号化を使用する SSL（Secure Socket Layer）Version 3.0 を採用する。

1.19 DocExchange 要素

DocExchange 要素によって、合意する必要がある文書交換についての情報が指定される。そのような情報には、メッセージ取扱サービスプロパティ（ebXML *メッセージ取扱サービス*[ebMS]など）が含まれている。

CPP の ***DocExchange*** 要素の構造は次の通りである。以降の節では、個々の子要素について詳細に述べる。

```
<DocExchange docExchangeId = "N06">
      <ebXMLBinding version = "0.92">
            <ReliableMessaging>  <!--cardinality 0 or 1-->
            ...
            </ReliableMessaging>
            <NonRepudiation>  <!--cardinality 0 or 1-->
```

```
                ...
            </NonRepudiation>
            <DigitalEnvelope>  <!--cardinality 0 or 1-->
                ...
            </DigitalEnvelope>
            <NamespaceSupported> <!-- 1 or more -->
                ...
            </NamespaceSupported>
        </ebXMLBinding>
</DocExchange>
```

CPP の ***DocExchange*** 要素によって、*CPP* から合成される *CPA* と共に使用されるメッセージ取扱サービスのプロパティが定義される。

DocExchange 要素は、***ebXMLBinding*** という子要素 1 つから成る。

> 注意：文書交換節は、他のサービスを記述する ***xxxBinding*** 要素およびそれらの子要素をさらに追加することで、ebXML メッセージ取扱サービス以外のメッセージ取扱サービスに拡張することができる。これらの他のサービスを使用すると、xxx の部分が、追加されたバインディングの名前に置き換えられる。たとえば、XML プロトコル仕様のサポートを定義する場合の、***XPBinding*** がその例となる。.

1.19.1 docExchangeId 属性

DocExchange 要素には、暗黙の ***docExchangeId*** 属性が付与される。docExchangeId 属性は、*CPP* 文書内の他の場所で参照されてもよいような、[XML]の ID 型の一意な識別子である。

1.19.2 ebXMLBinding 要素

ebXMLBinding 要素は、ebXML *メッセージ取扱サービス*[ebMS]に特有のプロパティを示す。***ebXMLBinding*** 要素は、次に示す子要素から成る。

- 0 個以上の ***ReliableMessaging*** 要素。***ReliableMessaging*** 要素は、信頼性の高いメッセージングの特性を指定する。
- 0 個以上の ***NonRepudiation*** 要素。***NonRepudiation*** 要素は、*メッセージ*への署名のための要件を指定する。
- 0 個以上の ***DigitalEnvelope*** 要素。***DigitalEnvelope*** 要素は、デジタルエンベロープ[DIGENV]メソッドによる暗号化の要件を指定する。
- 0 個以上の ***NamespaceSupported*** 要素。***NamespaceSupported*** 要素は、メッセージ取扱サービスの実装によりサポートされている名前空間の拡張子を特定する。

1.19.3 version 属性

ebXMLBinding 要素には、必須の ***version*** 属性が付与される。***version*** 属性によって、使用されている ebXML *メッセージ取扱サービス*仕様のバージョンが指定される。

1.19.4 ReliableMessaging 要素

ReliableMessaging 要素によって、信頼性の高い ebXML *メッセージ*交換のプロパティが指定される。***ReliableMessaging*** 要素が省略された場合に適用されるデフォルト値は、「BestEffort」である。7.6.4.1 節を参照のこと。要素の構造は次の通りである。

```
<ReliableMessaging deliverySemantics="OnceAndOnlyOnce"
            idempotency="false"
            messageOrderSemantics="Guaranteed">
   <!--The pair of elements Retries, RetryInterval
        has cardinality 0 or 1-->
      <Retries>5</Retries>
      <RetryInterval>60</RetryInterval> <!--time in seconds-->
      <PersistDuration>30S</PersistDuration>
</ReliableMessaging>
```

ReliableMessaging 要素は、次に示す子要素から成る。これらの要素には、0 または 1 の多重度が付与される。すべてが存在するか存在しないかのいずれかでなければならない。

- ***Retries*** 要素
- ***RetryInterval*** 要素
- ***PersistDuration*** 要素

ReliableMessaging 要素の属性は、次の通りである。

- 必須の ***deliverySemantics*** 属性。
- 必須の ***idempotency*** 属性。
- 暗黙の ***messageOrderSemantics*** 属性。

1.19.4.1 deliverySemantics 属性

ReliableMessaging 要素の ***deliverySemantics*** 属性によって、*メッセージ*配信の信頼性の程度が指定される。この属性の値は、次の値のうちのいずれかを列挙したものである。

- "OnceAndOnlyOnce",
- "BestEffort".

「OnceAndOnlyOnce」という値は、*メッセージ*が 1 回だけ配信されなければならないことを示している。「BestEffort」という値は、信頼性の高いメッセージ取扱意味情報は使用されないということを示している。

1.19.4.2 idempotency 属性

ReliableMessaging 要素の ***idempotency*** 属性によって、文書交換層において、交換されるすべてのメッセージに対する等べきテスト（idempotency test; 重複メッセージの適切な処理）の適用を*当事者*が要求するかどうかが指定される。この属性の値は、「true」ま

たは「false」（論理値）である。この属性の値が「true」である場合、すべての*メッセージ*にテストを適用する。値が「false」である場合は、文書交換層においては、*メッセージ*に等べきテストを適用しない。重複に対するテストは、*メッセージ*識別子に基づいて行われる。*メッセージ*ヘッダに含まれている他の情報も、コンテキストによってはテストしてもよい。

> 注意: 重複に対する追加のテストを、*メッセージ*内のアプリケーション情報（購買注文番号など）に基づいて、*ビジネス*アプリケーションにおいて実行してもよい。

通信プロトコルによって重複*メッセージ*が常にチェックされる場合、その通信プロトコルにおけるチェックによって、*CPA* 内の等べき型（idempotency）の仕様が上書きされる。

1.19.4.3 messageOrderSemantics 属性

信頼性の高いメッセージングが有効である場合、***ReliableMessaging*** 要素の ***messageOrderSemantics*** 属性によって、*メッセージ*が受信される順序が制御される（***deliverySemantics*** 属性の値は「OnceAndOnlyOnce」である）。この属性の値は次のいずれかである。

- 「Guaranteed」やり取りごとに、送信側のアプリケーションで指定されている順序で、メッセージが受信側のアプリケーションに渡される。
- 「NotGuaranteed」送信側のアプリケーションで指定されている順序とは異なる順序で、メッセージが受信側のアプリケーションに渡されてもよい。

messageOrderSemantics 属性の値が「Guaranteed」である場合、*メッセージ*の順序付けがやり取りごとに個別に適用されることを理解すること。*メッセージ*の相対順序は指定されない。

messageOrderSemantics 属性のデフォルト値は、「NotGuaranteed」である。この属性は、***deliverySemantics*** 属性の値が「OnceAndOnlyOnce」以外である場合、存在してはならない。

ebXML メッセージ取扱サービス[ebMS]の送信によって、メッセージヘッダ内の ***QualityOfServiceInfo*** 要素の ***messageOrderSemantics*** 属性の値が、*CPA* 内で「送信先」の*当事者*によって指定されている ***messageOrderSemantics*** 属性の値に設定される。

1.19.4.4 Retries 要素と RetryInterval 要素

Retries 要素および ***RetryInterval*** 要素によって、タイムアウト後の依頼の再試行の、許容回数および間隔が秒単位で指定される。***RetryInterval*** 要素の目的は、エラーを発生させた一時的な条件が取り除かれるまで再試行を引き延ばすことにより、再試行における成功率を向上させることにある。

Retries 要素および ***RetryInterval*** 要素は、含まれる場合は、同時に含まれていなければならない。あるいは、省略される場合は、同時に省略されていてもよい。これらの要素が省略される場合、それらの値（再試行回数および再試行間隔）は個々の*当事者*のローカルな問題となる。

1.19.4.5 PersistDuration 要素

PersistDuration 要素の値は、信頼性の高い方法で送付された*メッセージ*のデータがそのメッセージを受信する ebXML の*メッセージ取扱サービス*の実装によって*持続的保存*される時間の最小時間長である。この値は、XML スキーマ[XMLSCHEMA-2]の timeDuration として表現される。

1.19.5 NonRepudiation 要素

非否認によって、メッセージを送信した側を立証すると共に、*メッセージ*内容を後で否認することを避ける。非否認は、XML デジタル署名[XMLDSIG]を用いて*メッセージ*に署名を付けることに基づく。要素の構造は次の通りである。

```
<NonRepudiation>
      <Protocol version = "1.0">XMLDSIG</Protocol>
      <HashFunction>sha1</HashFunction>
      <SignatureAlgorithm>rsa</SignatureAlgorithm>
      <CertificateRef certId = "N03"/>
</NonRepudiation>
```

NonRepudiation 要素が省略されている場合は、メッセージはデジタル署名されていない。

文書交換レベルのセキュリティは、セキュリティが実現されている*取引トランザクション*における双方向の*メッセージ*に適用される。

NonRepudiation 要素は、次に示す子要素から成る。

- 必須の ***Protocol*** 要素。
- 必須の ***HashFunction*** 要素（SHA1、MD5 など）。
- 必須の ***SignatureAlgorithm*** 要素。
- 必須の ***Certificate*** 要素。

1.19.5.1 Protocol 要素

必須の ***Protocol*** 要素によって、*メッセージ*にデジタル署名するために使用される技術を指定する。***Protocol*** 要素には、文字列の値を持つ、暗黙の ***version*** 属性が付与される。***version*** 属性によって、指定された技術のバージョンが特定される。***Protocol*** 要素の例は次の通りである。

```
<Protocol version="2000/10/31">http://www.w3.org/2000/09/xmldsig#
</Protocol>
```

1.19.5.2 HashFunction 要素
必須の ***HashFunction*** 要素によって、署名される*メッセージ*のダイジェストを計算するために使用されるアルゴリズムが特定される。

1.19.5.3 SignatureAlgorithm 要素
必須の ***SignatureAlgorithm*** 要素によって、デジタル署名の値を計算するために使用されるアルゴリズムが特定される。

1.19.5.4 CertificateRef 要素
必須の ***CertificateRef*** 要素の暗黙の ***certId*** IDREF 属性によって、*CPP* 文書内の他の場所にある *Certificate* 要素の 1 つが参照される。

1.19.6 DigitalEnvelope 要素

DigitalEnvelope 要素[DIGENV]は、対称暗号化（共通秘密鍵）によって*メッセージ*が暗号化されると共に、秘密鍵が送付先の公開鍵で暗号化されて*メッセージ*送付先に送信される手順である。要素の構造は次の通りである。

```
<DigitalEnvelope>
      <Protocol version = "2.0">S/MIME</Protocol>
      <EncryptionAlgorithm>rsa</EncryptionAlgorithm>
      <CertificateRef certId = "N03"/>
</DigitalEnvelope>
```

文書交換レベルのセキュリティは、セキュリティが実現されている*取引トランザクション*における双方向の*メッセージ*に適用される。

1.19.6.1 Protocol 要素
必須の ***Protocol*** 要素によって、使用されるセキュリティプロトコルが特定される。固定の ***version*** 属性によって、プロトコルのバージョンが特定される。

1.19.6.2 EncryptionAlgorithm 要素
必須の ***EncryptionAlgorithm*** 要素によって、使用される暗号化アルゴリズムが特定される。

1.19.6.3 CertificateRef 要素
必須の ***CertificateRef*** 要素によって、使用される証明書が ***certId*** 属性を用いて特定される。暗黙の ***certId*** 属性は、[XML]の IDREF タイプの属性である。この属性によって、*CPP* または *CPA* 内の他の場所にある ***Certificate*** 要素の ID 属性のうちの合致するものが参照される。

1.19.7 NamespaceSupported 要素

NamespaceSupported 要素は、メッセージ取扱サービスの実装によりサポートされてい

る名前空間の拡張子を特定する。例としては、Security Services Markup Language [S2ML]および Transaction Authority Markup Language [XAML]がある。たとえば、S2ML 名前空間のサポートは、次の通りに定義される。

```
<NamespaceSupported location = "http://www.s2ml.org/s2ml.xsd"
version = "0.8">http://www.s2ml.org/s2ml</NamespaceSupported>
```

1.20 Packaging 要素

Packaging 要素以下の構造には、*メッセージヘッダ*および搬送内容が転送の上位レベルの伝送のためにパッケージ化される方法についての特殊情報が指定される。その中には、どの文書レベルのセキュリティパッケージングが用いられるか、およびどのような方法でセキュリティ機能が適用されるかなどの重要な情報が含まれている。通常、***Packaging*** 要素の下の構造には、*メッセージ*の構成部分の組織化方法が示される。通常、MIME の処理能力は、この部分に、能力または合意として記述される。Packaging 要素によって、MIME のコンテンツタイプ、XML の名前空間、セキュリティパラメータ、および*当事者*の間で交換される MIME のデータ構造が示される。

Packaging 要素の例を次に示す。

```
<Packaging id="id">
<!--The Packaging triple MAY appear one or more times-->
      <ProcessingCapabilities parse="..."  generate="..."/>
      <SimplePart
            id="id" mimetype="type"/> <!--one or more-->
            <NamespaceSupported location = "" version="">
             URI
             </NamespaceSupported> <!--zero or more-->
       <!--The child of CompositeList is an enumeration of either
       Composite or Encapsulation.  The enumeration MAY appear one
       or more time, with the two elements intermixed-->
       <CompositeList>
            <Composite mimetype="type"
                  id="name"
                  mimeparameters="parameter">
                  <Constituent idref="name"/>
            </Composite>
            <Encapsulation mimetype="type" id="name">
                  <Constituent idref="name"/>
            </Encapsulation>
      </CompositeList>
</Packaging>
```

特定の例は、付録 F の「パッケージングの合致」を参照のこと。

Packaging 要素には、唯一の属性として、必須の ***id*** 属性（ID 型）が付与される。この値は、***ServiceBinding*** 要素および ***Override*** 要素内で、IDREF 属性である ***packageId*** を用いて参照される。

Packaging 要素の子要素は、***ProcessingCapabilities***、***SimplePart***、および ***CompositeList*** である。この要素のセットは、***Packaging*** 要素の子として、*CPP* 内に 1 回以上現れてもよい。また、*CPA* 内に 1 回だけ現れることとする。

1.20.1 ProcessingCapabilities 要素

ProcessingCapabilities 要素には、必須の論理値（「true」または「false」）を持つ属性が 2 つ付与される。2 つの属性とは、***parse*** および ***generate*** である。通常は、これらの属性の値が「true」である場合は、他方の子要素で指定されているパッケージングが、ソフトウェアの*メッセージ取扱サービス*層で作成および処理できることを示す。
generate 属性または ***parse*** 属性のうちの少なくとも 1 つは、「true」でなければならない。

1.20.2 SimplePart 要素

SimplePart 要素によって、MIME コンテンツタイプによって主に指定される構成部分の繰り返し可能なリストが指定される。***SimplePart*** 要素には、***id*** および ***mimetype*** という、2 つの必須の属性が付与される。複合パッケージングが存在している場合、***id*** 属性（ID 型）によって、構成部分が複合パッケージにパッケージ化される方法を指定する際に、この*メッセージ*部分を後で参照するために使用される識別用の値が指定される。
mimetype 属性によって、特定の*メッセージ*部分の実際のコンテンツタイプの値が指定される。

1.20.3 SimplePart 要素

SimplePart 要素には、0 個以上の ***NamespaceSupported*** 要素を付与できる。これらの属性それぞれによって、親要素の本体内でパッケージ化された XML のためにサポートされている名前空間の拡張子が特定される。例としては、Security Services Markup Language [S2ML]および Transaction Authority Markup Language [XAML]がある。たとえば、S2ML 名前空間のサポートは、次の通り定義される。

```
<NamespaceSupported location = "http://www.s2ml.org/s2ml.xsd"
version = "0.8">http://www.s2ml.org/s2ml</NamespaceSupported>
```

1.20.4 CompositeList 要素

Packaging 要素の最後の子要素は、***CompositeList*** である。***CompositeList*** 要素は、部分を組み合わせてグループにするか、またはセキュリティ関連の MIME コンテンツタイプにカプセル化するための特定の方法を指定するコンテナである。***CompositeList*** 要素は、セキュリティカプセル化も複合部分も使用されない場合、***Packaging*** 要素内で省略してもよい。***CompositeList*** 要素が存在する場合、***CompositeList*** 要素のコンテンツモデルは、***Composite*** 要素または ***Encapsulation*** 要素の繰り返し可能な列になる。***Composite*** 要素および ***Encapsulation*** 要素は、必要に応じて、混在してもよい。

MIME パッケージングの再帰的な特性により、***SimplePart*** の木構造内で特徴付けされるメッセージ部分に加えて、上記のような複合体（または稀にカプセル化）を含む場合があるので、選択肢が記述される順序は重要である。したがって、「最上位レベル」のパッケージングは、最後に記述される。

Composite 要素には、次に示す属性が付与される。

- 必須の ***mimetype*** 属性。
- 必須の ***id*** 属性。
- 暗黙の ***mimeparameters*** 属性。

mimetype 属性によって、このメッセージ部分に対する MIME コンテンツタイプの値が示される。その値は、「multipart/related」、「multipart/signed」などの MIME 複合タイプになる。順序内の後の方の要素の構成要素について記述する必要がある場合、***id*** 属性（ID 型）によってこの複合タイプを参照する方法が提供される。***mimeparameters*** 属性によって、コンテンツタイプの処理要求を理解するために必要な、重要な MIME パラメータの値（「type=application/vnd.eb+xml」など）が指定される。

Composite 要素には、***Constituent*** という 1 つの子要素が付与される。

Constituent 要素には、***idref*** という必須の属性（IDREF 型）、および空のコンテンツモデルが付与される。***idref*** 属性には、属性の値として、前の ***Composite*** 要素、***Encapsulation*** 要素、または ***SimplePart*** 要素の ***id*** 属性の値が付与される。この ***Constituents*** 要素の列の目的は、現在の ***Composite*** 要素または ***Encapsulation*** 要素内でパッケージ化されているコンテンツおよびその順序を示すことにある。

通常、***Encapsulation*** 要素は、MIME セキュリティメカニズム（[S/MIME]、Open-PGP [RFC2015]など）の使用を指定するために使用される。セキュリティの本体部分では、既に特徴付けられている MIME 部分をカプセル化できる。技術的にはデータが本体部分の「内部」にない場合でも、このようなセキュリティ構造はすべて、便宜的に、***Encapsulation*** 要素の下に付与されている。（言い換えると、MIME multipart/signed タイプを含む可能性のある署名構造は、簡単のため ***Encapsulation*** 要素の下に置かれている。）

Encapsulation 要素には、次に示す属性が付与される。

- 必須の ***mimetype*** 属性。
- 必須の ***id*** 属性。
- 暗黙の ***mimeparameters*** 属性。

mimetype 属性によって、このメッセージ部分に対する MIME コンテンツタイプの値（「application/pkcs7-mime」など）が示される。順序内の後の方の要素の構成要素について記述する必要がある場合、***id*** 属性（ID 型）によってこのカプセル化を参照する方

法が提供される。***mimeparameters*** 属性によって、コンテンツタイプの処理要求を理解するために必要な、重要な MIME パラメータの値が指定される。

Encapsulation 属性と ***Composite*** 要素の両方には、1 つの ***Constituent*** 要素、または ***Constituent*** 要素の繰り返し可能な順序列から成る子要素が付与される。

1.21 ds:Signature 要素

CPP は、『XML デジタル署名仕様』[XMLDSIG]に準拠した技術を使用してデジタル署名されてもよい。***ds:Signature*** 要素は、*CPP* の署名に使用してもよい要素の階層のルートである。構文は次の通りである。

```
<ds:Signature>...</ds:Signature>
```

ds:Signature 要素および ***ds:Signature*** 要素のサブ要素の内容は、『XML デジタル署名仕様』で定義されている。詳細は、8.7 節を参照のこと。***ds:Signature*** についての追加の拘束条件は次の通りである。

- 『XMLデジタル署名仕様』[XMLDSIG]で定義されているとおり、***ds:Signature***要素がコアの検証に失敗した場合、CPPは無効であると見なされなければならない。

- *CPP*が署名される際には常に、***ProcessSpecification***要素内の***ds: Reference***要素はすべて、参照の検証にパスしなければならない。また、***ds:Signature***要素はすべて、コアの検証にパスしなければならない。

注意: *CPP* が署名されない場合、ソフトウェアは、***ProcessSpecification*** 要素内の ***ds:Reference*** 要素を検証して例外を報告してもよい。

注意: *CPP* および *CPA* を作成するためのソフトウェアは、***ds:Signature*** 要素を認識して、*CPP* および *CPA* の署名の定義に必要な要素構造を挿入してもよい。署名の作成は、本書の範囲外の暗号化プロセスである。

注意: 妥当性テストを実行してもよい時点に関する議論は、7.5.4.5 節の非規範的な記述を参照のこと。

1.22 Comment 要素

CollaborationProtocolProfile 要素には、0 個以上の ***Comment*** 要素を付与してもよい。***Comment*** 要素は、著者が希望する目的に資するため、追加してもよい注意書きである。***Comment*** 要素に使用する言語は、必須の ***xml:lang*** 属性によって特定される。***xml:lang*** 属性は、[XML]で指定されている言語を識別する規則に従う。複数の ***Comment*** 要素が存在する場合は、***xml:lang*** 属性の値には一意な値を付与する必要がある。***Comment*** 要素の例は次の通りである。

```
<Comment xml:lang="en-gb">yadda yadda, blah blah</Comment>
```

2 つの *CPP* から *CPA* を合成する場合、当事者双方が別の方法に合意する場合以外は、両方の CPP の ***Comment*** 要素を *CPA* に含めることとする。

CPA の定義

*コラボレーションプロトコル合意書（CPA）*では、その特定の CPA について、電子*ビジネス*を行う際に利用できる能力についての、*当事者*双方の合意内容が定義される。本節では、*CPA* の詳細を定義および議論する。議論は、XML フラグメントの一部を用いて示される。

本節で述べられているほとんどの XML 要素は、7 節「CPP の定義」でも詳細に説明されている。一般に、本節では、7 節と同じ情報を重複しては述べない。本節における議論は、*CPP* に含まれない要素、または *CPA* のコンテキストにおいて要求される追加の議論に限定する。DTD および XML スキーマの詳細はそれぞれ付録 C および付録 D、*CPA* 文書のサンプルは付録 B を参照のこと。

1.23 CPA の構造

CPA の全体の構造は次の通りである。

```
<CollaborationProtocolAgreement
      xmlns="http://www.ebxml.org/namespaces/tradePartner"
      xmlns:bpm="http://www.ebxml.org/namespaces/businessProcess"
      xmlns:ds = "http://www.w3.org/2000/09/xmldsig#"
      xmlns:xlink = "http://www.w3.org/1999/xlink"
      cpaid="YoursAndMyCPA"
      version="1.2">
      <Status value = "proposed"/>
      <Start>1988-04-07T18:39:09</Start>
      <End>1990-04-07T18:40:00</End>
      <!--ConversationConstraints MAY appear 0 or 1 times-->
      <ConversationConstraints invocationLimit = "100"
            concurrentConversations = "4"/>
      <PartyInfo>
            …
      </PartyInfo>
      <PartyInfo>
            …
      </PartyInfo>
      <Packaging id="N20"> <!--one or more-->
            ...
      </Packaging>
      <!--ds:signature MAY appear 0 or more times-->
      <ds:Signature>any combination of text and elements
      </ds:Signature>
      <Comment xml:lang="en-gb">any text</Comment> <!--zero or more-->
</CollaborationProtocolAgreement>
```

1.24 CollaborationProtocolAgreement 要素

CollaborationProtocolAgreement 要素は、*CPA* のルート要素である。この要素には、文

書に一意な識別子を与える、[XML]の CDATA 型の必須の ***cpaid*** 属性が付与される。***cpaid*** 属性の値は、一方の当事者によって割り当てられ、両方の当事者によって使用される。***cpaid*** 属性の値は、URI であることが推奨される。***cpaid*** 属性の値は、ebXML *メッセージヘッダ*[ebMS]の ***CPAId*** 要素の値として、または別のメッセージ取扱サービスの*メッセージヘッダ*の類似要素の値として使用される場合がある。

注意: それぞれの当事者は、ローカル識別子を ***cpaid*** 属性と関連付けてもよい。

さらに、***CollaborationProtocolAgreement*** 要素には、暗黙の ***version*** 属性が含まれている。この属性は、*CPA* のバージョンを示している。この属性の目的は、当事者間の交渉が進展するのに応じて、*CPA* のインスタンスにバージョン管理機能を持たせることにある。***version*** 属性は、配備および修正された *CPA* にバージョン管理機能を持たせるために使用する必要がある。***version*** 属性の値は、「1.0」、「2.3」などの数値を表す文字列表現である必要がある。バージョン文字列の値は、交渉中および配備後の *CPA* 文書が変更された際には、必ず変更する必要がある。

注意: バージョン識別子の値の割り当て方法は、実装に委ねられている。

CollaborationProtocolAgreement 要素には、7 節「CPP の定義」で詳細に説明されている、必須の [XML]名前空間[XMLNS]宣言が付与される。

CollaborationProtocolAgreement 要素は、次に示す子要素から成る。個々の要素については以降の節で詳細に説明する。

- 必須の ***Status*** 要素。***Status*** 要素は、*CPA* を作成するプロセスの状態を特定する。
- 必須の ***Start*** 要素。***Start*** 要素は、*CPA* の効力が発生した日付と時刻を記録する。
- 必須の ***End*** 要素。***End*** 要素は、当事者が *CPA* について再交渉する必要がある日付と時刻を記録する。
- 0 個以上の ***ConversationConstraints*** 要素。この要素は、やり取りの処理についての合意内容を文書化する。
- 必須の 2 つの ***PartyInfo*** 要素。*CPA* の*当事者*ごとに 1 つ用意する。
- 1 つ以上の ***ds:Signature*** 要素。ds:Signature 要素は、『XML デジタル署名』[XMLDSIG]規格を用いた CPA への署名を提供する。

1.25 Status 要素

Status 要素は、*CPA* を作成するための合成/交渉プロセスの状態を記録する。***Status*** 要素の例は次の通りである。

```
<Status value = "proposed"/>
```

Status 要素には、*CPA* 合成の現在の状態を記録するための、必須の ***value*** 属性が付与さ

れる。この属性の値は、次の値のいずれかを列挙したものである。

- 「proposed」。*CPA* について*当事者*がまだ交渉を行っている状態。
- 「agreed」。*CPA* のコンテンツについて*当事者*双方が合意した状態。
- 「signed」。*CPA* について*当事者*双方が署名した状態。この「署名」は、次の 8.7 節で述べられているデジタル署名の形で行われてもよい。

注意: ***Status*** 要素は、*CPA* の作成プロセスを支援するために、*CPA* の合成および交渉ツールで使用してもよい。

1.26 CPA の有効期限

CPA の有効期限は、***Start*** 要素および ***End*** 要素によって指定される。構文は次の通りである。

```
<Start>1988-04-07T18:39:09</Start>
<End>1990-04-07T18:40:00</End>
```

1.26.1 Start 要素

Start 要素によって、*CPA* の開始の日付と時刻が指定される。***Start*** 要素の値は、文字列であり、『XML スキーマデータ型仕様』[XMLSCHEMA-2]で定義されているコンテンツモデルに準拠する。たとえば、1999 年 3 月 31 日 1:20pm UTC（協定世界時）を示す ***Start*** 要素の値は次の通りである。

```
1999-05-31T13:20:00Z
```

Start 要素は、協定世界時（UTC）で表現する。

1.26.2 End 要素

End 要素によって、*CPA* の終了の日付と時刻が指定される。***End*** 要素の値は、文字列であり、『XML スキーマデータ型仕様』[XMLSCHEMA-2]で定義されているコンテンツモデルに準拠する。たとえば、1999 年 3 月 31 日 1:20pm UTC（協定世界時）を示す ***End*** 要素の値は次の通りである。

```
1999-05-31T13:20:00Z
```

End 要素は、協定世界時（UTC）で表現する。

CPA の有効期限に達した場合、進行中の*取引トランザクション*は完了することが許される。また、新しい*取引トランザクション*が開始することはない。やり取り上で進行中のすべての*取引トランザクション*が終了し、完了していない*やり取り*も終了する。

注意: *ビジネスアプリケーション*が複数の*取引トランザクション*から成るやり取りを定義している場合、有効期限に達した際に、エラー表示なしにやり取りが終了してもよいことを理解すること。実行時システムが、アプリケーションに対してエラー表示を送る場合もある。

CPA を終了する前に重要なやり取りが終了するまで待つという方法は、やり取りが無制限に続く場合があるので、現実的ではない場合があることを理解すること。

注意: 実行時システムは、***End*** 要素で指定されている日付と時刻の後に、*CPA* 内で新しい*取引トランザクション*が開始されたときには、両方の*当事者*にエラー表示を返送する必要がある。

1.27 ConversationConstraints 要素

ConversationConstraints 要素によって、*CPA* 内のやり取りの回数が制限される。この要素の例は次の通りである。

```
<ConversationConstraints invocationLimit = "100"
                         concurrentConversations = "4"/>
```

ConversationConstraints 要素には、次に示す属性が付与される。

- 暗黙の ***invocationLimit*** 属性。
- 暗黙の ***concurrentConversations*** 属性。

1.27.1 invocationLimit 属性

invocationLimit 属性によって、*CPA* 内で処理できるやり取り数の上限が定義される。この数値に達したときに、*CPA* は終了される。その際、*CPA* は再交渉されなければならない。値が指定されていない場合は、やり取り数に上限はない。CPA の有効期限は、End 要素のみによって制御される。

注意: ***invocationLimit*** 属性によって、*CPA* 内で実行できる*ビジネス*ユニットの個数が制限される。これは、*ビジネス*パラメータであってパフォーマンスパラメータではない。

1.27.2 concurrentConversations 属性

concurrentConversations 属性によって、*CPA* 内で同時に処理できるやり取り数の上限が定義される。値が指定されていない場合は、やり取りの並行処理についてはローカルな問題となる。

注意: ***concurrentConversations*** 属性によって、特定の CPA 内で並行して処理できる

やり取りの個数を制限する必要がある場合に*当事者*が使用するパラメータが提供される。たとえば、バックエンドプロセスがサポートするやり取りの並行処理の個数に制限がある場合がある。新しいやり取りの要求が受信された際に、*CPA* で許されているやり取り数の上限に既に達している場合、実装は、新しいやり取りを拒否してもよい。または、既に実行中のやり取りが終了するまで要求を待ち行列に入れてもよい ***concurrentConversations*** 属性の値が付与されていない場合は、許容範囲を超える新しいやり取りの要求の処理方法は、ローカルの実装上の問題になる。

1.28 PartyInfo 要素

PartyInfo 要素の一般的な特性は、7.5 節で議論されている。

CPA には、*CPA* の*当事者*ごとに 1 つの ***PartyInfo*** 要素が付与されることとする ***PartyInfo*** 要素によって、*CPA* が参照している*プロセス仕様*文書によって定義される、*当事者*が合意した*取引/コラボレーション*の条件が指定される。*CPP* に複数の ***PartyInfo*** 要素が付与される場合、*CPA* を合成する際に、適切な ***PartyInfo*** 要素をそれぞれの *CPP* から選択することとする。

CPA には、***PartyInfo*** 要素ごとに、1 つの ***PartyId*** 要素が存在することとする。この要素の値は、『ebXML メッセージ取扱サービス仕様』[ebMS]または同様メッセージ取扱サービス仕様の ***PartyId*** 要素の値と同じである。ebXML *メッセージ*の ***To（送付先）*** または ***From（送付元）*** *ヘッダ*要素内では、1 つの ***PartyId*** 要素を使用する。

1.28.1 ProcessSpecification 要素

ProcessSpecification 要素によって、*当事者*双方が実行に合意した*取引コラボレーション*が特定される。1 つの *CPA* 内には、1 つ以上の ***ProcessSpecification*** 要素が存在してもよい。それぞれが別々の ***CollaborationRole*** 要素の子要素であるとする 7.5.3 節の議論を参照のこと。

1.29 ds:Signature 要素

CPA 文書は、データの保全性を保証するため、および合意を表現する（書類に会社役員の署名を入れるのと同様）ために、一方または両方の*当事者*によってデジタル署名されてもよい。***ds:Signature*** 要素は、*CPP* の署名に使用してもよい要素の階層上の最上位（ルート）に位置する。構文は次の通りである。

```
<ds:Signature>...</ds:Signature>
```

ds:Signature 要素および ***ds:Signature*** 要素のサブ要素の内容は、『XML デジタル署名仕様』[XMLDSIG]で定義されている。***ds:Signature*** についての追加の拘束条件は次の通りである。

- 『XMLデジタル署名仕様』で定義されているとおり、***ds:Signature***要素がコアの検証に失敗した場合、*CPA*は無効であると見なされなければならない。

- *CPA*が署名される際には常に、***ProcessSpecification***要素内の***ds: Reference***要素はすべて、参照の検証にパスしなければならない。また、***ds:Signature***要素はすべて、コアの検証にパスしなければならない。

注意: *CPA*が署名されない場合、ソフトウェアは、***ProcessSpecification***要素内の***ds:Reference***要素を検証して例外をレポートしてもよい。

注意: *CPP* および *CPA* を作成するためのソフトウェアは、***ds:Signature*** 要素を認識して、*CPP* および *CPA* の署名の定義に必要な要素構造を挿入してもよい。署名の作成は、本書の範囲外の暗号化プロセスである。

注意: *CPA* を有効にしてもよい時点に関する議論は、7.5.4.5 節の非規範的な記述を参照のこと。

1.29.1 持続的デジタル署名

[XMLDSIG]を使用して ebXML の *CPP* または *CPA* に署名する場合は、本節で定義されているプロセスが使用される。

1.29.1.1 署名の世代

デジタル署名を作成する手順は、次の通りである。

1. ***ds:Signature*** 要素の子要素として ***SignedInfo*** 要素を作成する。[XMLDSIG]に示されているように、***SignedInfo*** 要素には、***SignatureMethod***、***CanonicalizationMethod***、および ***Reference*** という子要素が付与されることとする。
2. [XMLDSIG]に示されているように、***SignedInfo*** に指定されているアルゴリズムに基づいて、***SignedInfo*** 内の ***SignatureValue*** を正規化し、計算する。
3. [XMLDSIG]に示されているように、***SignedInfo***、***KeyInfo***（推奨）、および ***SignatureValue*** 要素を含む ***Signature*** 要素を作成する。
4. 最後の ***PartyInfo*** 要素の次に、署名されたばかりの文書内の ***Signature*** 要素を挿入する。

1.29.1.2 ds:SignedInfo 要素

ds:SignedInfo 要素は、0 個以上の ***ds:CanonicalizationMethod*** 要素、***ds:SignatureMethod*** 要素、および 1 つ以上の ***ds:Reference*** 要素から成る。

1.29.1.3 ds:CanonicalizationMethod 要素

ds:CanonicalizationMethod 要素は、[XMLDSIG]で選択できると定義されている。すなわち、***ds:SignedInfo*** 要素のインスタンスに現れる必要はない。***ds:CanonicalizationMethod*** 要素が存在しない場合は、署名されるデータに適用されるデフォルトの正規化メソッド

は[XMLC14N]である。このデフォルト値は、ebXML の CPP および CPA 文書のデフォルトの正規化メソッドの役割も果たす。

1.29.1.4 ds:SignatureMethod 要素

ds:SignatureMethod 要素は、存在する。また、***ds:SignatureMethod*** 要素には、***Algorithm*** 属性が付与される。***Algorithm*** 属性の値として推奨される値は、次の通りである。

http://www.w3.org/2000/09/xmldsig#dsa-sha1

この推奨値は、ebXML の *CPP* または *CPA* 準拠のすべてのソフトウェア実装によってサポートされる。

1.29.1.5 ds:Reference 要素

CPP または *CPA* 文書の ***ds:Reference*** 要素は、必須の URI 属性の値として ""を付与される。これにより、***ds:Signature*** 要素を含む文書（*CPA* または *CPP* 文書）に署名が適用される。[XMLDSIG]に従って、*CPP* または *CPA* 文書の ***ds:Reference*** 要素には、次の値を持つ暗黙の ***type*** 属性が含まれていてもよい。

```
"http://www.w3.org/2000/09/xmldsig#Object"
```

この属性の情報はそのままでも有用なものである。また、この属性は省略してもよい。ebXML の *CPA* または *CPP* 文書を作成および処理するためのソフトウェアは、これらいずれかの場合を処理するように実装する。***ds:Reference*** 要素には、***id*** 属性（ID 型）を付与してもよい。***id*** 属性を用いて、***ds:Signature*** 要素から ***ds:Reference*** 要素を参照してもよい。

1.29.1.6 ds:Transform 要素

CPA または *CPP* 文書の ***ds:Reference*** 要素には、リンク元の ***ds:Signature*** 要素とそのすべての子孫要素を除外する、***ds:Transform*** 要素が付与されることとする。次の場所で示されているように、この除外の操作は、***Transform*** 要素の ***ds:Algorithm*** 属性を指定する方法から得られる。

```
"http://www.w3.org/2000/09/xmldsig#enveloped-signature".
```

たとえば、次の通りである。

```
<ds:Reference ds:URI="">
      <ds:Transforms>
            <ds:Transform
ds:Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-signature "/>
      </ds:Transforms>
      <ds:DigestMethod
 ds:Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
       <ds:DigestValue>...</ds:DigestValue>
</ds:Reference>
```

1.29.1.7 ds:Xpath 要素

ds:Transform 要素には、次の値を持つ ***ds:Algorithm*** 属性が付与される。

```
http://www.w3.org/2000/09/xmldsig#enveloped-signature
```

注意: *CPA* にデジタル署名する際、それぞれの*当事者*が上記のプロセスに従って文書に署名することが推奨される。*CPA* に最初に署名する*当事者*は、*CPA* のコンテンツにのみ署名する。2 番目の*当事者*は、*CPA* のコンテンツ、および最初の*当事者*の署名が付与された ***ds:Signature*** 要素に署名する。両当事者の署名に対し、公証人の署名が必要となる場合もある。

1.30 Comment 要素

CollaborationProtocolAgreement 要素には、0 個以上の ***Comment*** 要素を付与してもよい。**Comment** 要素の構文の詳細は、7.9 節を参照のこと。

1.31 2 つの CPP から 1 つの CPA を合成する

本節では、2 つの *CPP* から *CPA* を合成する際の規範的な問題について議論する。付録 F「2 つの *CPP* から *CPA* を合成する（非規範的）」も参照のこと。

1.31.1 ID 属性の重複

2 つの *CPP* から *CPA* を合成する際には、2 つの *CPP* の ID 属性が重複する値を持つ危険がある。2 つの *CPP* から *CPA* を合成する場合、重複する ID 属性の値をテストすることとする。ID 属性の値が重複している場合は、重複している属性の 1 つに新しい値を与え、対応する *CPP* の対応する IDREF 属性の値を修正する。

1.32 CPA 情報のプロセス仕様文書パラメータの修正

*プロセス仕様*文書は、XML 属性で表現する多数のパラメータを含む。*CPA* の ***Characteristics*** 要素の属性に対応するセキュリティ属性がその例である。これらの属性の値は、デフォルト値または推奨値であると見なすことができる。*CPA* が作成される際に、*当事者*によって、*プロセス仕様*の中で推奨値の受け入れが決定されてもよい。あるいは、パラメータの値として、ニーズにより適合した値に合意してもよい。

CPA を使用して実行時システムのシステム構成を設定する場合は、*CPA* で示されている選択肢が、参照されている*プロセス仕様*文書で示されている選択肢を上書きしなければならない。特に、*CPA* の ***Characteristics*** 要素および ***Packaging*** 要素によって表現されるすべての選択肢は、*当事者*双方によって合意されたとおりに実装されなければならない。これらの選択肢は、*プロセス仕様*文書に示されているデフォルト値に優先する。*CPA* および*プロセス仕様*文書からの情報をインストールするプロセスでは、結果の選択肢がすべて相互に一貫していることが確認されなければならない。また、一貫していない場合はエラー信号が発信されなければならない。

注意: *CPA* からの情報が*プロセス仕様*文書からの情報を上書きする方法には、い

くつかの種類がある。たとえば、次の通りである。

- *CPA* 合成ツールが*プロセス仕様*文書の別のコピーを作成する場合がある。コピーが作成されれば、このツールによって、*プロセス仕様*文書を *CPA* からの情報によって直接修正できるようになる。この方法の第 1 の利点は、上書き処理の全体が *CPA* 合成ツールによって実行される点にある。また、第 2 の利点は、特定の *CPA* に*プロセス仕様*文書の個別のコピーを関連付けることによって、*CPA* が作成される時点と*当事者*システムにインストールされる時点の間、*プロセス仕様*文書の修正内容が開示されない点である。
- *CPA* インストールツールが、*当事者*システムに *CPA* および*プロセス仕様*文書がインストールされる際に、*CPA* 内の対応するパラメータからの情報を用いて*プロセス仕様*文書内のパラメータを動的に上書きする場合がある。これによって、*プロセス仕様*文書の個別のコピーを作成する必要性がなくなる。
- もう 1 つ考えられる方法は、*CPA*、*プロセス仕様*文書、またはその両方のパラメータ情報の XSLT 変換に基づく場合である。

関連文書

下に列挙されている関連文書の中には、*CPP* および *CPA* 内で示されている特定の XML 定義に基づいて特定される機能を指定するための文書もある。また、本書で仕様を参照する際は、プラグインを取得するか、または *CPP* および *CPA* に設定されている特定の XML 要素を必要としないカスタムサポートソフトウェアを作成してもよい場合について、キーワードとして用いられる用語が示される。

ある機能に関する仕様として、権限が必要な仕様しか入手可能でない場合もある。その場合は、下の引用内に明示されている。

[ccOVER] ebXML Core Components and Business Process Document Overview, http://www.ebxml.org.

[DIGENV] Digital Envelope, RSA Laboratories, http://www.rsasecurity.com/rsalabs/. 注意: 現在の時点で、デジタルエンベロープに関する入手可能な仕様は、RSA 研究所の仕様だけのようである。

[ebBPSS] ebXML ビジネスプロセス仕様スキーマ, http://www.ebxml.org.

[ebGLOSS] ebXML Glossary, http://www.ebxml.org.

[ebMS] ebXML Message Service Specification, http://www.ebxml.org.

[ebRS] ebXML Registry Services Specification, http://www.ebxml.org.

[ebTA] ebXML Technical Architecture Specification, http://www.ebxml.org.

[HTTP] Hypertext Transfer Protocol, Internet Engineering Task Force RFC2616.

[IPSEC] IP Security Document Roadmap, Internet Engineering Task Force RFC 2411.

[ISO6523] Structure for the Identification of Organizations and Organization Parts, International Standards Organization ISO-6523.

[MIME] MIME (Multipurpose Internet Mail Extensions) Part One: Mechanisms for Specifying and Describing the Format of Internet *Message* Bodies. Internet Engineering Task Force RFC 1521.

[RFC822] Standard for the Format of ARPA Internet Text Messages, Internet Engineering Task Force RFC 822.

[RFC959] File Transfer Protocol (FTP), Internet Engineering Task Force RFC 959.

[RFC1123] Requirements for Internet Hosts -- Application and Support, R. Braden, Internet Engineering Task Force, October 1989.

[RFC1579] Firewall-Friendly FTP, S. Bellovin, Internet Engineering Task Force, February 1994.

[RFC2015] MIME Security with Pretty Good Privacy, M. Elkins, Internet Engineering Task Force, RFC 2015.

[RFC2119] Key Words for use in RFCs to Indicate Requirement Levels, Internet Engineering Task Force RFC 2119.

[RFC2396] Uniform Resource Identifiers (URI): Generic Syntax; T. Berners-Lee, R. Fielding, L. Masinter - August 1998.

[S/MIME] S/MIME Version 3 Message Specification, Internet Engineering Task Force RFC 2633.

[S2ML] Security Services Markup Language, http://s2ml.org/.

[SMTP] Simple Mail Transfer Protocol, Internet Engineering Task Force RFC 821.

[SSL] Secure Sockets Layer, Netscape Communications Corp. http://developer.netscape.com. 注意: 現在の時点で、SSL に関する入手可能な仕様は、Netscape の仕様だけのようである。SSL に置き換わるものとして予定されている「トランスポート層セキュリティ」は、IETF において作成中である。

[XAML] Transaction Authority Markup Language, http://xaml.org/.

[XLINK] XML Linking Language, http://www.w3.org/TR/xlink/.

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

[XMLC14N] Canonical XML, Ver. 1.0, http://www.w3.org/TR/XML-C14N/.

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

[XMLNS] Namespaces in XML, T. Bray, D. Hollander, and A. Layman, Jan. 1999, http://www.w3.org/TR/REC-xml-names/.

[XMLSCHEMA-1] XML Schema Part 1: Structures, http://www/w3/org/TR/xmlschema-1/.

[XMLSCHEMA-2] XML Schema Part 2: Datatypes, http://www.w3.org/TR/xmlschema-2/.

[XPOINTER] XML Pointer Language, ver. 1.0, http://www.w3.org/TR/xptr.

適合性

本書に準拠するには、実装において次を実現する。

a) 本書で定義されている機能およびインタフェースの要件をすべてサポートすることとする。
b) 本書と矛盾する、または本書への非適合性の原因となるような要件を規定しない。

本書に準拠している実装は、本書の適用可能な部分の適合要件を満たしていることとする。

AebXMLのCPPまたはCPAのインスタンス文書を作成または保守するためのツールまたはサービスの実装は、付録Aおよび次の場所から入手可能なCPPまたはCPAに関するXMLスキーマ[XMLSCHEMA-1]の定義に基づいて、そのツールまたはサービスによって作成または修正されたCPPまたはCPAのインスタンス文書を、W3CのXMLスキーマ仕様[XMLSCHEMA-1, XMLSCHEMA-2]に準拠した2つ以上のXMLスキーマ検証パーサを用いて検証することにより、適合性が認められることとする。

```
http://www.ebxml.org/schemas/cpp-cpa-v1_0.xsd
```

この適合性テストの目的は、テストされている実装が本書で述べられている要件に準拠しているかどうかを見極めることにある。適合性テストによって、ベンダは、互換性のある相互運用可能なシステムを実装できる。実装およびアプリケーションは、入手可能なテストスイートを用いて、本書への適合性を確認することによりテストされることとする。

OASIS、U.S.A. National Institute of Science and Technology（NIST）などのベンダから独立した団体から公に入手可能なテストスイーツは、本書への適合性を主張する、実装、アプリケーション、およびコンポーネントの適合性を検査するために使用される必要がある。さらに、ベンダが自社の製品のインタフェースの互換性、適合性、および相互運用性をテストできるようにしたオープンソース実装が入手可能であってもよい。

免責

本書の記述内容は各著者の個人的な見解 / 仕様であり、所属企業の従業員の見解 / 仕様と必ずしも一致するとは限らない。本書の記述を使用した結果 (使用法が正しいかどうかの如何を問わない) 不都合が生じたとしても、著者および所属企業の従業員は一切、責任を負うものではない。

連絡先

Martin W. Sachs (Team Leader)
IBM T. J. Watson Research Center
P.O.B. 704
Yorktown Hts, NY 10598
USA
Phone: 914-784-7287
email: mwsachs@us.ibm.com

Chris Ferris
XML Technology Development
Sun Microsystems, Inc
One Network Drive
Burlington, Ma 01824-0903
USA
Phone: 781-442-3063
email: chris.ferris@east.sun.com

Dale W. Moberg
Cyclone Commerce
17767 North Perimeter Dr., Suite 103
Scottsdale, AZ 85255
USA
Phone: 480-627-1800
email: dmoberg@columbus.rr.com

Tony Weida
Edifecs
2310 130th Ave. NE, Suite 100
Bellevue, WA 98005
USA
Phone: 212-678-5265
email: TonyW@edifecs.com

Copyright Statement

付録 A　CPP 文書の例（非規範的）

この例は、次の ASCII ファイルでも入手可能である。

http://ebxml.org/project_teams/trade_partner/cpp-example.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<tp:CollaborationProtocolProfile
       xmlns:tp="http://www.ebxml.org/namespaces/tradePartner"
       xmlns:xsi="http://www.w3.org/2000/10/XMLSchema-instance"
       xmlns:xlink="http://www.w3.org/1999/xlink"
       xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
       xsi:schemaLocation="http://www.ebxml.org/namespaces/tradePartner
http://ebxml.org/project_teams/trade_partner/cpp-cpa-v1_0.xsd"
       tp:version="1.1">
       <tp:PartyInfo>
              <tp:PartyId tp:type="DUNS">123456789</tp:PartyId>
              <tp:PartyRef tp:href="http://example.com/about.html"/>
              <tp:CollaborationRole tp:id="N00">
                     <tp:ProcessSpecification tp:version="1.0" tp:name="buySell"
xlink:type="simple" xlink:href="http://www.ebxml.org/processes/buySell.xml"/>
                     <tp:Role tp:name="buyer" xlink:type="simple"
xlink:href="http://ebxml.org/processes/buySell.xml#buyer"/>
                     <tp:CertificateRef tp:certId="N03"/>
                     <tp:ServiceBinding tp:channelId="N04" tp:packageId="N0402">
                            <tp:Service
tp:type="uriReference">uri:example.com/services/buyerService</tp:Service>

                            <tp:Override tp:action="orderConfirm" tp:channelId="N07"
tp:packageId="N0402" xlink:href="http://ebxml.org/processes/buySell.xml#orderConfirm"
xlink:type="simple"/>
                     </tp:ServiceBinding>
              </tp:CollaborationRole>
              <tp:Certificate tp:certId="N03">
                     <ds:KeyInfo/>
              </tp:Certificate>
              <tp:DeliveryChannel tp:channelId="N04" tp:transportId="N05"
tp:docExchangeId="N06">
                     <tp:Characteristics tp:syncReplyMode="none"
tp:nonrepudiationOfOrigin="true" tp:nonrepudiationOfReceipt="false"
tp:secureTransport="true" tp:confidentiality="true" tp:authenticated="true"
tp:authorized="false"/>
              </tp:DeliveryChannel>
              <tp:DeliveryChannel tp:channelId="N07" tp:transportId="N08"
tp:docExchangeId="N06">
                     <tp:Characteristics tp:syncReplyMode="none"
tp:nonrepudiationOfOrigin="true" tp:nonrepudiationOfReceipt="false"
tp:secureTransport="false" tp:confidentiality="true" tp:authenticated="true"
tp:authorized="false"/>
              </tp:DeliveryChannel>
              <tp:Transport tp:transportId="N05">
                     <tp:SendingProtocol tp:version="1.1">HTTP</tp:SendingProtocol>
                     <tp:ReceivingProtocol tp:version="1.1">HTTP</tp:ReceivingProtocol>
                     <tp:Endpoint
tp:uri="https://www.example.com/servlets/ebxmlhandler" tp:type="allPurpose"/>
                     <tp:TransportSecurity>
                            <tp:Protocol tp:version="3.0">SSL</tp:Protocol>
                            <tp:CertificateRef tp:certId="N03"/>
                     </tp:TransportSecurity>
              </tp:Transport>
              <tp:Transport tp:transportId="N08">
```

```
                  <tp:SendingProtocol tp:version="1.1">HTTP</tp:SendingProtocol>
                  <tp:ReceivingProtocol tp:version="1.1">SMTP</tp:ReceivingProtocol>
                  <tp:Endpoint tp:uri="mailto:ebxmlhandler@example.com"
tp:type="allPurpose"/>
            </tp:Transport>
            <tp:DocExchange tp:docExchangeId="N06">
                  <tp:ebXMLBinding tp:version="0.98b">
                        <tp:ReliableMessaging tp:deliverySemantics="OnceAndOnlyOnce"
tp:idempotency="true" tp:messageOrderSemantics="Guaranteed">
                              <tp:Retries>5</tp:Retries>
                              <tp:RetryInterval>30</tp:RetryInterval>
                              <tp:PersistDuration>P1D</tp:PersistDuration>
                        </tp:ReliableMessaging>
                        <tp:NonRepudiation>

      <tp:Protocol>http://www.w3.org/2000/09/xmldsig#</tp:Protocol>

      <tp:HashFunction>http://www.w3.org/2000/09/xmldsig#sha1</tp:HashFunction>

      <tp:SignatureAlgorithm>http://www.w3.org/2000/09/xmldsig#dsa-
sha1</tp:SignatureAlgorithm>
                              <tp:CertificateRef tp:certId="N03"/>
                        </tp:NonRepudiation>
                        <tp:DigitalEnvelope>
                              <tp:Protocol tp:version="2.0">S/MIME</tp:Protocol>
                              <tp:EncryptionAlgorithm>DES-
CBC</tp:EncryptionAlgorithm>
                              <tp:CertificateRef tp:certId="N03"/>
                        </tp:DigitalEnvelope>
                  </tp:ebXMLBinding>
            </tp:DocExchange>
      </tp:PartyInfo>
      <tp:Packaging tp:id="N0402">
            <tp:ProcessingCapabilities tp:parse="true" tp:generate="true"/>
            <tp:SimplePart tp:id="N40" tp:mimetype="text/xml">
                  <tp:NamespaceSupported
tp:location="http://ebxml.org/project_teams/transport/messageService.xsd"
tp:version="0.98b">http://www.ebxml.org/namespaces/messageService</tp:NamespaceSupport
ed>
                  <tp:NamespaceSupported
tp:location="http://ebxml.org/project_teams/transport/xmldsig-core-schema.xsd"
tp:version="1.0">http://www.w3.org/2000/09/xmldsig</tp:NamespaceSupported>
            </tp:SimplePart>
            <tp:SimplePart tp:id="N41" tp:mimetype="text/xml">
                  <tp:NamespaceSupported
tp:location="http://ebxml.org/processes/buysell.xsd"
tp:version="1.0">http://ebxml.org/processes/buysell.xsd</tp:NamespaceSupported>
            </tp:SimplePart>
            <tp:CompositeList>
                  <tp:Composite tp:id="N42" tp:mimetype="multipart/related"
tp:mimeparameters="type=text/xml;">
                        <tp:Constituent tp:idref="N40"/>
                        <tp:Constituent tp:idref="N41"/>
                  </tp:Composite>
            </tp:CompositeList>
      </tp:Packaging>
      <tp:Comment tp:xml_lang="en-us">buy/sell agreement between example.com and
contrived-example.com</tp:Comment>
</tp:CollaborationProtocolProfile>
```

付録 B　CPA 文書の例（非規範的）

この付録に示されている例は、XML スキーマパーサによって構文解析できるように作られている。この例は、次の ASCII ファイルでも入手可能である。

http://ebxml.org/project_teams/trade_partner/cpa-example.xml

DTD で構文解析できる例は、次の場所から入手可能である。

http://ebxml.org/project_teams/trade_partner/cpa-example-dtd.xml

注意: 少なくとも既存のツールのいくつかは、DTD を割り当てるための <!DOCTYPE...>を持つこと、および名前空間限定子を持たないことを要求するので、2 つの独立した *CPA* の例が必要である。

```
<?xml version="1.0"?>
<!-- edited with XML Spy v3.5 (http://www.xmlspy.com) by christopher ferris (sun
microsystems, inc) -->
<tp:CollaborationProtocolAgreement
       xmlns:tp="http://www.ebxml.org/namespaces/tradePartner"
       xmlns:xsi="http://www.w3.org/2000/10/XMLSchema-instance"
       xsi:schemaLocation="http://www.ebxml.org/namespaces/tradePartner
http://ebxml.org/project_teams/trade_partner/cpp-cpa-v1_0.xsd"
       xmlns:xlink="http://www.w3.org/1999/xlink"
       xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
       tp:cpaid="uri:yoursandmycpa"
       tp:version="1.2">
       <tp:Status tp:value="proposed"/>
       <tp:Start>2001-05-20T07:21:00Z</tp:Start>
       <tp:End>2002-05-20T07:21:00Z</tp:End>
       <tp:ConversationConstraints tp:invocationLimit="100"
tp:concurrentConversations="100"/>
       <tp:PartyInfo>
              <tp:PartyId tp:type="DUNS">123456789</tp:PartyId>
              <tp:PartyRef xlink:href="http://example.com/about.html"/>
              <tp:CollaborationRole tp:id="N00">
                     <tp:ProcessSpecification tp:version="1.0" tp:name="buySell"
xlink:type="simple" xlink:href="http://www.ebxml.org/processes/buySell.xml"/>
                     <tp:Role tp:name="buyer" xlink:type="simple"
xlink:href="http://ebxml.org/processes/buySell.xml#buyer"/>
                     <tp:CertificateRef tp:certId="N03"/>
                     <tp:ServiceBinding tp:channelId="N04" tp:packageId="N0402">
                            <tp:Service
tp:type="uriReference">uri:example.com/services/buyerService</tp:Service>
                            <tp:Override tp:action="orderConfirm" tp:channelId="N08"
tp:packageId="N0402" xlink:href="http://ebxml.org/processes/buySell.xml#orderConfirm"
xlink:type="simple"/>
                     </tp:ServiceBinding>
              </tp:CollaborationRole>
              <tp:Certificate tp:certId="N03">
                     <ds:KeyInfo/>
              </tp:Certificate>
              <tp:DeliveryChannel tp:channelId="N04" tp:transportId="N05"
tp:docExchangeId="N06">
                     <tp:Characteristics tp:syncReplyMode="none"
tp:nonrepudiationOfOrigin="true" tp:nonrepudiationOfReceipt="false"
tp:secureTransport="true" tp:confidentiality="true" tp:authenticated="true"
tp:authorized="false"/>
```

```
            </tp:DeliveryChannel>
            <tp:DeliveryChannel tp:channelId="N07" tp:transportId="N08"
tp:docExchangeId="N06">
                  <tp:Characteristics tp:syncReplyMode="none"
tp:nonrepudiationOfOrigin="true" tp:nonrepudiationOfReceipt="false"
tp:secureTransport="false" tp:confidentiality="true" tp:authenticated="true"
tp:authorized="false"/>
            </tp:DeliveryChannel>
            <tp:Transport tp:transportId="N05">
                  <tp:SendingProtocol tp:version="1.1">HTTP</tp:SendingProtocol>
                  <tp:ReceivingProtocol tp:version="1.1">HTTP</tp:ReceivingProtocol>
                  <tp:Endpoint
tp:uri="https://www.example.com/servlets/ebxmlhandler" tp:type="allPurpose"/>
                  <tp:TransportSecurity>
                        <tp:Protocol tp:version="3.0">SSL</tp:Protocol>
                        <tp:CertificateRef tp:certId="N03"/>
                  </tp:TransportSecurity>
            </tp:Transport>
            <tp:Transport tp:transportId="N18">
                  <tp:SendingProtocol tp:version="1.1">HTTP</tp:SendingProtocol>
                  <tp:ReceivingProtocol tp:version="1.1">SMTP</tp:ReceivingProtocol>
                  <tp:Endpoint tp:uri="mailto:ebxmlhandler@example.com"
tp:type="allPurpose"/>
            </tp:Transport>
            <tp:DocExchange tp:docExchangeId="N06">
                  <tp:ebXMLBinding tp:version="0.98b">
                        <tp:ReliableMessaging tp:deliverySemantics="OnceAndOnlyOnce"
tp:idempotency="true" tp:messageOrderSemantics="Guaranteed">
                              <tp:Retries>5</tp:Retries>
                              <tp:RetryInterval>30</tp:RetryInterval>
                              <tp:PersistDuration>P1D</tp:PersistDuration>
                        </tp:ReliableMessaging>
                        <tp:NonRepudiation>

      <tp:Protocol>http://www.w3.org/2000/09/xmldsig#</tp:Protocol>

      <tp:HashFunction>http://www.w3.org/2000/09/xmldsig#sha1</tp:HashFunction>

      <tp:SignatureAlgorithm>http://www.w3.org/2000/09/xmldsig#dsa-
sha1</tp:SignatureAlgorithm>
                              <tp:CertificateRef tp:certId="N03"/>
                        </tp:NonRepudiation>
                        <tp:DigitalEnvelope>
                              <tp:Protocol tp:version="2.0">S/MIME</tp:Protocol>
                              <tp:EncryptionAlgorithm>DES-
CBC</tp:EncryptionAlgorithm>
                              <tp:CertificateRef tp:certId="N03"/>
                        </tp:DigitalEnvelope>
                  </tp:ebXMLBinding>
            </tp:DocExchange>
      </tp:PartyInfo>
      <tp:PartyInfo>
            <tp:PartyId tp:type="DUNS">987654321</tp:PartyId>
            <tp:PartyRef xlink:type="simple" xlink:href="http://contrived-
example.com/about.html"/>
            <tp:CollaborationRole tp:id="N30">
                  <tp:ProcessSpecification tp:version="1.0" tp:name="buySell"
xlink:type="simple" xlink:href="http://www.ebxml.org/processes/buySell.xml"/>
                  <tp:Role tp:name="seller" xlink:type="simple"
xlink:href="http://ebxml.org/processes/buySell.xml#seller"/>
                  <tp:CertificateRef tp:certId="N33"/>
                  <tp:ServiceBinding tp:channelId="N34" tp:packageId="N0402">
                        <tp:Service
```

```
tp:type="uriReference">uri:example.com/services/sellerService</tp:Service>
                        </tp:ServiceBinding>
                </tp:CollaborationRole>
                <tp:Certificate tp:certId="N33">
                        <ds:KeyInfo/>
                </tp:Certificate>
                <tp:DeliveryChannel tp:channelId="N34" tp:transportId="N35"
tp:docExchangeId="N36">
                        <tp:Characteristics tp:nonrepudiationOfOrigin="true"
tp:nonrepudiationOfReceipt="false" tp:secureTransport="true" tp:confidentiality="true"
tp:authenticated="true" tp:authorized="false"/>
                </tp:DeliveryChannel>
                <tp:Transport tp:transportId="N35">
                        <tp:SendingProtocol tp:version="1.1">HTTP</tp:SendingProtocol>
                        <tp:ReceivingProtocol tp:version="1.1">HTTP</tp:ReceivingProtocol>
                        <tp:Endpoint tp:uri="https://www.contrived-
example.com/servlets/ebxmlhandler" tp:type="allPurpose"/>
                        <tp:TransportSecurity>
                                <tp:Protocol tp:version="3.0">SSL</tp:Protocol>
                                <tp:CertificateRef tp:certId="N33"/>
                        </tp:TransportSecurity>
                </tp:Transport>
                <tp:DocExchange tp:docExchangeId="N36">
                        <tp:ebXMLBinding tp:version="0.98b">
                                <tp:ReliableMessaging tp:deliverySemantics="OnceAndOnlyOnce"
tp:idempotency="true" tp:messageOrderSemantics="Guaranteed">
                                        <tp:Retries>5</tp:Retries>
                                        <tp:RetryInterval>30</tp:RetryInterval>
                                        <tp:PersistDuration>P1D</tp:PersistDuration>
                                </tp:ReliableMessaging>
                                <tp:NonRepudiation>

        <tp:Protocol>http://www.w3.org/2000/09/xmldsig#</tp:Protocol>

        <tp:HashFunction>http://www.w3.org/2000/09/xmldsig#sha1</tp:HashFunction>

        <tp:SignatureAlgorithm>http://www.w3.org/2000/09/xmldsig#dsa-
sha1</tp:SignatureAlgorithm>
                                        <tp:CertificateRef tp:certId="N33"/>
                                </tp:NonRepudiation>
                                <tp:DigitalEnvelope>
                                        <tp:Protocol tp:version="2.0">S/MIME</tp:Protocol>
                                        <tp:EncryptionAlgorithm>DES-
CBC</tp:EncryptionAlgorithm>
                                        <tp:CertificateRef tp:certId="N33"/>
                                </tp:DigitalEnvelope>
                        </tp:ebXMLBinding>
                </tp:DocExchange>
        </tp:PartyInfo>
        <tp:Packaging tp:id="N0402">
                <tp:ProcessingCapabilities tp:parse="true" tp:generate="true"/>
                <tp:SimplePart tp:id="N40" tp:mimetype="text/xml">
                        <tp:NamespaceSupported
tp:location="http://ebxml.org/project_teams/transport/messageService.xsd"
tp:version="0.98b">http://www.ebxml.org/namespaces/messageService</tp:NamespaceSupport
ed>
                        <tp:NamespaceSupported
tp:location="http://ebxml.org/project_teams/transport/xmldsig-core-schema.xsd"
tp:version="1.0">http://www.w3.org/2000/09/xmldsig</tp:NamespaceSupported>
                </tp:SimplePart>
                <tp:SimplePart tp:id="N41" tp:mimetype="text/xml">
                        <tp:NamespaceSupported
tp:location="http://ebxml.org/processes/buysell.xsd"
tp:version="1.0">http://ebxml.org/processes/buysell.xsd</tp:NamespaceSupported>
```

```
            </tp:SimplePart>
            <tp:CompositeList>
                  <tp:Composite tp:id="N42" tp:mimetype="multipart/related"
tp:mimeparameters="type=text/xml;">
                        <tp:Constituent tp:idref="N40"/>
                        <tp:Constituent tp:idref="N41"/>
                  </tp:Composite>
            </tp:CompositeList>
      </tp:Packaging>
      <tp:Comment xml:lang="en-us">buy/sell agreement between example.com and
contrived-example.com</tp:Comment>
</tp:CollaborationProtocolAgreement>
```

付録 C　完全な CPP/CPA 定義（規範的）に対応する DTD

この DTD は、次の ASCII ファイルでも入手可能である。

http://ebxml.org/project_teams/trade_partner/cpp-cpa-v1_0.dtd

```
<?xml version="1.0" encoding="UTF-8"?>
<!--Generated by XML Authority-->
<!ELEMENT CollaborationProtocolAgreement (Status, Start, End, ConversationConstraints?,
PartyInfo+, Packaging, ds:Signature*, Comment*)>
<!ATTLIST CollaborationProtocolAgreement
       cpaid CDATA #IMPLIED
       version CDATA #IMPLIED
>
<!ELEMENT CollaborationProtocolProfile (PartyInfo+, Packaging, ds:Signature?,
Comment*)>
<!ATTLIST CollaborationProtocolProfile
       version CDATA #IMPLIED
>
<!ELEMENT ProcessSpecification (ds:Reference?)>
<!ATTLIST ProcessSpecification
       version CDATA #REQUIRED
       name CDATA #REQUIRED
       xlink:type CDATA #FIXED "simple"
       xlink:href CDATA #IMPLIED
>
<!ELEMENT Protocol (#PCDATA)>
<!ATTLIST Protocol
       version CDATA #IMPLIED
>
<!ELEMENT SendingProtocol (#PCDATA)>
<!ATTLIST SendingProtocol
       version CDATA #IMPLIED
>
<!ELEMENT ReceivingProtocol (#PCDATA)>
<!ATTLIST ReceivingProtocol
       version CDATA #IMPLIED
>
<!ELEMENT CollaborationRole (ProcessSpecification, Role, CertificateRef?,
ServiceBinding+)>
<!ATTLIST CollaborationRole
       id ID #IMPLIED
>
<!ELEMENT PartyInfo (PartyId+, PartyRef, CollaborationRole+, Certificate+,
DeliveryChannel+, Transport+, DocExchange+)>
<!ELEMENT PartyId (#PCDATA)>
<!ATTLIST PartyId
       type CDATA #IMPLIED
>
<!ELEMENT PartyRef EMPTY>
<!ATTLIST PartyRef
       xlink:type (simple) #IMPLIED
       xlink:href CDATA #IMPLIED
```

```
>
<!ELEMENT DeliveryChannel (Characteristics)>
<!ATTLIST DeliveryChannel
       channelId ID #REQUIRED
       transportId IDREF #REQUIRED
       docExchangeId IDREF #REQUIRED
>
<!ELEMENT Transport (SendingProtocol+, ReceivingProtocol, Endpoint+,
TransportSecurity?)>
<!ATTLIST Transport
       transportId ID #REQUIRED
>
<!ELEMENT Endpoint EMPTY>
<!ATTLIST Endpoint
       uri CDATA #REQUIRED
       type (login | request | response | error | allPurpose) "allPurpose"
>
<!ELEMENT Retries (#PCDATA)>
<!ELEMENT RetryInterval (#PCDATA)>
<!ELEMENT TransportSecurity (Protocol, CertificateRef?)>
<!ELEMENT Certificate (ds:KeyInfo)>
<!ATTLIST Certificate
       certId ID #REQUIRED
>
<!ELEMENT DocExchange (ebXMLBinding)>
<!ATTLIST DocExchange
       docExchangeId ID #REQUIRED
>
<!ELEMENT PersistDuration (#PCDATA)>
<!ATTLIST PersistDuration
       e-dtype NMTOKEN #FIXED "timeDuration"
>
<!ELEMENT ReliableMessaging (Retries, RetryInterval, PersistDuration)?>
<!ATTLIST ReliableMessaging
       deliverySemantics (OnceAndOnlyOnce | BestEffort) #REQUIRED
       messageOrderSemantics (Guaranteed | NotGuaranteed) "NotGuaranteed"
       idempotency CDATA #REQUIRED
>
<!ELEMENT NonRepudiation (Protocol, HashFunction, SignatureAlgorithm, CertificateRef)>
<!ELEMENT HashFunction (#PCDATA)>
<!ELEMENT EncryptionAlgorithm (#PCDATA)>
<!ELEMENT SignatureAlgorithm (#PCDATA)>
<!ELEMENT DigitalEnvelope (Protocol, EncryptionAlgorithm, CertificateRef)>
<!ELEMENT CertificateRef EMPTY>
<!ATTLIST CertificateRef
       certId IDREF #REQUIRED
>
<!ELEMENT ebXMLBinding (ReliableMessaging?, NonRepudiation?, DigitalEnvelope?,
NamespaceSupported*)>
<!ATTLIST ebXMLBinding
       version CDATA #REQUIRED
>
<!ELEMENT NamespaceSupported (#PCDATA)>
<!ATTLIST NamespaceSupported
```

```
      location CDATA #REQUIRED
      version CDATA #IMPLIED
>
<!ELEMENT Characteristics EMPTY>
<!ATTLIST Characteristics
      syncReplyMode (responseOnly | signalsAndResponse | signalsOnly | none) #IMPLIED
      nonrepudiationOfOrigin CDATA #IMPLIED
      nonrepudiationOfReceipt CDATA #IMPLIED
      secureTransport CDATA #IMPLIED
      confidentiality CDATA #IMPLIED
      authenticated CDATA #IMPLIED
      authorized CDATA #IMPLIED
>
<!ELEMENT ServiceBinding (Service, Override*)>
<!ATTLIST ServiceBinding
      channelId IDREF #REQUIRED
      packageId IDREF #REQUIRED
>
<!ELEMENT Service (#PCDATA)>
<!ATTLIST Service
      type CDATA #IMPLIED>

<!ELEMENT Status EMPTY>
<!ATTLIST Status
      value (agreed | signed | proposed) #REQUIRED
>
<!ELEMENT Start (#PCDATA)>
<!ELEMENT End (#PCDATA)>
<!ELEMENT Type (#PCDATA)>
<!ELEMENT ConversationConstraints EMPTY>
<!ATTLIST ConversationConstraints
      invocationLimit CDATA #IMPLIED
      concurrentConversations CDATA #IMPLIED
>
<!ELEMENT Override EMPTY>
<!ATTLIST Override
      action CDATA #REQUIRED
      channelId ID #REQUIRED
      packageId IDREF #REQUIRED
      xlink:href CDATA #IMPLIED
      xlink:type CDATA #FIXED "simple"
>
<!ELEMENT Role EMPTY>
<!ATTLIST Role
      name CDATA #REQUIRED
      xlink:type CDATA #FIXED "simple"
      xlink:href CDATA #IMPLIED
>
<!ELEMENT Constituent EMPTY>
<!ATTLIST Constituent
      idref CDATA #REQUIRED
>
<!ELEMENT ProcessingCapabilities EMPTY>
<!ATTLIST ProcessingCapabilities
```

```
       parse CDATA #REQUIRED
       generate CDATA #REQUIRED
>
<!ELEMENT SimplePart (NamespaceSupported*)>
<!ATTLIST SimplePart
       id ID #IMPLIED
       mimetype CDATA #REQUIRED
>
<!ELEMENT Encapsulation (Constituent)>
<!ATTLIST Encapsulation
       id ID #IMPLIED
       mimetype CDATA #REQUIRED
       mimeparameters CDATA #IMPLIED
>
<!ELEMENT Composite (Constituent+)>
<!ATTLIST Composite
       id ID #IMPLIED
       mimetype CDATA #REQUIRED
       mimeparameters CDATA #IMPLIED
>
<!ELEMENT CompositeList (Encapsulation | Composite)+>
<!ELEMENT Packaging (ProcessingCapabilities, SimplePart+, CompositeList?)>
<!ATTLIST Packaging
       id ID #REQUIRED
>
<!ELEMENT Comment (#PCDATA)>
<!ATTLIST Comment
       xml:lang CDATA #REQUIRED
>
<!ELEMENT ds:Signature ANY>
<!ELEMENT ds:Reference ANY>
<!ELEMENT ds:KeyInfo ANY>
```

付録 D　完全な CPP および CPA 定義に対応する XML スキーマ文書（規範的）

この XML スキーマ文書は、次の ASCII ファイルでも入手可能である。

http://ebxml.org/project_teams/trade_partner/cpp-cpa-v1_0.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<schema targetNamespace="http://www.ebxml.org/namespaces/tradePartner"
xmlns:xml="http://www.w3.org/XML/1998/namespace"
xmlns="http://www.w3.org/2000/10/XMLSchema"
xmlns:tns="http://www.ebxml.org/namespaces/tradePartner"
xmlns:xlink="http://www.w3.org/1999/xlink"
xmlns:xsi="http://www.w3.org/2000/10/XMLSchema-instance"
xmlns:ds="http://www.w3.org/2000/09/xmldsig#" elementFormDefault="qualified"
attributeFormDefault="unqualified" version="1.0">
       <import namespace="http://www.w3.org/1999/xlink"
schemaLocation="http://ebxml.org/project_teams/transport/xlink.xsd"/>
       <import namespace="http://www.w3.org/2000/09/xmldsig#"
schemaLocation="http://ebxml.org/project_teams/transport/xmldsig-core-schema.xsd"/>
       <import namespace="http://www.w3.org/XML/1998/namespace"
schemaLocation="http://ebxml.org/project_teams/transport/xml_lang.xsd"/>
       <attributeGroup name="pkg.grp">
              <attribute ref="tns:id"/>
              <attribute name="mimetype" type="tns:non-empty-string" use="required"/>
              <attribute name="mimeparameters" type="tns:non-empty-string"/>
       </attributeGroup>
       <attributeGroup name="xlink.grp">
              <attribute ref="xlink:type"/>
              <attribute ref="xlink:href"/>
       </attributeGroup>
       <element name="CollaborationProtocolAgreement">
              <complexType>
                     <sequence>
                            <element ref="tns:Status"/>
                            <element ref="tns:Start"/>
                            <element ref="tns:End"/>
                            <element ref="tns:ConversationConstraints" minOccurs="0"/>
                            <element ref="tns:PartyInfo" maxOccurs="unbounded"/>
                            <element ref="tns:Packaging"/>
                            <element ref="ds:Signature" minOccurs="0"
maxOccurs="unbounded"/>
                            <element ref="tns:Comment" minOccurs="0"
maxOccurs="unbounded"/>
                     </sequence>
                     <attribute name="cpaid" type="tns:non-empty-string"/>
                     <attribute ref="tns:version"/>
                     <anyAttribute namespace="##targetNamespace
http://www.w3.org/2000/10/XMLSchema-instance" processContents="lax"/>
              </complexType>
       </element>
       <element name="CollaborationProtocolProfile">
              <complexType>
```

```
                    <sequence>
                           <element ref="tns:PartyInfo" maxOccurs="unbounded"/>
                           <element ref="tns:Packaging"/>
                           <element ref="ds:Signature" minOccurs="0"/>
                           <element ref="tns:Comment" minOccurs="0"
maxOccurs="unbounded"/>
                    </sequence>
                    <attribute ref="tns:version"/>
                    <anyAttribute namespace="##targetNamespace
http://www.w3.org/2000/10/XMLSchema-instance" processContents="lax"/>
             </complexType>
      </element>
      <element name="ProcessSpecification">
             <complexType>
                    <sequence>
                           <element ref="ds:Reference" minOccurs="0"/>
                    </sequence>
                    <attribute ref="tns:version"/>
                    <attribute name="name" type="tns:non-empty-string"
use="required"/>
                    <attributeGroup ref="tns:xlink.grp"/>
             </complexType>
      </element>
      <element name="Service" type="tns:service.type"/>
      <element name="Protocol" type="tns:protocol.type"/>
      <element name="SendingProtocol" type="tns:protocol.type"/>
      <element name="ReceivingProtocol" type="tns:protocol.type"/>
      <element name="CollaborationRole">
             <complexType>
                    <sequence>
                           <element ref="tns:ProcessSpecification"/>
                           <element ref="tns:Role"/>
                           <element ref="tns:CertificateRef" minOccurs="0"/>
                           <element ref="tns:ServiceBinding" maxOccurs="unbounded"/>
                    </sequence>
                    <attribute ref="tns:id"/>
             </complexType>
      </element>
      <element name="PartyInfo">
             <complexType>
                    <sequence>
                           <element ref="tns:PartyId" maxOccurs="unbounded"/>
                           <element ref="tns:PartyRef"/>
                           <element ref="tns:CollaborationRole" maxOccurs="unbounded"/>
                           <element ref="tns:Certificate" maxOccurs="unbounded"/>
                           <element ref="tns:DeliveryChannel" maxOccurs="unbounded"/>
                           <element ref="tns:Transport" maxOccurs="unbounded"/>
                           <element ref="tns:DocExchange" maxOccurs="unbounded"/>
                    </sequence>
             </complexType>
      </element>
      <element name="PartyId">
             <complexType>
                    <simpleContent>
```

```
                        <extension base="tns:non-empty-string">
                              <attribute name="type" type="tns:non-empty-string"/>
                        </extension>
                  </simpleContent>
            </complexType>
      </element>
      <element name="PartyRef">
            <complexType>
                  <attributeGroup ref="tns:xlink.grp"/>
                  <attribute name="type" type="tns:non-empty-string"/>
            </complexType>
      </element>
      <element name="DeliveryChannel">
            <complexType>
                  <sequence>
                        <element ref="tns:Characteristics"/>
                  </sequence>
                  <attribute name="channelId" type="ID" use="required"/>
                  <attribute name="transportId" type="IDREF" use="required"/>
                  <attribute name="docExchangeId" type="IDREF" use="required"/>
            </complexType>
      </element>
      <element name="Transport">
            <complexType>
                  <sequence>
                        <element ref="tns:SendingProtocol" maxOccurs="unbounded"/>
                        <element ref="tns:ReceivingProtocol"/>
                        <element ref="tns:Endpoint" maxOccurs="unbounded"/>
                        <element ref="tns:TransportSecurity" minOccurs="0"/>
                  </sequence>
                  <attribute name="transportId" type="ID" use="required"/>
            </complexType>
      </element>
      <element name="Endpoint">
            <complexType>
                  <attribute name="uri" type="uriReference" use="required"/>
                  <attribute name="type" type="tns:endpointType.type" use="default"
value="allPurpose"/>
            </complexType>
      </element>
      <element name="Retries" type="string"/>
      <element name="RetryInterval" type="string"/>
      <element name="TransportSecurity">
            <complexType>
                  <sequence>
                        <element ref="tns:Protocol"/>
                        <element ref="tns:CertificateRef" minOccurs="0"/>
                  </sequence>
            </complexType>
      </element>
      <element name="Certificate">
            <complexType>
                  <sequence>
                        <element ref="ds:KeyInfo"/>
```

```
                </sequence>
                <attribute name="certId" type="ID" use="required"/>
            </complexType>
        </element>
        <element name="DocExchange">
            <complexType>
                <sequence>
                    <element ref="tns:ebXMLBinding"/>
                </sequence>
                <attribute name="docExchangeId" type="ID" use="required"/>
            </complexType>
        </element>
        <element name="ReliableMessaging">
            <complexType>
                <sequence minOccurs="0">
                    <element ref="tns:Retries"/>
                    <element ref="tns:RetryInterval"/>
                    <element name="PersistDuration" type="timeDuration"/>
                </sequence>
                <attribute name="deliverySemantics" type="tns:ds.type"
use="required"/>
                <attribute name="idempotency" type="boolean" use="required"/>
                <attribute name="messageOrderSemantics" type="tns:mos.type"
use="optional" value="NotGuaranteed"/>
            </complexType>
            <!-- <element name="PersistDuration" type="duration"/> -->
        </element>
        <element name="NonRepudiation">
            <complexType>
                <sequence>
                    <element ref="tns:Protocol"/>
                    <element ref="tns:HashFunction"/>
                    <element ref="tns:SignatureAlgorithm"/>
                    <element ref="tns:CertificateRef"/>
                </sequence>
            </complexType>
        </element>
        <element name="HashFunction" type="string"/>
        <element name="EncryptionAlgorithm" type="string"/>
        <element name="SignatureAlgorithm" type="string"/>
        <element name="DigitalEnvelope">
            <complexType>
                <sequence>
                    <element ref="tns:Protocol"/>
                    <element ref="tns:EncryptionAlgorithm"/>
                    <element ref="tns:CertificateRef"/>
                </sequence>
            </complexType>
        </element>
        <element name="CertificateRef">
            <complexType>
                <attribute name="certId" type="IDREF" use="required"/>
            </complexType>
        </element>
```

```
	<element name="ebXMLBinding">
		<complexType>
			<sequence>
				<element ref="tns:ReliableMessaging" minOccurs="0"/>
				<element ref="tns:NonRepudiation" minOccurs="0"/>
				<element ref="tns:DigitalEnvelope" minOccurs="0"/>
				<element ref="tns:NamespaceSupported" minOccurs="0"
maxOccurs="unbounded"/>
			</sequence>
			<attribute ref="tns:version"/>
		</complexType>
	</element>
	<element name="NamespaceSupported">
		<complexType>
			<simpleContent>
				<extension base="uriReference">
					<attribute name="location" type="uriReference"
use="required"/>
					<attribute ref="tns:version"/>
				</extension>
			</simpleContent>
		</complexType>
	</element>
	<element name="Characteristics">
		<complexType>
			<attribute ref="tns:syncReplyMode"/>
			<attribute name="nonrepudiationOfOrigin" type="boolean"/>
			<attribute name="nonrepudiationOfReceipt" type="boolean"/>
			<attribute name="secureTransport" type="boolean"/>
			<attribute name="confidentiality" type="boolean"/>
			<attribute name="authenticated" type="boolean"/>
			<attribute name="authorized" type="boolean"/>
		</complexType>
	</element>
	<element name="ServiceBinding">
		<complexType>
			<sequence>
				<element ref="tns:Service"/>
				<element ref="tns:Override" minOccurs="0"
maxOccurs="unbounded"/>
			</sequence>
			<attribute name="channelId" type="IDREF" use="required"/>
			<attribute name="packageId" type="IDREF" use="required"/>
		</complexType>
		<unique name="action.const">
			<selector xpath=".//Override"/>
			<field xpath="@action"/>
		</unique>
	</element>
	<element name="Status">
		<complexType>
			<attribute name="value" type="tns:statusValue.type"
use="required"/>
		</complexType>
```

```
	</element>
	<element name="Start" type="timeInstant"/>
	<element name="End" type="timeInstant"/>
	<!--
	<element name="Start" type="dateTime"/>
	<element name="End" type="dateTime"/>
	-->
	<element name="Type" type="string"/>
	<element name="ConversationConstraints">
		<complexType>
			<attribute name="invocationLimit" type="int"/>
			<attribute name="concurrentConversations" type="int"/>
		</complexType>
	</element>
	<element name="Override">
		<complexType>
			<attribute name="action" type="tns:non-empty-string"
use="required"/>
			<attribute name="channelId" type="ID" use="required"/>
			<attribute name="packageId" type="IDREF" use="required"/>
			<attributeGroup ref="tns:xlink.grp"/>
		</complexType>
	</element>
	<element name="Role">
		<complexType>
			<attribute name="name" type="tns:non-empty-string"
use="required"/>
			<attributeGroup ref="tns:xlink.grp"/>
		</complexType>
	</element>
	<element name="Constituent">
		<complexType>
			<attribute ref="tns:idref"/>
		</complexType>
	</element>
	<element name="Packaging">
		<complexType>
			<sequence>
				<element name="ProcessingCapabilities">
					<complexType>
						<attribute name="parse" type="boolean"
use="required"/>
						<attribute name="generate" type="boolean"
use="required"/>
					</complexType>
				</element>
				<element name="SimplePart" maxOccurs="unbounded">
					<complexType>
						<sequence>
							<element ref="tns:NamespaceSupported"
minOccurs="0" maxOccurs="unbounded"/>
						</sequence>
						<attributeGroup ref="tns:pkg.grp"/>
					</complexType>
```

```
                        </element>
                        <element name="CompositeList" minOccurs="0">
                              <complexType>
                                    <choice maxOccurs="unbounded">
                                          <element name="Encapsulation">
                                                <complexType>
                                                      <sequence>
                                                            <element
ref="tns:Constituent"/>
                                                      </sequence>
                                                      <attributeGroup
ref="tns:pkg.grp"/>
                                                </complexType>
                                          </element>
                                          <element name="Composite">
                                                <complexType>
                                                      <sequence>
                                                            <element
ref="tns:Constituent" maxOccurs="unbounded"/>
                                                      </sequence>
                                                      <attributeGroup
ref="tns:pkg.grp"/>
                                                </complexType>
                                          </element>
                                    </choice>
                              </complexType>
                        </element>
                  </sequence>
                  <attribute ref="tns:id"/>
            </complexType>
      </element>
      <element name="Comment">
            <complexType>
                  <simpleContent>
                        <extension base="tns:non-empty-string">
                              <attribute ref="xml:lang"/>
                        </extension>
                  </simpleContent>
            </complexType>
      </element>
      <!-- COMMON -->
      <simpleType name="ds.type">
            <restriction base="NMTOKEN">
                  <enumeration value="OnceAndOnlyOnce"/>
                  <enumeration value="BestEffort"/>
            </restriction>
      </simpleType>
      <simpleType name="mos.type">
            <restriction base="NMTOKEN">
                  <enumeration value="Guaranteed"/>
                  <enumeration value="NotGuaranteed"/>
            </restriction>
      </simpleType>
      <simpleType name="statusValue.type">
```

```
		<restriction base="NMTOKEN">
			<enumeration value="agreed"/>
			<enumeration value="signed"/>
			<enumeration value="proposed"/>
		</restriction>
	</simpleType>
	<simpleType name="endpointType.type">
		<restriction base="NMTOKEN">
			<enumeration value="login"/>
			<enumeration value="request"/>
			<enumeration value="response"/>
			<enumeration value="error"/>
			<enumeration value="allPurpose"/>
		</restriction>
	</simpleType>
	<simpleType name="non-empty-string">
		<restriction base="string">
			<minLength value="1"/>
		</restriction>
	</simpleType>
	<simpleType name="syncReplyMode.type">
		<restriction base="NMTOKEN">
			<enumeration value="responseOnly"/>
			<enumeration value="signalsAndResponse"/>
			<enumeration value="signalsOnly"/>
			<enumeration value="none"/>
		</restriction>
	</simpleType>
	<complexType name="service.type">
		<simpleContent>
			<extension base="tns:non-empty-string">
				<attribute name="type" type="tns:non-empty-string"/>
			</extension>
		</simpleContent>
	</complexType>
	<complexType name="protocol.type">
		<simpleContent>
			<extension base="tns:non-empty-string">
				<attribute ref="tns:version"/>
			</extension>
		</simpleContent>
	</complexType>
	<attribute name="idref" type="IDREF" form="unqualified"/>
	<attribute name="id" type="ID" form="unqualified"/>
	<attribute name="version" type="tns:non-empty-string"/>
	<attribute name="syncReplyMode" type="tns:syncReplyMode.type"/>
</schema>
```

付録 E　CPP および CPA の情報の形式（規範的）

本節では、[XML]の仕様によって定義されていない形式、および特定の要素の記述では定義されていない形式についての情報を定義する。

文字列の形式

Protocol 要素および Version 要素

Protocol、***Version***、および類似の要素の値は柔軟である。一般に、その選択によって DTD およびスキーマ（したがって仕様）の変更が必要にならない限り、*CPA* の両方の当事者にとってそのサポートソフトウェアが入手可能なプロトコルおよびバージョンを選択してもよい。

> 注意: 実装は、これらの要素の値をサポートするために、プラグインなどを使用したものであってもよい。

英数字文字列

英数字文字列は、本節では詳細には定義されていないが、個々の要素の記述に別に示されていなければ次の規則に従う。

- 別に記述がなければ、要素の値は大文字と小文字を区別しない
- ファイルまたはディレクトリの名前を表現するための文字列は、UNIX と Windows の両方で許容されることを保証するために、大文字と小文字を区別する。

数字文字列

数字文字列は、32 ビットの 2 進数の範囲（-2,147,483,648～+2,417,483,647）の符号付きまたは符号なしの整数である。負の数は、特定の要素で許されていてもいなくてもよい。

付録 F　2 つの CPP から CPA を合成する（非規範的）

概要および制限事項

この付録では、*CPP* から *CPA* を作成する作業について議論する。現在のところ、*CPA* を作成する詳細な手順は、実装者に委ねられている。したがって、*CPA* 作成のアルゴリズムについて、規範的な仕様は提供されていない。この最初の節では、*CPA* 作成作業の背景についてある程度示す。

CPA 作成アルゴリズムを提示するのではなく、*CPA* 作成を構成する作業について記述する基本的な理由は 3 つある。

1. *CPA* 作成手順に対する入力となる情報はさまざまである。
2. *CPA* 作成に対する手法としては、少なくとも 2 つの異なる手法が存在している。1 つの手法は、*CPA* テンプレートから CPA の基礎を作成するという状況を捉えた手法である。もう 1 つの手法は、*CPP* から合成する手法である。
3. 特定の 2 つの *CPP* に対して特定の *CPA* を出力する条件には、さまざまなレベルおよび程度の相互運用性が関わり得る。言い換えると、要件のすべてのレベルおよびその他のすべての制約を満たす最適な解決策は存在しない。*当事者*は、「十分良い」実装のための要件が十分満たされている *CPA* を望んでもよい。また、何が十分良い実装であるかを見極めるために、ユーザの入力を求めてもよい。一方、優先順位を表現するためにユーザが選択するシステム構成も多様であってもよい。実際のシステムでは、許容範囲にある*ビジネス*の手段を見つけるために、セキュリティ、メッセージングの信頼性、信号および承認のレベルなどの問題について妥協してもよい。

これらの理由のそれぞれについて、以降の節で詳細に述べる。

入力の多様性

ユーザの優先順位は、*CPA* の作成における入力の多様性をもたらす原因の 1 つである。本節では、個々の*当事者*が、当事者になる可能性のある相手に対して自社の *CPP* を入手可能な状態にしていると仮定する。通常は、一方の当事者に、意中の当事者たちと共に実装したいと考える特定の*取引コラボレーション*（*プロセス仕様*文書で定義されている）がある。したがって、入力される情報には通常、*CPP* だけでなく、意中の*取引コラボレーション*に関するユーザの優先順位が含まれる。

CPA 作成ツールは、作成される *CPA* に反映してもよいローカルのユーザ情報で、*CPP* では宣伝されていない情報にもアクセスしてもよい。ユーザは、システム能力のうち、問題発生の余地のない能力のみを宣伝するよう選択してもよい。たとえば、ユーザは、

FTP が使用できても、HTTP だけを宣伝して FTP の宣伝を省略するかもしれない。FTP を省略する理由は、ユーザのアカウント、ディレクトリ、および FTP セッションのパスワードの漏出を恐れているからかもしれない。FTP 機能を宣伝していない場合でも、設定ソフトウェアが自社の FTP 機能についての暗黙の知識を利用して、FTP 機能だけが利用できる当事者たちとの間の *CPA* を作成してもよい。言い換えると、この場合、問題を避けるという方針よりもビジネス上の利益が優先するかもしれない。両者の暗黙の知識および詳細な優先順位についての情報によって、*CPA* 作成のための入力の多様性が生じる。

さまざまな手法

CPA テンプレートから *CPA* を作成する典型的な理由は、一方の*当事者*の能力が限定されておりそれが既に暗に知られていることである。たとえば、ブラウザのフォーム機能を実装において使用するために、*CPA* テンプレートが Web ブラウザに暗に提示される場合、テンプレートの作成者は相手が適切な Web 機能を利用できる（またはダウンロードできる）ことを前提とする。したがって、実行する必要があるのは、***PartyRef***、***Certificate***、および類似の項目を *CPA* テンプレートの代替物として提供することだけである。*CPA* テンプレートには、さまざまなレベルにおける*当事者*双方のあらゆるの能力についての情報が既に付与されており、*当事者*が値を埋められるようになっているプレースホルダーも付与されることになる。必要な情報を収集して *CPA* を生成するためには、単純なフォームが適当であるかもしれない。

多様な出力に「十分」な機能という方針

CPA は、*取引*/*コラボレーション*に必要なすべての技術レベルに関して合意に達した、完全に相互運用可能な設定をサポートできる。その際、能力の合致は、すべての関連技術レベルにおいて実現される。

しかし、*取引*/*コラボレーション*の一部の側面において合致しない *CPA* であっても、相互運用可能なシステムを構成することも可能である。パッケージング、セキュリティ、信号伝達、メッセージングの信頼性などの分野でギャップが存在してもよい。その場合でも、システムが*取引*/*データ*を転送すること、および例外を処理する特別な方法を採用することは可能である。そのような状況では、*CPA* は、方針を反映してもよいし、システム構成の欠点を無視することを認可するようにユーザに要請してもよい。システムは、*取引*/*コラボレーション*内で、受取の否認不可という推奨される機能をサポートするための応答が送信できないかもしれないが、*ビジネス*としては許容できるかもしれない。また、システムは、「application/vnd.eb+xml」タイプの値に対して「multipart/related」の処理をサポートするために必要なすべての処理を扱えないかもしれないが、「multipart/mixed」に従ったマルチパートは扱えるので、*取引*/*コラボレーション*の実行が許されるかもしれない。データの転送能力および*取引*/*コラボレーション*に関連するデ

ータの提供能力が欠如していても、*ビジネス*上の利益を優先する場合、一時的にも永久的にも妥協できないような機能は、実際には少ない。「部分的な相互運用性」がしばらく続くことが期待されるため、相互運用性のために十分であるかどうかを見極めるためのアルゴリズムを用意する。

要約すると、上記の点から、*CPP* から *CPA* を作成するための単純なアルゴリズムを模索することが現時点では最善の方法である。機能の特徴付けと交換がより精錬され、*CPA* の作成と交渉を特徴付ける点でより進歩していくことが期待される。

上記の多様性をすべて包含するような *CPA* 作成のための単純なアルゴリズムを提案するのは時期尚早であるが、現在でも、*CPP* のマッチング機能に関する基本的な作業を列挙できる。この情報は、*取引/コラボレーション*のシステム構成に役立つ、部分的に自動化され部分的に対話的なソフトウェアシステムを設計しているソフトウェア実装者が、十分なレベルの相互運用性に達するための助けになっているかもしれない。各構成作業を特徴付けるコンテキストを理解するには、*CPP* および *CPA* の全体像を思い起こす必要がある。

CPA 作成の構成作業

技術的に言えば、*CPA* によって提供されるのは、*取引/コラボレーション*仕様（*プロセス仕様*文書に定義されている）と、それらの仕様を実装するために用いられるサービスおよびプロトコルの間の「バインディング」である。実装は、いくつかのレベルで実行され、それぞれのレベルにおけるさまざまなサービスを扱う。*取引/コラボレーション*と現在実現中のサービスおよびプロトコルの間の完全に相互運用可能なバインディングに達した *CPA* は、相互運用可能なアプリケーション間の統合に到達したものと見なすことができる。また、この目標に達していない *CPA* でも、コラボレーションの*当事者*にとって役に立ち許容できるものとなる場合もある。確かに、条件に合致するデータ転送能力が得られない場合、*CPA* は、相互運用可能な企業間統合を十分には提供できない。同様に、部分的な *CPA* の場合は、十分満足のいくアプリケーション間の統合が実現する前に、膨大な規模のシステム作業が必要になる。しかしそれでも、部分的な統合で、コラボレーションを実現し、自動化レベルの向上による利益を享受するのに十分である場合もある。

実際には、*CPA* 作成プロセスの結果、*CPA* の失敗をたくさん生成してしまう場合、ユーザとのギャップの一覧によりユーザとの話し合いが必要になる場合、または*取引/当事者*たちにとって「十分良い」部分的な相互運用性を実装した *CPA* が完成する場合がある。マッチング機能と相互運用性の両方を程度の問題として扱う必要がある場合もあるので、個々の構成作業において、さまざまなレベルのさまざまなサービスについて、能力の合致を探ることになる。以降の節では、これらの構成作業の多くについて特徴付ける。

CPP から CPA を作成する: 作業の列挙

議論を簡単にするために、以降の節では、ソフトウェアエージェントが直面する作業に焦点を合わせることにする。

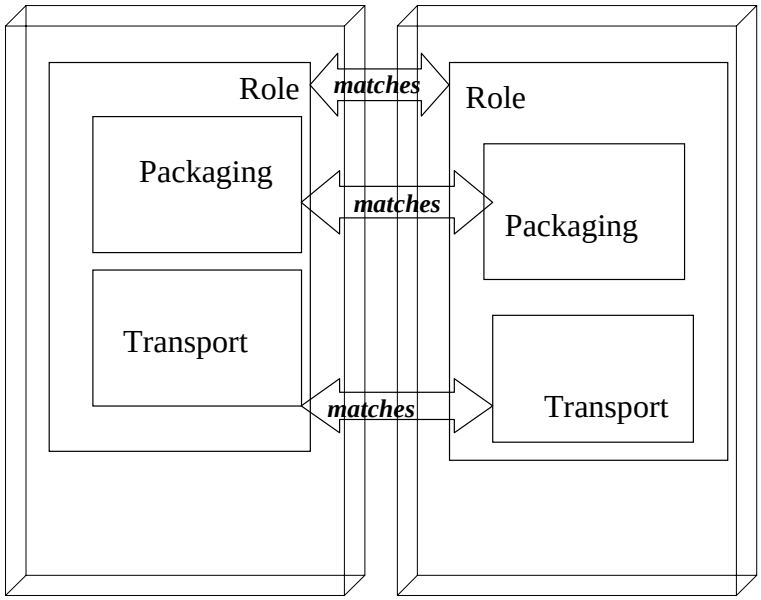
1. 意中の当事者が見つけ、当事者の *CPP* を取得する。
2. 意中の当事者たちの間の*取引/コラボレーション*を選択する。
3. ソフトウェアエージェントが*取引/コラボレーション*において果たす特定の役割が決まっている。
4. *CPP* で宣伝される能力が分かっている。

ここで、エージェントが、自分自身が供給者の役割を果たし、現在の顧客の 1 つを相手として選択し、その相手が補完的な役割を果たす購買注文*取引/コラボレーション*を開始したいと考えていると仮定する。また、簡便のため、能力についての情報は、我々のエージェントの *CPP* および意中の当事者の *CPP* で利用可能なものに限る。

一般に、構成作業は、さまざまなレベルのプロトコルスタックにおいて、これらのさまざまなレベルで供給されるサービスに関して、我々の能力と意中の当事者の能力の間の「合致」を見つけることから成る。

図 6 に、2 つの *CPP* から *CPA* を作成するための基本的な作業（役割の合致、パッケージングの合致、および転送の合致）を示す。

Figure 6: Basic Tasks in Forming a CPA



考慮すべき最初の作業は、意中の当事者たちと我々自身が補完的な役割能力を持っているかどうかを見極めるという、最も基本的な作業である。

役割の合致

我々のエージェントは、*取引/コラボレーション*において、既に役割を選択している。次に、当事者たちの *CPP* に付与されている ***Role*** 要素のチェックを開始する。検査する最初の要素は、***PartyInfo*** 要素である。***PartyInfo*** 要素には、***CollaborationRole*** という要素の階層を含んでいる。このセットを検索して、選択した*取引/コラボレーション*内のエージェントの役割を補完する役割を発見する。単純なバイナリコラボレーションの場合、通常は、意中の当事者たちの ***CollaborationRole*** セットに、われわれが実装しようとしている ***ProcessSpecification*** 要素が含まれており、その役割が我々の役割と同一でないことを確かめれば十分である。より一般的なコラボレーションの場合は、プロセス内で利用可能な役割のリストを得て、個々の当事者たちを追跡し、選択された役割が*プロセス仕様*文書内で指定されている役割のインスタンスを生成するものであることを確認する必要がある。2 つの以上の役割を含むコラボレーションについては、ここでは議論しない。

転送の合致

この時点で、候補となる ***CollaborationRole*** 要素のリストと ***ProcessSpecification*** 要素（購買注文; 意中の当事者たちが買い手の役割を果たす）が得られた。簡単のため、関連するそれぞれの *CPP* の中で、1 つの ***CollaborationRole*** 要素だけがこれらの条件を満たすと仮定し、リスト内での繰り返しについては議論しないこととする。（こうした考慮の上で、繰り返しが可能な場合も、1 つの要素だけが存在すると仮定して議論を進めることにする。）

「転送の合致」という言葉は、まず、意中の当事者たちの ***SendingProtocol*** 能力が我々の側の ***ReceivingProtocol*** 能力と合致するということを意味する。*CPP* の DTD またはスキーマを丹念に読むと、***ServiceBinding*** 要素が、双方の側の ***channelId*** 属性付きの ***CollaborationRole*** 要素からの関連情報を得るための入口になることが分かる。***channelId*** 属性の値を使用すれば、個々の *CPP* 内の ***DeliveryChannels*** を見つけることができる。***DeliveryChannel*** には、***transportId*** 属性が付与されている。***transportId*** 属性によって、関連する ***Transport*** の階層を見つけることができる。

たとえば、意中の買い手が、次のような ***Tranport*** エントリを持っているとする。

```
<Transport transportId = "buyerid001">
      <SendingProtocol>HTTP</SendingProtocol>
      <ReceivingProtocol>
      HTTP
```

```
      </ReceivingProtocol>
      <Endpoint uri = "https://www.buyername.com/po-response"
                  type = "allPurpose"/>
      <TransportSecurity>
            <Protocol version = "1.0">TLS</Protocol>
            <CertificateRef certId =  certid001">BuyerName</CertificateRef>
      </TransportSecurity>
</Transport>
```

さらに、売り手が、次のような ***Transport*** エントリを持っているとする。

```
<Transport transportId = "sellid001">
      <SendingProtocol>HTTP</SendingProtocol>
      <ReceivingProtocol>
      HTTP
      </ReceivingProtocol>
      <Endpoint uri = "https://www.sellername.com/pos_here"
                  type = "allPurpose"/>
      <TransportSecurity>
            <Protocol version = "1.0">TLS</Protocol>
            <CertificateRef certId ="certid002">Sellername</CertificateRef>
      </TransportSecurity>
</Transport>
```

依頼のための転送の合致には、開始者の役割（買い手）に、我々の ***ReceivingProtocols*** の 1 つと合致する ***SendingProtocol*** が付与されていることを確認することを含む。この例では「HTTP」が合致している。応答のための転送の合致には、応答者の役割（売り手）に、買い手の ***ReceivingProtocols*** の 1 つと合致する ***SendingProtocol*** が付与されていることを確かめることが含まれる。やはり、この例では「HTTP」が合致している。合致が存在している場合、転送レベルで相互運用可能なソリューションを見つけたことになる。そうではない場合は、利用可能な *CPA* がないことになる。優先順位の高いギャップが発見されたため、例外処理手続きにより修復する必要がある。

転送セキュリティの合致

上記のような転送セキュリティの合致は、バージョンおよびプロトコル値の一致に基づく。ここでは、ソフトウェアから、ある程度の知識の提供が可能である。すなわち、一方が SSL-3、他方が TLS-1 を使用している場合、TLS から SSL へのフォールバックによってセキュリティが可能であることを推測することができる。

文書パッケージングの合致

最も複雑な合致の判定の 1 つは、恐らく、文書パッケージング能力の合致の判断である。セキュリティと他の MIME 処理能力の両方が組み合わさるため、完全な相互運用性が得られるかどうかの評価が複雑になる。

この作業の実行に必要な情報へのアクセスは、***ServiceBinding*** 要素を介して行われる。ここでも、双方の側の ***ServiceBinding*** 要素は 1 つずつであると仮定する。ただし、それぞれの役割に対して、2 つの ***Packaging*** 要素が利用可能であると最初は仮定する。合致作業については非常にさまざまな方法が考えられる。十分良い合致が存在しているかどうかを評価する際には、いくつかの合致作業の方法を実行してもよい。

上記の購買注文の例では、パッケージングは、本体部分、XML インスタンス（ヘッダと搬送内容）、およびデータソースからの*メッセージ*の組み立てに使用されるセキュリティカプセル化の組み合せであった。依頼および応答の両方にパッケージングが付与される。最も完全なパッケージングの仕様（常に必要なわけではないかもしれないは、次の条件から成る。

1. 購買注文送信のために生成できるパッケージングおよび購買注文の応答*メッセージ*生成のために構文解析できるパッケージングを、買い手が宣言している。
2. 購買注文への応答のために生成できるパッケージングおよび受信した購買注文のために構文解析できるパッケージングを、売り手が宣言している。

構造的な比較による合致には、買い手が構文解析できる購買注文と、売り手によって生成された購買注文のパッケージングの詳細な比較が含まれる。この比較においては、まず対応する階層中の ***SimplePart*** 要素の MIME タイプが合致していることが確認され、続いて ***CompositeList*** の MIME タイプおよび合成順序が合致していることがチェックされる。

たとえば、*CPP* がパッケージング階層を含み適切な ***ServiceBindings*** の下に配置している場合、構造的な比較による直接的な合致が存在することになる。

```
<Packaging id="I1001">
      <ProcessingCapabilities parse = "true" generate = "true"/>
      <SimplePart id = "P1" mimetype = "text/xml"/>
         <NamespaceSupported location
               = "http://schemas.xmlsoap.org/soap/envelope/" version = "1.1">
              http://schemas.xmlsoap.org/soap/envelope
         </NamespaceSupported>
         <NamespaceSupported location =
              "http://www.ebxml.org/namespaces/messageHeader"
              version = "1.0">
              http://www.ebxml.org/namespaces/messageHeader
         </NamespaceSupported>         <NamespaceSupported location =
                 "http://www.w3.org/2000/09/xmldsig#"
                version = "1.0">
                http://www.w3.org/2000/09/xmldsig#
         </NamespaceSupported>
      <SimplePart id = "P2" mimetype = "application/xml"/>
      <CompositeList>
            <Composite mimetype = "multipart/related" id = "P3"
                  mimeparameters = "type=text/xml">
                  <Constituent idref = "P1"/>
                  <Constituent idref = "P2"/>
```

```
            </Composite>
      </CompositeList>
</Packaging>
<Packaging id="I2001">
      <ProcessingCapabilities parse = "true" generate = "true"/>
      <SimplePart id = "P11" mimetype = "text/xml"/>
      <SimplePart id = "P12" mimetype = "application/xml"/>
      <CompositeList>
            <Composite mimetype = "multipart/related" id = "P13"
                  mimeparameters = "type=text/xml">
                  <Constituent idref = "P11"/>
                  <Constituent idref = "P12"/>
            </Composite>
      </CompositeList>
</Packaging>
```

ただし、時間の経過と共に、依頼元と応答者の役割に関して、個々の ***ServiceBinding*** 内で*生成*されるパッケージングの種類を宣言できるようになることが期待されている。この単純化においては、双方の側が、正しく処理できる MIME タイプ、正しく処理できるカプセル化、および正しく処理できる合成モードについての知識を持っていることが仮定されている。内部的な能力のリストに沿ってパッケージング仕様をスキャンすることにより、他方の側で生成されたパッケージングスキームが、こうした条件の下で処理および受け入れ可能なものであるかどうかが見極められる。他方の側によって生成されたパッケージングスタイルを知ることによって、ソフトウェアエージェントは、受信*メッセージ*の MIME タイプおよびパッケージングスタイルだけを用いて、パッケージングスキームを提案できる。このように提案されたパッケージングスキームは、*CPA* の案に含める際に、他方の側が許容する可能性が高くなる。時間の経過と共に、提案および交渉の規約が確立されていくので、パッケージング能力の合致の見極めに使用される方法は、構造的な比較から、より簡潔な表現を用いた方法に変化していくであろう。たとえば、構文解析の能力は、構文解析および意味処理が可能なパッケージングと内容ラベル付けのスキームを受け付ける、コンパクトな文法記述を用いて取得されるようになるかもしれない。

文書レベルセキュリティの合致

文書レベルセキュリティの合致作業は、パッケージング合致作業のサブ作業であるが、[S/MIME]、OpenPGP [RFC2015]、および XMLDsig [XMLDSIG]に示されている、3 つの主な文書レベルセキュリティの手法に関連して、いくつかの具体的な内容について議論することは有用である。

XMLDsig の合致は、ebXML *メッセージ取扱サービス*[ebMS]のパッケージングを使用している場合、文書の合致から推論して得られる。ただし、この合致には、確認する必要がある他の情報源がある。たとえば、***SimplePart*** 要素には、***NamespaceSupported*** 要素を付与できる。また、XMLDsig 機能が存在する必要がある。同様に、この合致に対する詳細なチェックを行うことにより、ebXMLBinding 要素の下の ***NonRepudiation*** 要素および類似の要素内の情報が検査され、ハッシュ関数およびアルゴリズム内の互換性が確認

される。

文書レベルセキュリティに対するいくつかの極めて異なった手法が存在するという状況、および特定の*当事者*が複数の形式のセキュリティを利用することは現在のところ稀であるという状況は、セキュリティフレームワークの合致において基本的な障害が発生する可能性を示唆している。したがって、すべてのレベルで完全に相互運用性をサポートするような能力の合致はあり得ないかもしれない。しばらくの間は、文書レベルセキュリティの合致には、双方の側が同じセキュリティの複合タイプ（S/MIME による multipart/signed など）を処理できることが必要である。

しかし、転送およびトランスポート層のセキュリティレベルでは合致しているが、一方の側が PGP 署名を使用し他方の側が S/MIME を使用しているため、双方の側で文書セキュリティ層の障害が発生している場合、*CPA* を提案することはできないのであろうか。必ずしもそうではない。

S/MIME と OpenPGP の両方が、「multipart/signed」複合タイプにおけるパッケージングを許している。そのような場合、データを抽出した結果、否認不可に関しては失敗してしまうような部分的な実装が得られる場合があってもよい。互いに他方の側の署名をチェックすることはできないが、*取引/データ*について、機密書類の伝達および転送レベルの認証を行うことは可能である。最終的に、これらの例外的な状況を識別して、ダウングレードされたセキュリティ機能を持つ *CPA* 案を「再生回収」できる *CPA* 作成ソフトウェアを作成してもよい。他方の側でそのような *CPA* 案を受け入れるかどかは、当然*取引/コラボレーション*の開始および犠牲になるセキュリティ機能についてのその*当事者*の優先順位に依存している。*CPA* 作成ソフトウェアには、最終的に、これらの調整機能を持たせてもよい。当面は、このような状況のために、人間による支援が必要である。

もちろん、ある実装において、相互運用可能な実装のための要因の合致に失敗した場合、*CPA* の探索を中断することになるかもしれない。少なくともユーザには、セキュリティ機能、その他の必要な機能、または*取引/コラボレーション*で推奨されている機能が欠落した *CPA* しか提案できないと警告する必要がある。

その他の問題点

複数の能力の間の優先順位は、列挙されている順序によって示されるが、結合が発生する可能性もある。現在のところ、そうした結合は、本書では議論されていない交渉プロセスにおける解決に委ねられている。

著作権について

Copyright Statement