

ebXML 解説書

第 2 部

ebXML ビジネスプロセス

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2部 ebXML ビジネスプロセス 序文

「ebXML ビジネスプロセス」は、ebXML イニシャチブにより作成されたビジネスプロセスおよびビジネス情報分析概要、および ebXML ビジネスプロセス仕様スキーマを掲載する。

<参照 ebXML 仕様書>

Business Process and Business Information Analysis Overview V1.01 2001 年 5 月 11 日

ebXML Business Process Specification Schema V1.0 2001 年 5 月 11 日



Creating A Single Global Electronic Market

ビジネスプロセスおよびビジネス情報分析の概要 v1.0

ビジネスプロセス・プロジェクトチーム

2001年5月11日

技術検証 ECOM XML/EDI 標準化専門委員会

本文書の位置づけ

本技術報告書はビジネスプロセス・プロジェクトチームによって承認され、ebXML 総会によって受理された。

本文書には、ebXML の概念の解釈およびその実装を案内するための情報が記載されている。

本文書は自由に配布可能である。

本文書はインターネット・ソサイアティーの標準 RFC 形式で記載している。

本バージョン:

<http://www.ebxml.org/specs/bpOVER.pdf>

最新バージョン:

<http://www.ebxml.org/specs/bpOVER.pdf>

ebXML 参加者

ビジネスプロセス・プロジェクトチーム共同リーダー

Paul Levine, Telcordia

Marcia McLure, McLure-Moynihan, Inc.

以下の各氏には、本文書の作成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。

編集者:

Randy Clark, Baker Hughes, Inc

Brian Hayes, Commerce One

協力者:

James Bryce Clark, Spolin Silverman & Cohen LLP

Jim Clark, I.C.O.T.

Charles Fineman, Arzoon.com

Bob Haugen, Logistical Software LLC

Stephan de Jong, Philips International B.V.

Larissa Leybovich, Vitria Technology

Paul Levine, Telcordia

Bill McCarthy, Michigan State University

Marcia McLure, McLure-Moynihan, Inc.

Karsten Riemer, Sun Microsystems

Nita Sharma, IONA Technologies

David Welsh, Nordstrom.com

目次

1	本文書の位置づけ	vi
	ebXML 参加者.....	vi
2	目次	vii
3	はじめに	ix
3.1	総括	ix
3.2	適用範囲および対象読者	ix
3.3	関連資料	x
3.4	文書規約	x
4	目標および目的	xii
4.1	目標	xii
4.2	目的	xii
4.3	警告および前提条件	xii
5	取引コラボレーションの概要	xiii
5.1	ebXML の電子取引コラボレーション	xiii
5.2	ビジネスプロセスにおける経済要素	xv
5.3	ebXML の設計時および実行時参照モデル	xvii
6	ビジネスプロセスモデリング	xix
6.1	概要	xix
6.2	ビジネスプロセスおよび情報メタモデル	xix
7	分析プロセス	xxii
7.1	はじめに	xxii
7.2	推奨されているビジネスプロセスとビジネス情報の分析手法およびメタモデル	xxii
7.3	ビジネスプロセスおよび取引文書	xxiii
7.4	分析プロセス	xxvi
8	ビジネスプロセスとコア構成要素の間の関連	xxxi
8.1	はじめに	xxxi
8.2	ビジネス情報オブジェクト	xxxii
8.3	コア構成要素の分析	xxxiv
8.4	コア構成要素のコンテキスト内での分類	xxxiv
8.5	コンテキストと一般的なビジネスプロセス	xxxv
9	分析サポート:ワークシートおよびツール	xxxvi
9.1	分析ワークシートおよびガイドライン	xxxvi
9.1.1	分析ワークシートおよびエディタ	xxxvi
9.1.2	ビジネスプロセスエディタおよび文書エディタ	xxxvii
10	参考資料	xxxviii
11	免責	xxxviii
12	連絡先	xxxix
	著作権について	xliii



図 6.1-1 ebXML 取引コラボレーションプロセス	xiv
図 6.2-1 製品に関する典型的な注文-履行ビジネスプロセスにおける REA 経済要素の概要	xvii
図 6.3-1 ebXML の設計時および実行時参照モデル	xviii
図 7.2-1 UMM メタモデルと ebXML ビジネスプロセス仕様スキーマの関係	xx
図 7.2-2 ビジネスプロセス仕様と CPP/CPA の関係	xxi
図 8.3-1 ビジネスプロセス、コラボレーション、およびトランザクション概念ビュー	xxiii
図 8.3-2 文書概念ビュー	xxiv
図 8.3-3 メッセージおよびエンベロープの概念ビュー	xxv
図 8.4-1 ビジネスプロセスおよびビジネス情報の分析に関連したアクティビティ	xxvi
図 8.4-2 ビジネスプロセスおよびビジネス情報の分析	xxvii
図 8.4-3 ビジネスプロセスおよびビジネス情報の分析	xxviii
図 8.4-4 ビジネスプロセス分析アクティビティ	xxix
図 8.4-5 モデリング、XML への変換、および登録のアクティビティフロー	xxx
図 9.1-1 ビジネスプロセスとコア構成要素の間の関連	xxxii
図 9.2-1 ビジネス情報オブジェクトの構成	xxxiv
図 9.4-1 コンテキスト値の例	xxxiv
図 10.1.1-1 ビジネスプロセス分析ワークシートの使用	xxxvi

はじめに

1.1 総括

ebXML の構想では、企業が規模や所在地に関係なく、XML ベースのメッセージ交換を通して、互いに接触し取引を行うための、単一のグローバルな取引市場を構築することを目指している。ebXML によって、任意の企業が任意の場所で任意の相手と取引を行うことが可能になる。ただし、ebXML 準拠および採用は、時間の経過と共に漸増的に行われることが期待される。

企業が互いに電子ビジネスを行うためには、まず互いに相手の存在を発見し、その製品やサービスについて知る必要がある。次に、それらの製品やサービスを得るためには、どのようなビジネスプロセスおよび文書が必要になるかを決定する必要がある。その後、情報交換の実現方法を決定し、契約条件に合意する。このすべての手順が完了すれば、これらの合意に従って情報および商品/サービスを交換できるようになる。

この手順を促進するために、ebXML では、データ通信の相互運用性を実現するための基盤、商取引の相互運用性を実現するための意味情報フレームワーク、および企業が互いの存在を発見し、その関係を確立して取引を行うための機構を提供している。

データ通信の相互運用性は、正確に定義されたインターフェース、パッケージング規則、予測可能な納品/セキュリティモデル、および受信者、送信者のいずれかの側で受信メッセージおよび発信メッセージを処理するためのインターフェースによって実現されている。

商取引の相互運用性は、ビジネスプロセスモデルを定義する仕様スキーマと取引文書を定義するためのコア構成要素とコンテキストによって提供されている。ebXML では、1つの手法を推奨し、その方法に基づいて、モデルを作成するための一連のワークシートおよびガイドラインを提供している。ビジネスプロセスモデルおよび情報モデルのビジネスライブラリ（カタログ）によって、ビジネスプロセスや事前に定義されたビジネスプロセスの一部の再利用が奨励され、その結果、取引効率が向上する。

実際の取引行為を実現するために、ebXML では、企業が互いに当事者プロフィール情報から商品/サービスを発見するための共有リポジトリ、ビジネス契約書（コラボレーションプロトコル合意書; CPA）を取り交わすためのプロセス、会社プロフィール、ビジネスプロセスモデルおよび関連するビジネスメッセージを格納する共有リポジトリを提供している。

1.2 適用範囲および対象読者

本文書では、他の企業と電子ビジネスを行うために必要となる商取引の相互運用性（特に、企業がこれらのビジネスプロセスおよび取引文書を分析、識別、定義するためのプロセスについて）の ebXML フレームワークにおけるさまざまな側面について取り扱っている。

本文書は、マーケティング、ビジネス開発、経営管理、流通、ソフトウェア開発、IT など、企業内のさまざまな機能分野の代表者を対象として書かれている。

1.3 関連資料

ebXML Technical Architecture Specification. Version 1.0.4. 16 February 2001. ebXML Technical Architecture Project Team.

UN/CEFACT Modelling Methodology. CEFACT/TMWG/N090R9. February 2001. UN/CEFACT Technical Modeling Working Group.

Information Technologies - Open-EDI Reference Model. ISO/IEC 14662:1997(E). International Organization for Standardization (ISO) and International Electrotechnical Commission (IEC).

[bpWS] ebXML Business Process Analysis Worksheets and Guidelines v1.0. May 11, 2001. ebXML Business Process Project Team.

[bpPROC] ebXML Catalog of Business Processes. Version 1.0. Date May 11, 2001. ebXML Business Process Project Team.

[bpPATT] ebXML Business Process and Simple Negotiation Patterns. Version 1.0, May 11 2001. ebXML Business Process Project Team.

[ebBPSS] ebXML Business Process Specification Schema. Version 1.0 May 11 2001. Context/*Meta Model* Group of the CC/BP Joint Delivery Team.

[ebCCD&A] ebXML Methodology for the Discovery and Analysis of Core Components. V1.0, May 11 2001. ebXML Core Components Project Team.

[enCNTXT] ebXML The role of context in the re-usability of Core Components and Business Processes ebXML Core Components. Version 1.0, May 11 2001. ebXML Core Components Project Team.

[ebCCDOC} ebXML specification for the application of XML based assembly and context rules. Version 1.0, May 11 2001. ebXML Core Components.

[ebGLOSS] ebXML TA Glossary. Version 1.0, May 11 2001. Technical Architecture Project Team.

[ebRIM] ebXML Registry Information Model. Version 1.0, 11 May 2001. ebXML Registry Project Team.

[ebRS] ebXML Registry Services. Version 1.0, May 11 2001. ebXML Registry Project Team.

[ebCPP] ebXML Collaboration-Protocol Profile and Agreement Specification. Version 1.0, May 11 2001

[secRISK] ebXML Technical Architecture Risk Assessment Report. Version 1.0, May 11 2001.

1.4 文書規約

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)、する必要はある (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。

「メタモデル」という用語が使用された場合、UN/CEFACT モデリング手法[UMM]で定義されている電子ビジネスプロセスメタモデルを意味するものとする。

「仕様スキーマ」という用語が使用された場合、『ebXML ビジネスプロセス仕様スキーマ』で定義されているメタモデルおよび文書型定義（DTD）を意味するものとする。

目標および目的

1.5 目標

本文書の目標は、読者が ebXML フレームワークにおいて、ビジネスプロセスおよび文書作成の定義およびそれを識別するための一般的な方法を理解することにある。

1.6 目的

5.1 で設定された目標を達成するために、本文書では次の事柄を実現している。

- 電子取引コラボレーションの概要を示す。
- ビジネスプロセスモデリングの役割および利用法について議論する。
- 分析プロセスについて記述する。
- ビジネスプロセスにおける経済要素について議論する。
- コア構成要素のビジネスプロセスとの関係を確立する。

1.7 警告および前提条件

本文書の目的は、ビジネスプロセスおよび取引文書分析の概要を提供することである。仕様を意図したものではない。

対象となる読者が、ebXML フレームワークについてある程度の一般的な理解を得ていること、および特に『テクニカルアーキテクチャ仕様書』に精通していることが前提となっている。

ビジネスプロセス内の経済要素についての概念をより良く理解するには、資源-イベント-エージェント (REA) 企業存在論 (オントロジ) に精通していることが助けになる。

取引コラボレーションの概要

1.8 ebXMLの電子取引コラボレーション

ebXML テクニカルアーキテクチャの特徴は、電子取引コラボレーションのフレームワークを提供している点にある。このアーキテクチャによって、企業は協力して、ビジネスプロセスを規定し、互いの存在を発見し、提携契約書を取り交わし、ビジネスプロセスを実行することができる。ebXMLの電子取引コラボレーションにおける実装と実行に関する重要な活動を図 6.1-1 に示す。

プロセスの全体は、ビジネスプロセスおよび取引文書分析を利用してプロセスを定義することから始まり、論理的な順に、取引相手の検索、取引当事者間の合意、電子取引環境設定、プロセス実行、プロセス管理、プロセス改善と進む。

- **プロセス定義:** 企業が、ビジネスプロセス分析および取引文書分析を利用して、電子商取引に必要なプロセスを決定して定義する。取引当事者 (AIAG¹、RosettaNet² など) は、ビジネスプロセスをコミュニティ内で使用できるように定義する場合もある。これらのビジネスプロセスは、(周知の) モデルに従って定義され、合意された形式で記述される。
- **取引相手の検索:** ebXML 準拠のレジストリに登録されている会社プロフィールを検索することによって電子取引の当事者になる可能性のある相手を特定する。
- **取引当事者間の合意:** 次に、コラボレーションの契約条件となる契約書を取り交わす。
- **電子取引環境設定:** 取引当事者は、契約書に従ってインターフェースおよび企業サービスの設定を行う。
- **プロセス実行:** 文書を交換し、契約内容に従って商取引トランザクションを完了し、合意したビジネスプロセスを実行する。
- **プロセス管理:** 「プロセス定義」段階で定義し「取引当事者間の合意」段階で合意したビジネスプロセスは、取引当事者合意書に準拠しているかどうか、および、その実行に成功しているかどうかを確認するためにモニタされる。
- **プロセス改善:** 電子市場の参加者が、既存のプロセスを評価し、プロセスリエンジニアリングによってプロセスを改良し、マーケットの必要に合わせて新しいプロセスを作成する。

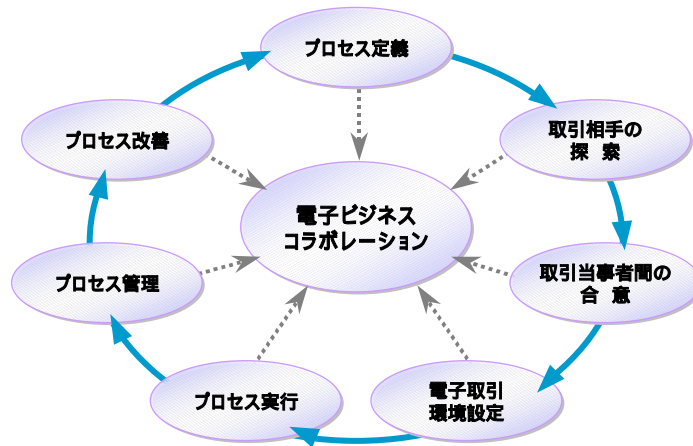


図 1.8-1 ebXML 取引コラボレーションプロセス

次の表に、図 6.1-1 内の ebXML プロジェクトチーム、重要な ebXML 文書、およびアクティビティの関係を示す。

アクティビティ	ebXML プロジェクトチーム	ebXML 文書
プロセス定義	ビジネスプロセス、CC/BP 分析サブチーム、レジストリ	<i>UN/CEFACT Modelling Methodology³, ebXML Business Process Specification Schema, Business Process and Business Document Analysis Overview, ebXML Business Process Analysis Worksheets and Guidelines, ebXML Catalog of Business Processes, ebXML The role of context in the re-usability of Core Components and Business Processes, and ebXML specification for the application of XML based assembly and context rules, ebXML Registry Services, ebXML Registry Information Model</i>
取引相手の検索	技術アーキテクチャ、取引当事者、レジストリ	<i>ebXML Technical Architecture Specification, Collaboration-Protocol Profile and Agreement Specification, ebXML Registry Services, ebXML Registry Information Model.</i>

³ UMM は、ebXML の文書ではないが、UN/CEFACT が管理する重要な文書である。

取引当事者間の合意	取引当事者、技術アーキテクチャ	<i>Collaboration-Protocol Profile and Agreement Specification, and Business Collaboration Patterns.</i>
電子取引の環境設定	技術アーキテクチャ、取引当事者	<i>Collaboration-Protocol Profile and Agreement Specification, ebXML Technical Architecture Specification, Information Technologies - Open-EDI Reference Model [ISO14662E]</i>
プロセス実行	取引当事者、技術アーキテクチャ、転送、ルーティング、およびパッケージング	<i>Collaboration-Protocol Profile and Agreement Specification, ebXML Technical Architecture Specification, Information Technologies - Open-EDI Reference Model [ISO14662E]</i>
プロセス管理	なし	<i>Information Technologies - Open-EDI Reference Model [ISO14662E] (Section Open-EDI Support Infrastructure)⁴</i>
プロセス改善	なし	なし - ebXML の適用範囲外

1.9 ビジネスプロセスにおける経済要素

最も一般的な ebXML 取引コラボレーションは、企業間の資源の交換（製品およびサービスの売買）である。資源の交換における最も一般的なコラボレーションパターンは、おそらく、注文-履行-支払いであろう。ebXML メタモデルでは、これらのコラボレーションを技術的な用語ではなくビジネス用語および経済用語で表現するための経済モデリング要素が提供されている。経済要素には次のものが含まれる。

- 商契約: 単純な注文から長期的なコンポーネント契約までを含む。
- 経済資源: 製品、サービス、現金など。
- 経済イベント: 商品またはサービスの配布、支払いなど。
- 当事者タイプ: 取引コラボレーションにおいて資源の売買契約および交換が許されている当事者、役割など。

これらの要素を使用して、取引コラボレーションを決定することが可能である。

- いつ商契約が形成されるか。
- いつ経済イベントが認識される必要があるか。
- いつ経済資源または資源に対する苦情は、一般会計原則（GAAP）に従って認識される必要があるか。

⁴ 『情報技術（IT）- 標準電子取引基準』[ISO14662E]は ebXML の文書ではない。しかし、UMM および ebXML の『Technical Architecture Specification』にとって重要な文書である。

- 納品により契約が履行されるかどうか。
- 納品によって注文が履行されない場合、どのようなイベントが後続するか。
- 取引上の観点では、いつ交換が完了するか。
- 典型的な取引関係における他の多くの側面。

ebXML 経済モデリング要素を用いれば、典型的な取引コラボレーションパターンを一度設計するだけでさまざまな状況で再利用が可能である。⁵ 図 6.2-1 に、製品に関する典型的な注文-履行ビジネスプロセスにおける REA 経済要素の概要を示す。

上記の概念および関係は UMM で指定されるが、最初のバージョンの『ビジネスプロセス仕様スキーマ』では、そのためのプログラムとしてのまとまったサポートは含まれていない。しかし、UMM メタモデルに基づいて、取引コラボレーション管理ソフトウェア(ツール)内でその実装を行ってもよい。

ビジネスプロセスは、『Catalog of Common Business Processes(共通のビジネスカタログ)』[CCBP] およびその他のビジネスライブラリから直接取得した、いくつかの取引コラボレーションから構成される。

- Query Product Information(商品情報照会)は、注文可能な製品に関する商品マスターまたはカタログ情報を受け取る。REA では、製品は取引財貨タイプである。
- Distribute Inventory Report(在庫レポート配信)は、現在入手可能な製品の情報を取得する。この処理は、QueryAvailability(入手)プロセスによっても達成できる。REA では、在庫は取引財貨である。個々の在庫要素は、商品マスター(取引財貨タイプ)によってタイプ分けされる。
- Create Order(注文生成)は、明細項目(取引約定)から成る注文書(取引契約)を形成する。個々の明細項目は、約束された数量の注文商品タイプを約束された日付および時刻に確保する。
- Notify of Shipment(出荷通知)を実行すると、1つ以上の Purchase Order の明細項目を履行する出荷(取引行為)が実行される。
- Process Payment(支払処理)を実行すると、出荷(REA の「双対」関係)に対する支払(取引行為)が実行される。

すべての明細項目が履行されすべての出荷に対して支払が完了すると、ビジネスプロセスが完了する。この単純な例においては、契約条件は、「受取り時に支払う」と指定されている。そうでなければ、このビジネスプロセスにはもう1つ別のステップ(請求処理など)が追加される。

何か問題が生じたため出荷が契約通りに行われず、支払が出荷に見合わなくなるか、あるいは取引行為が遅れたり不正確になったりした場合であっても、上述の REA の概念および関係を用いてその問題を表現することができる。

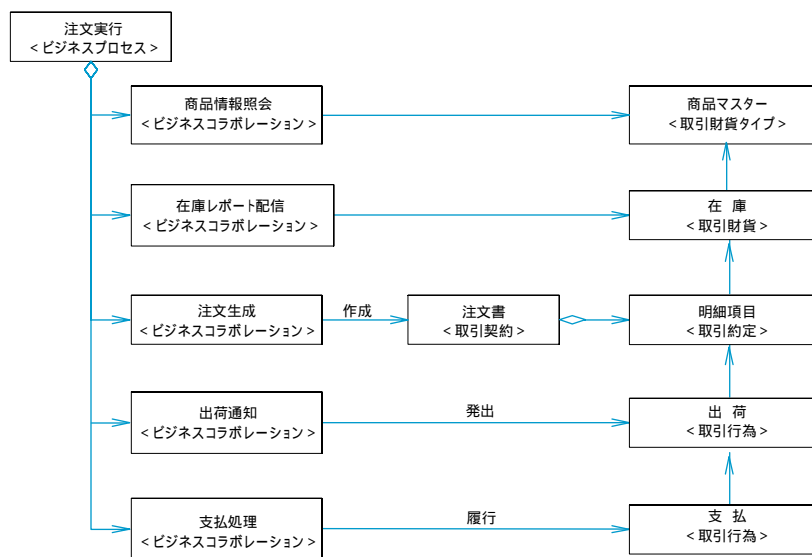


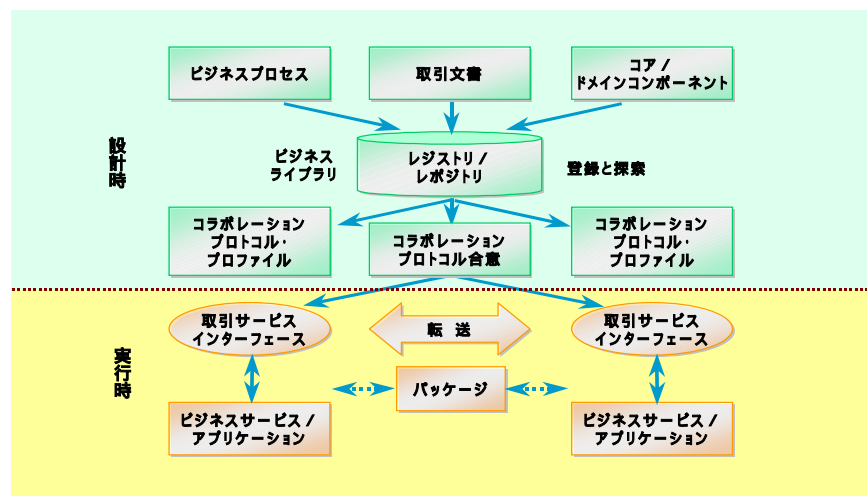
図 1.9-1 製品に関する典型的な注文-履行ビジネスプロセスにおける REA 経済要素の概要

1.10 ebXML の設計時および実行時参照モデル

適切なコンテキストをビジネスプロセスおよびビジネス情報分析に配置するには、ebXML のテクニカルアーキテクチャを考慮することが有用である。ebXML のテクニカルアーキテクチャは、設計時と実行時という、2つの基本的なコンポーネントから成る。ビジネスプロセスおよびビジネス情報分析は、設計時におけるコンポーネントの一部である。設計時におけるコンポーネントは、ebXML の基盤のアプリケーションを作成するための手続き、および取引トランザクションの実現に必要な ebXML 関連資源の発見および使用を

扱う。ビジネスプロセスおよびビジネス情報分析は、テクニカルアーキテクチャの設計時におけるコンポーネントを達成する 1 つの方法である。

実行時におけるコンポーネントは、実際の ebXML トランザクションを用いた ebXML シナリオの実行を扱っている。



ebXML アーキテクチャの設計時および実行時コンポーネントを、図 6.3-1 に示す。

図 1.10-1 ebXML の設計時および実行時参照モデル

設計時に人工的に生成される成果物によって、システムによるビジネスプロセス(合意済み)の実行が可能になる。ビジネスプロセスおよび取引文書は、ビジネスプロセスおよびビジネス情報分析アクティビティの間に定義される。コア構成要素およびドメインコンポーネントは、文書の内容および構造を指定するために使用される再利用可能な情報の部品といえる。これらのコンポーネントは、ebXML のコア構成要素の発見および分析のための手法を用いて特定および定義できる。定義済みのビジネスプロセスおよび取引文書についての仕様は、ビジネスライブラリに格納および登録されている。ビジネスライブラリには、ビジネスプロセスオブジェクトおよびビジネス情報オブジェクト(文書コンポーネント)のカタログも含まれており、これらのカタログは、ebXML 準拠のレジストリおよびレポジトリに格納されている。

ビジネスプロセスモデリングの結果、ebXML ビジネスプロセス仕様が生成され、この仕様は、企業のコラボレーションプロトコルプロファイル(CPP)内で参照可能であり、ビジネス当事者間で確立されたコラボレーションプロトコル合意書(CPA)の基礎となっている。最終的には、CPAで規定されているビジネスプロセスによって、プロセスを実行するための取引サービスインターフェースが起動され、必要な文書が送付される。

ビジネスプロセスモデリング

1.11 概要

ビジネスプロセスモデルは、ビジネスプロセスの記述方法を定義している。ビジネスプロセスは、電子ビジネスの「動詞」概念を表し、モデル作成ツールを用いて表現できる。ビジネスプロセス定義の仕様を使用すれば、企業は、他の企業が理解できるようにビジネスプロセスを表現でき、これによって、企業内、企業間のビジネスプロセスの統合が可能になる。

ビジネスプロセスモデルによって、ビジネスプロセスが規定され、ビジネス当事者が協力できるようになる。万が一ビジネス上の慣習が組織によって異なっていたとしても、ほとんどのアクティビティを一般的なビジネスプロセスに分解することで、適応させることができる。このようなビジネスモデリングを用いた分析を行うことによって、標準化される可能性の高いビジネスプロセスおよびビジネス情報メタモデルを特定できる。ebXMLのアプローチでは、相互操作可能にするための標準的で再利用可能なコンポーネントを規定することを目指している。

1.12 ビジネスプロセスおよび情報メタモデル

メタモデルとは、取引当事者が一貫したモデリング手法を用いて、特定のビジネスシナリオを詳細まで把握できる、ビジネスの意味情報についての記述のことである。ビジネスプロセスには、取引当事者が、共有されたビジネスプロセス内の他の取引当事者と相互作用する為には、どのように役割、関係、および責任を用いて促進していくかが詳細に記述される。役割の間の相互作用は、取引トランザクションの集合として実現される。各々の取引トランザクションは電子ビジネス文書の交換として表現される。交換の順序は、ビジネスプロセス、メッセージング、およびセキュリティを考慮して定義される。ビジネス文書は、再利用可能なビジネス情報コンポーネントから構成される。下位レベルでは、ビジネスプロセスは再利用可能なコアプロセス、ビジネス情報オブジェクトは再利用可能なコア構成要素からそれぞれ構成できる。

UMMメタモデルでは、ビジネスプロセスがサポートされ、ビジネスプロセスと情報の統合および相互運用性を促進するために必要な意味情報および生成物の仕様の基礎が構成される。

また、UMMメタモデルでは、ebXMLビジネスプロセス仕様スキーマも提供されており、1組のebXML取引トランザクションを実行するシステムを構成するために必要な要素の項目名一覧が仕様によってサポートされる。ebXMLビジネスプロセス仕様スキーマは、他のいくつかのビューからのモデリング要素の描出に基づいて、UMMメタモデルの意味情報の部分集合を構成する。ebXMLビジネスプロセス仕様スキーマは、UMLバージョンおよびXMLバージョンの2種類の表現形式が利用できる。

現在のところ、ebXMLで使用されるUMMメタモデルの必須の部分は、ebXMLビジネスプロセス仕様スキーマで表現される意味情報のみである。しかし、UN/CEFACTによって、UMMの仕様がバージョンアップするとき、UMMメタモデルの他の部分も必須になると思われる。

UMMメタモデルとebXMLビジネスプロセス仕様スキーマの関係は、次のように示される。

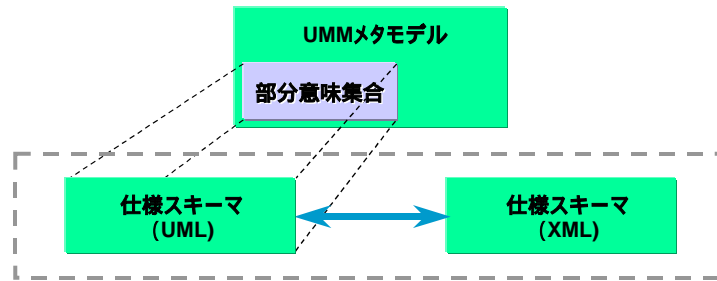


図 1.12-1 UMM メタモデルと ebXML ビジネスプロセス仕様スキーマの関係

ebXMLビジネスプロセス仕様スキーマでは、取引トランザクションの仕様、および取引コラボレーションへの振り付け図がサポートされている。各取引トランザクションは、入手可能な多くの標準パターンの1つを用いて実装できる。これらのパターンによって、必要な電子トランザクションを達成するための、取引当事者間の実際の取引文書およびシグナルの交換が決定される。パターンを規定するために、UMMでは標準パターンが提供されており、ebXMLビジネスプロセス仕様スキーマではそれらのパターンをサポートするモデリング要素が提供されている。ebXMLのビジネスプロセスの仕様は、「ビジネスプロセス仕様」と呼ばれる。ビジネスプロセス仕様は、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)を構成するための主要な入力情報として機能する。

これは、下図の通り示すことができる。

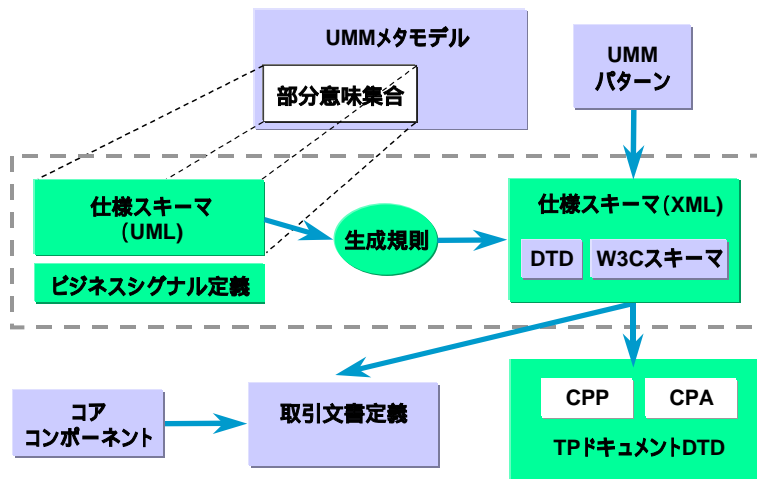


図 1.12-2 ビジネスプロセス仕様と CPP/CPA の関係

単で一貫したモデリング手法を使用する主な利点の1つは、モデルを比較して、既存のビジネスプロセスとの重複がないようにすることができるという点である。

さらに一貫したビジネスプロセスおよび情報モデルの作成を促進するために、ebXMLでは、コアライブラリだけでなく共通のビジネスプロセスも定義している。こうすることで、ebXML基盤のユーザが、この集合を拡張して、独自のビジネスプロセスを使用したいと考えた場合にも、対処することができる。

分析プロセス

1.13 はじめに

以下に記述されているプロセスは、他の企業と電子商取引を行うために必要なビジネスプロセスおよびビジネス文書の分析によって、企業をサポートするためのものである。このうち、ビジネスプロセスの分析は、電子ビジネスを行うのに必要な上位レベルのプロセスを扱う。ビジネス情報およびドキュメントアクティビティの分析を行うことで、ビジネスプロセスのコラボレーション内のビジネストランザクションに關与するビジネス文書を特定できる。アクティビティの分析のアウトプットは、ビジネスプロセスの仕様およびビジネス文書の定義である。

分析は、IT、マーケティング、ソフトウェア開発、経営分析、流通の専門家などの幅広い分野の人々で構成させる分析チームによって行うのが最善である。本節に記述されている分析プロセスを適用する際の分析チームは、ビジネスプロセス分析またはプロセスリエンジニアリングの経験がある人員で構成されていることが望ましい。分析者は、不適切に定義された要件やオブジェクトを含むビジネスプロセスの分析を行う際に生じる問題について十分理解していることが前提となっている。

そのような分析チームは、『ebXML ビジネスプロセス分析ワークシート』、UML モデル作成ツール、またはそれらと類似した機能を提供するビジネスプロセスエディタを使用することが望ましい。(10 節参照) また、分析チームは、企業全体によって審査および検証できる ebXML ビジネスプロセス仕様、およびメタモデル、仕様スキーマに基づいてモデルを作成するために必要なすべての情報を開発できる。この分析プロセスは、新しいプロセスおよびプロセスリエンジニアリングの分析、および既存のプロセスの分析、文書化をサポートしている。

1.14 推奨されているビジネスプロセスとビジネス情報の分析手法およびメタモデル

分析チームでは、電子ビジネスコミュニティ内でビジネスプロセスを規定するための手法およびメタモデルが使用される。分析手法によって、ビジネスプロセスを定義する際にチームによって実行される、全体のプロセスとサブプロセスが規定される。分析プロセスの最中に発見および文書化される必要がある情報はメタモデルの意味情報によって定義される。手法には、多くの場合、モデルの「設計」の促進および類似概念の一般表現の獲得をサポートするためのパターンが含まれている。

ebXML では、分析チームにおいて UN/CEFACT モデリング手法によって指定される手法の使用を推奨している(必須ではない)。別の手法が使用された場合でも、UN/CEFACT モデリング手法を用いて作成されるビジネスプロセスモデルと互換性のあるビジネスプロセスモデルを作成できるように、UN/CEFACT モデリング手法に準拠することが強く推奨される。

ebXML では、分析の結果生成されるビジネスプロセスとビジネス情報の生成物が、UN/CEFACT モデリング手法の電子ビジネスプロセスメタモデルで定義されている意味情報と UN/CEFACT モデリング手法で定義されている意味情報に準拠することが要求される。これは、ビジネスプロセスモデルとモデルのサブコンポーネントの間に互換性を持たせる為である。この意味情報の準拠性は、モデルを利用可能および再利用可能にし、比較および対比を可能にするために必要である。電子ビジネスのプロセスメタモデルに準拠しているモデルを用いれば、ユーザの働きとツールの利用によって、そのモデルの ebXML ビジネスプロセス仕様スキーマの XML インスタンスを生成できる。さらに、そうしたモデルは、ebXML 準拠のモデル作成ツール(UML ツールを含むがそれだけではない)の間で自由に共有可能である。

1.15 ビジネスプロセスおよび取引文書

ビジネスプロセスとは、基本的なレベルでは、「ビジネス上の行為において1つ以上のアクティビティを到達するための手法」を意味する。ビジネスプロセス内に、2つ以上のトランザクションを含んだ1つ以上のコラボレーションが存在する場合もある。下の図 8.3-1 は、ビジネスプロセスを単純な表現した図であり、材料の注文を完了するために、顧客と業者の間で必要となる可能性があるビジネスプロセスの種類を示している。

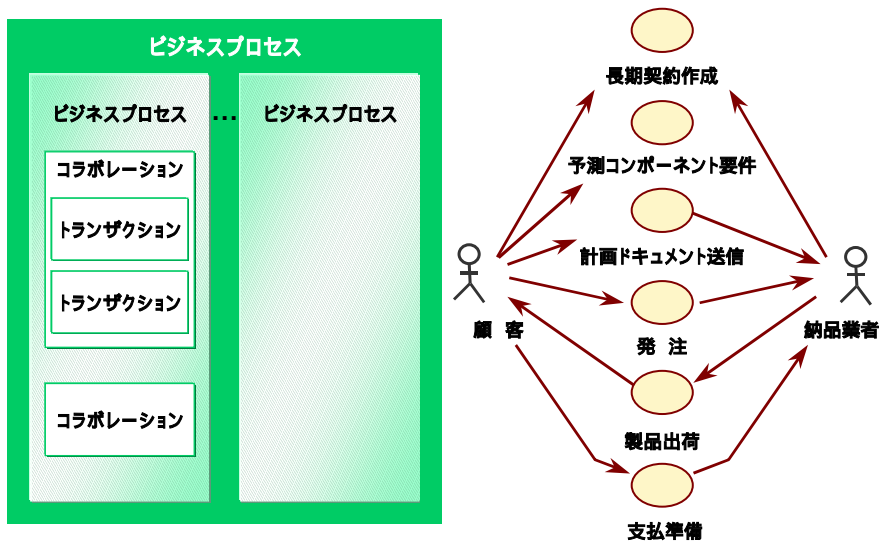


図 1.15-1 ビジネスプロセス、コラボレーション、およびトランザクション概念ビュー

取引文書定義とは、取引文書およびそれに含まれている情報コンポーネントを構成するビジネス文書スキーマおよび情報コンポーネントの仕様のことである。ビジネス文書のスキーマ状の表現を下の図 8.3-2 に示す。

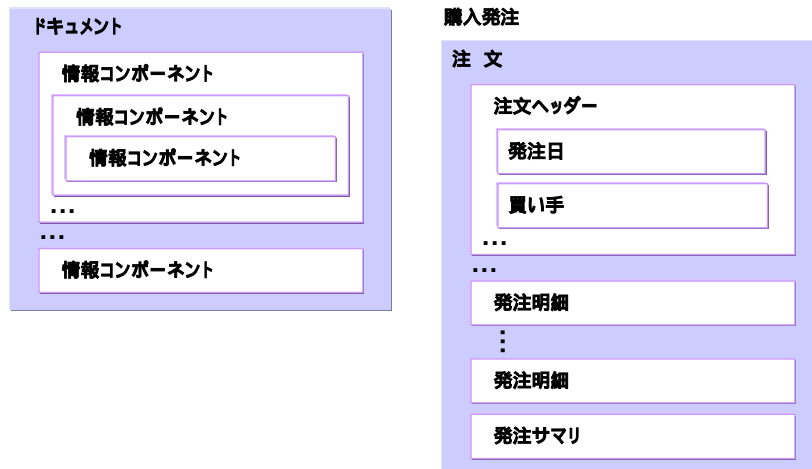


図 1.15-2 文書概念ビュー

注文書、請求書などの文書は、ビジネスプロセスレベルに存在し、取引トランザクションで交換される。文書は、文書エンベロープに入れられる。送付先および送付元のビジネス識別子（DUNS 番号）を用いて宛名書きされる。これは、通常の郵便の宛名における「Attention:」（宛）の欄に相当する。文書エンベロープは、メッセージエンベロープに配置され、ビジネスサービスインターフェースの間で交換される。メッセージエンベロープは、送付先のビジネスサービスインターフェースの URN を用いて宛名書きすることもできる。メッセージには、タイムアウトが付けられ、他のトランザクションの制御機構が関連付けられる。メッセージエンベロープは、転送/ルーティングエンベロープに配置され、電子ビジネスネットワークの中で下位レベルの送信に使用される。メッセージエンベロープに付けられるアドレスは、送付先のメッセージインボックスサービスの URL でもよい。入れ子になったエンベロープ構造の論理的な表現を図 8.3-3 に示す。

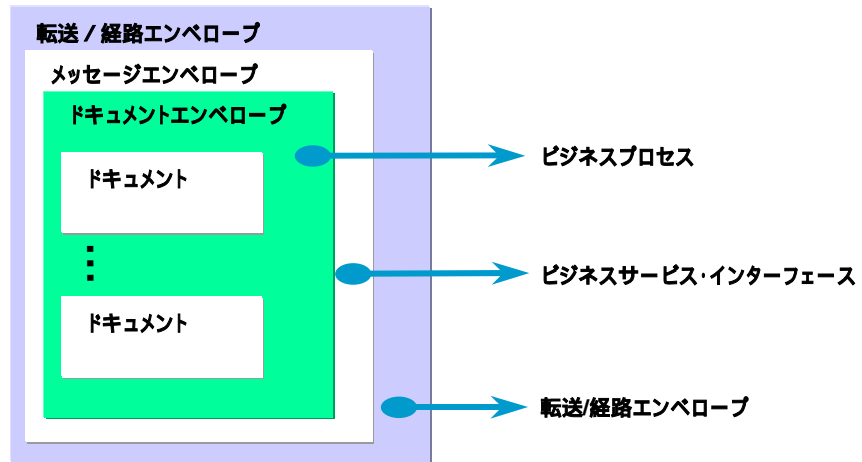


図 1.15-3 メッセージおよびエンベロープの概念ビュー

1.16 分析プロセス

ビジネスプロセスおよびビジネス情報分析に関連した高レベルのアクティビティを図 8.4-1 に示す。

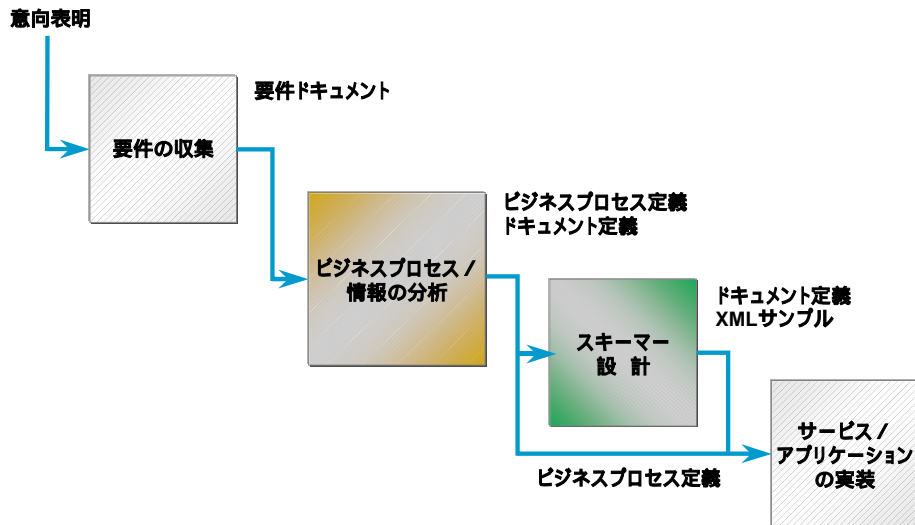


図 1.16-1 ビジネスプロセスおよびビジネス情報の分析に関連したアクティビティ

最初のステップでは、チームの目標を一致させるために、分析アクティビティの適用範囲および目的について明確に述べた、目的の宣言 (Statement of Intent) を作成すると役立つ。

次のステップには、目的の宣言に基づいて要件を収集することが含まれる。マーケティングおよび製品管理チームでは、多くの場合、要件収集が実行される。このアクティビティの結果は、マーケティング要件文書または商品要件文書のいずれかにまとめられる。ただし、いずれの場合も、その結果は明確に定義された要件の集合でなければならない。

要件が定義され合意されたら、実際の分析を開始する。図 8.4-2 に示したように、要求された出力を生成するのに必要なプロセスには、さまざまな要素の入力および側面が存在する。分析プロセスの入力は、要件、顧客と取引先、規格、他の既存のモデル、および業務専門家から得られる可能性がある。要件は、商品要件文書、機能の宣言、顧客変更要求などの形式である場合がある。顧客、当事者、および業務専門家は、入力する要素をプロセスおよび文書作成をレビューする際に検討する。既存の規格 (業界に特定されない規格および特定の業界の規格) および他の既存のモデル (EDI メッセージ実装ガイドなど) も参照可能である。

分析アクティビティの制御は、手法（UMM）、メタモデル、パターン、および他の分析技術によって行われる。⁸これらの制御によって、ビジネスプロセスおよび情報分析プロセスが正しい出力を生成するために必要なプロセスおよび情報モデルが規定される。パターンには、トランザクションパターン[UMM]およびコラボレーションパターン[ECPAT]が含まれる。

分析アクティビティのメカニズムは、分析者、ツール、およびレビュー者によって行われる。分析者とは、メタモデルに基づいてプロセスおよび文書を定義する人たちのことである。

分析の助けになる主要なツールの1つは、10節「分析支援:ワークシートおよびツール」で述べられている『ビジネスプロセス分析ワークシート』である。

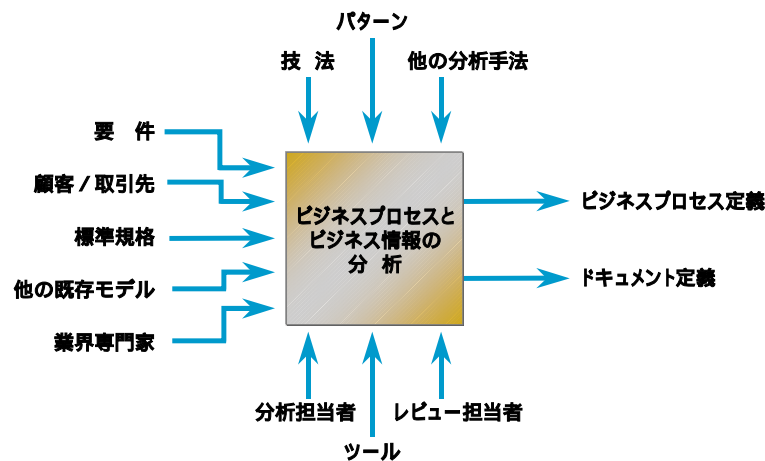


図 1.16-2 ビジネスプロセスおよびビジネス情報の分析

ビジネスプロセスおよびビジネス情報の分析アクティビティは、図 8.4-3 に示す通り、「ビジネスプロセスの分析」および「ビジネス情報の分析」という2つの独立であるが相互に関連したアクティビティに論理的に分割される。

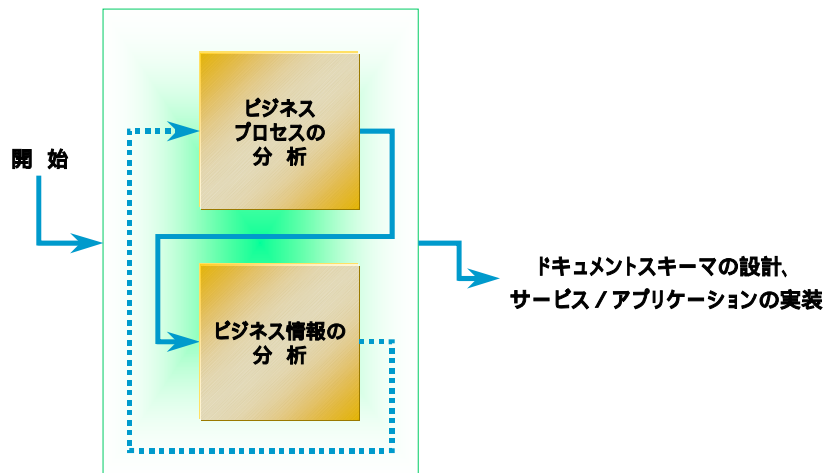


図 1.16-3 ビジネスプロセスおよびビジネス情報の分析

ビジネスプロセスの分析とそれに関連したビジネス情報の分析は、同時に行う方が、より効果的になる。ビジネス情報を分析する分析者は、ビジネスプロセスに精通していなければならないと同時に、ビジネスプロセス分析に参加することになる。そうでない場合は、ビジネス情報を分析する分析者は、より効果的に問題を解決するために、ドメイン専門家、顧客、および当事者に再度インタビューしなければならない。さらに、ビジネス情報分析者は、ビジネスプロセスに影響を受けることが予測される、主要なビジネス情報要素を特定する際の手助けとなるような背景を持つことになるだろう。

「ビジネスプロセス分析」アクティビティは、モデリングの焦点によって異なる経過を辿る場合がある。たとえば、特定の業界のビジネス参照モデルを構築することが目標である場合は、UMMの議論で終わる可能性が高い。しかし、X12またはEDIFACTの既存の文書およびそれに関連したビジネスプロセスをモデル化することが目的である場合は、プロセスは、取引トランザクションおよび役割の詳細化から開始することになる。この場合、通常は、ドメイン専門家が、関連したビジネスプロセスについてを非常に深く理解しているはずである。図 8.4-4 に示すように、ビジネスプロセス分析は、4つの高レベルのアクティビティに分割できる。⁹

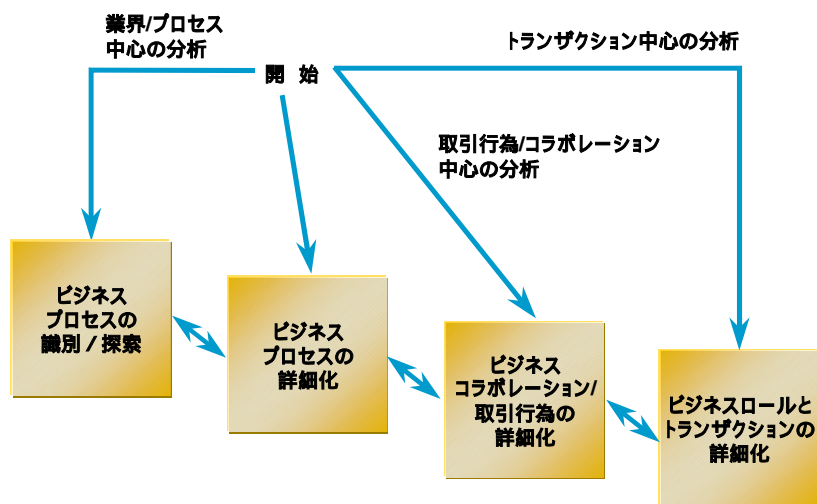


図 1.16-4 ビジネスプロセス分析アクティビティ

ビジネスプロセスおよびビジネス情報の分析の次のアクティビティは、スキーマの開発およびサービスの実装である。スキーマの開発には、文書および情報コンポーネントスキーマ（XML スキーマ/DTD または EDI メッセージ、データ要素定義）、サンプル文書の作成が含まれている。サービス/アプリケーションの実現には、ビジネスプロセス定義および文書スキーマに従って行う、取引サービスインターフェースおよびサービス/アプリケーション（バックエンドシステムなど）の実装または設定が含まれている。

分析が完了し、ビジネスプロセスおよび文書が完全に定義され開発されたら、その仕様をビジネスライブラリ[TAS]に登録する必要がある。ビジネスライブラリは、1つの業界内または複数の業界にわたって共有されるビジネスプロセス仕様およびビジネス情報オブジェクトのリポジトリである。ビジネスライブラリには、「パブリック（public）」および「プライベート（private）」、「コントロールあり（controlled）」および「コントロールなし（not controlled）」など、さまざまなものがある。パブリックライブラリは、公にアクセス可能なライブラリである。通常、パブリックライブラリの内容は、標準化団体（ebXML、UN/EDIFACT など）および大規模な電子コミュニティ（自動車の取引市場など）が所有することになる。一方、プライベートライブラリは公にはアクセスできない。これは、参加者がビジネスプロセスの性格を開示したくないと考える場合に、内輪で情報交換を行うための方法として用いられる。明らかに、公にアクセス可能なビジネスライブラリは、さまざまな電子コミュニティにおける取引当事者の間の相互運用性を促進する最も有用な手段である。たとえば、自動車コミュニティの取引当事者の電子ビジネスシステムが化学製品コミュニティに登録されているビジネスプロセスにアクセスするためには、パブリックライブラリが必要である。

穏当なビジネスライブラリは、標準化団体や電子コミュニティなどの組織が管理している。ビジネスプロセスおよびビジネス情報の仕様は、コントロールされているビジネスライブラリを運用している作業グループまたはその他の監督アクティビティに提出されることになる。この作業グループは、提出され

た情報の品質および精度を審査することになる。この仕様は、パブリックに配置する場合も、承認のためにコミュニティで票決する場合もある。次に、承認された仕様は、ビジネスライブラリに登録される。その際、主要なモデル要素（ビジネスプロセス、取引コラボレーション、取引トランザクションなど）は、『Business Identifier Naming Scheme』[BPAWAG]に従って、正式な識別子が割り当てられる。これらの識別子によって、ビジネスプロセス仕様、コア構成要素の文脈分類、CPP、およびCPAで参照できる一意の識別子が与えられるため、再利用性および相互運用性が促進される。通常、コントロールされているビジネスライブラリは、コントロールされていないビジネスライブラリよりも信頼性が高い。コントロールされていないビジネスライブラリの場合、コミュニティの任意のメンバーが登録することを許可している。したがって、その品質および精度は疑わしくなる。しかし、コントロールされていないライブラリがビジネスプロセス仕様において重要になる場合もある。利用度の高いビジネスプロセス仕様は、時間の経過と共に広く認識されるようになる。

これらの仕様が企業の分析プロセスとして利用されるかどうかは、主に、さまざまなビジネスモデル作成ツールで相互運用可能な形式で格納されているかどうかによるので、これらの仕様を格納する形式は十分考慮する必要がある。RDFによって、ビジネスプロセスのライフサイクル内の分析、設計、および「後代のための記録」段階で、ビジネスプロセスモデルをカプセル化する機会が得られる。さらに、RDFの利用によって、UN/CEFACTにおけるebXMLの元来の目標の1つであった、さまざまな規格間のモデル仕様の相互交換が可能になり、その結果、ビジネスプロセスのグローバルなモデリングおよび共通解法の再利用が促進される。XMIなどの他の形式と比較した場合、RDFの利点は、名前空間を特定の問題領域に制限できる点にある。他の形式では通常、より一般的なUMLドメインに従っている。RDFにおけるメタ構造の記述および有効化の機能によって、モデル内容の適用可能性について効果的に制御できるようになる。B2Bなどの緊縮的なドメインでモデルを使用する際には、メタ構造に従ってモデル内容を有効化できる機能は魅力的である。また、ビジネス情報の見地から見ると、RDFがBusinessAction要素への関連付け（ビジネスプロセスのコンテキストにメッセージを置くこと）を許していることは有用である。

分析全体およびその結果の要約を、図 8.4-5 に示す。

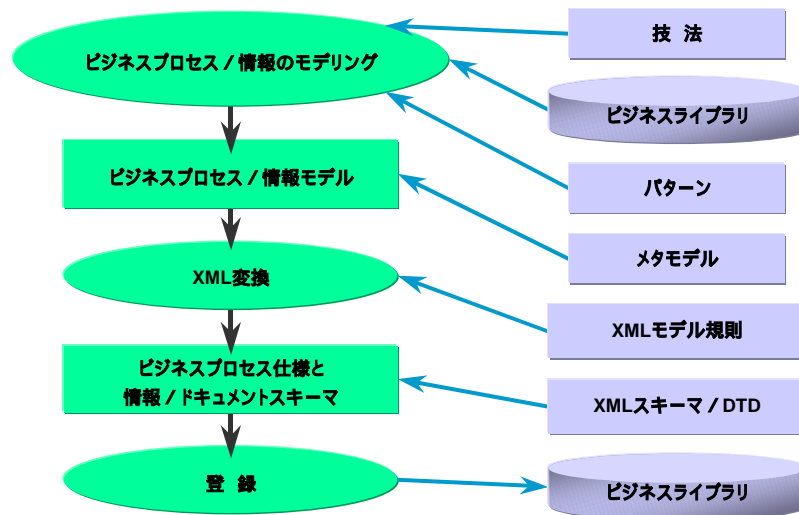


図 1.16-5 モデリング、XML への変換、および登録のアクティビティフロー

このフローの全体は、ビジネスプロセスおよびビジネス情報の分析およびモデリングから始まる。UMM 手法は、『ビジネスプロセスワークシート』またはビジネスプロセスエディタを通して直接または間接的に利用できる。ビジネスライブラリ内の再利用可能なビジネスプロセスおよび情報コンポーネントの適用は、コラボレーション、トランザクションパターンと同様に利用できる。分析の結果、メタモデルに基づくビジネスプロセスモデルおよびビジネス情報モデルが生成される。生成されたモデルは、生成規則に従って、XML ベースのビジネスプロセス仕様および情報/文書スキーマに変換される。そして最後に、CPA で再利用および参照できるように、仕様およびスキーマが登録および格納される。

ビジネスプロセスとコア構成要素の間の関連

1.17 はじめに

前述のように、ビジネスプロセスモデルによって、ビジネスプロセスの記述方法が定義され、電子ビジネスの「動詞」概念が表現される。情報モデルによって、ビジネスコンテキスト内で標準の方法で適用できる再利用可能なコンポーネントが定義される。コア構成要素およびドメインコンポーネントによって、電子ビジネスの「名詞および形容詞」概念が表現される。これらの情報では、すべての企業に共通な同一性アイテムを用いて定義される。その結果、ユーザは、自社の事業にとって意味のあるデータを定義することが可能になると共に、他のビジネスアプリケーションとも相互運用性を維持することができるようになる。図 9.1-1 に、再利用可能な情報コンポーネントがビジネスプロセス内で機能する様子を示す。

Components used in modeling a Business

Scenario

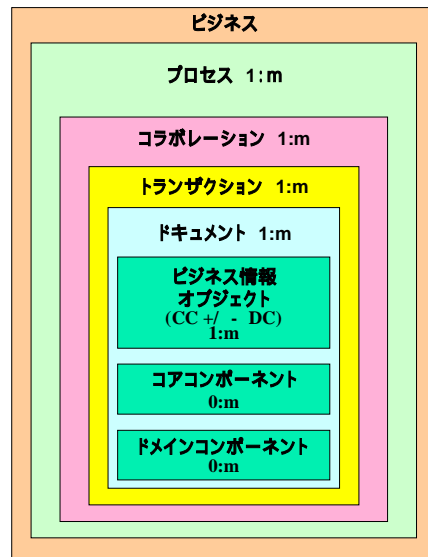


図 1.17-1 ビジネスプロセスとコア構成要素の間の関連

1.18 ビジネス情報オブジェクト

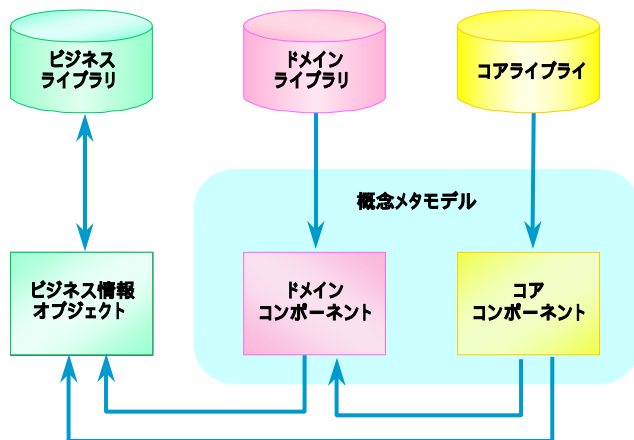
ビジネス情報オブジェクトは、コア構成要素、ドメインコンポーネント、および他のビジネス情報オブジェクトから構成されていてもよい。コンポーネントおよびビジネス情報オブジェクトの定義は、ビジネスライブラリに格納される。コア構成要素は、特別な名前の付いたコアライブラリに格納できる。ビジネス文書定義は、ビジネス情報オブジェクト、ドメインコンポーネント、およびコア構成要素から構成される。取引文書定義の開発方法は、次のように記述される。

1. ビジネスライブラリで、ビジネス情報オブジェクト内で利用可能な属性を検索する。
2. 適切な属性の付いたビジネス情報オブジェクトが利用できない場合は、新しいビジネス情報オブジェクトを作成する必要がある。
3. コアライブラリ内のビジネスライブラリおよびコア構成要素に含まれるドメインコンポーネントは、コンテキストが適切な場合は、ビジネス情報オブジェクトの属性候補である可能性がある。
4. 属性を追加するには、既存のビジネス情報オブジェクトに新しい属性を追加する。または、ビジネスライブラリの変更を管理する登録プロセスを通して新しいビジネス情報オブジェクトを作成する。
5. 新しい属性を使用しているビジネスライブラリ内に、取引文書を作成する。

要約すると、ビジネスプロセスおよび情報モデルにおいて取引文書を作成するための主要なソースは、ビジネスライブラリ内のビジネス情報オブジェクトであるということになる。一方、適切なビジネス情報オブ

エクトが見つからない場合、2 次的なソースとして、ビジネスライブラリ内のドメインコンポーネントおよびコアライブラリ内のコア構成要素を使用する。ビジネスライブラリが構築または信頼できるソースからインポートされるまで、コア構成要素が頻繁に使用される傾向がある。コア構成要素は、まずビジネスライブラリ内のビジネス情報オブジェクトのレパートリーに追加され、次にビジネス文書の作成のために使用される。

Core Components Business Information Objects, Domain Components



Core Components Business Information Objects, Domain Components

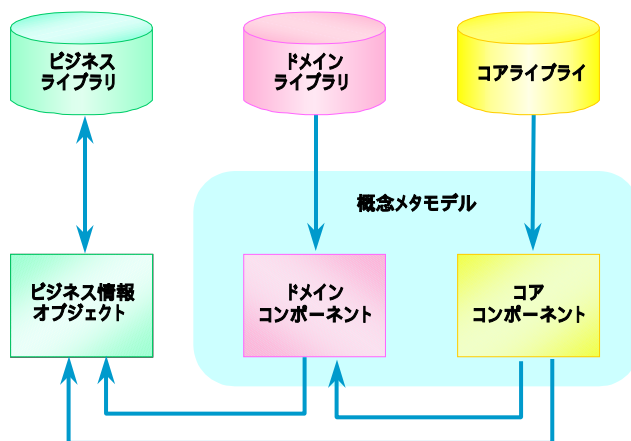


図 1.18-1 ビジネス情報オブジェクトの構成

1.19 コア構成要素の分析

コア構成要素の検索および分析方法（Methodology for the Discovery and Analysis of Core Components）によって、業界横断的に再利用可能な情報コンポーネントを特定するプロセスが記述される（このため、「コア構成要素」という用語が用いられる）。コア構成要素は、ドメインコンポーネントおよびビジネス情報オブジェクトの構築に使用される。ビジネスプロセス仕様のライブラリ『ebXML Catalog of Common Business Processes』[CCBP]』を含むビジネスライブラリは、コア構成要素およびドメインコンポーネントの発見および分析の手段として使用される。

ビジネスプロセス仕様には、コア構成要素およびコア構成要素の要素のコンテキスト内での利用法を記述する値が含まれている。この点は、さらに 9.4 節「コア構成要素のコンテキスト内での分類」で議論される。コア構成要素の分析を、関連するビジネスプロセス、トランザクション、および文書をさまざまな標準（EDIFACT、X12、xCBL、RosettaNet、CII、OAG など）から特定することによってサポートするために、ビジネスライブラリの相互参照『ebXML Catalog of Common Business Processes』などが用意されている。

1.20 コア構成要素のコンテキスト内での分類

メタモデルによって、ビジネスプロセスをモデリングする際に捉える必要のある情報が指定される。モデルには、ビジネスプロセスおよび文書モデルの他の要素の間の相互条件に重大な影響を与えると見なされる要素および属性が多数含まれている。分析プロセスの間に、これらのさまざまなモデル要素の間のコンテキスト的な依存関係を理解することは有用であり、さらに、将来は、コンテキスト的な依存関係の実行時の適用が可能になる場合がある。⁸

コンテキスト的な依存関係という概念（「コンテキスト」と呼ばれる）は、再利用可能な情報コンポーネントの分析に重要な役割を果たすので、ebXMLのコア構成要素プロジェクトチーム（Core Components Project Team）によって詳細に検討されている。ビジネスプロセスを実行する際には、コンテキストカテゴリおよびそれに関連した値の集合によって実行場所となるコンテキストを指定できる。たとえば、自動車製造業者が化学製品製造業者から塗料を購入する場合、コンテキスト値は次のようになるかもしれない。

コンテキストカテゴリ	値
プロセス	調達
商品分類	塗料
地域	米国
業界（買い手）	自動車
業界（売り手）	化学製品

図 1.20-1 コンテキスト値の例

⁸ ドキュメント要素（共通情報構成要素）に関するこの話題についての詳細は、『共通情報構成要素およびビジネスプロセスの再利用可能性におけるコンテキストの役割』を参照。

コンテキストカテゴリ（「コア構成要素およびビジネスプロセスの再利用可能性におけるコンテキストの役割」で特定される）は、UMM ビジネスプロセスメタモデルに準拠したビジネスプロセス内の既存の要素および属性に単純に対応している。たとえば、コンテキストカテゴリ「プロセス」は、メタモデル要素の BusinessProcess、ProcessArea、および BusinessArea に対応している。コンテキストカテゴリの対応付けは、付録 A に示されている。

1.21 コンテキストと一般的なビジネスプロセス

ビジネスプロセスモデルにおけるコンテキストの役割は、ebXML の適用範囲を超えているため、ebXML では正式には取り扱われていない。しかし、共通のビジネスプロセスモデルを自社のコンテキスト利用法に基づいて拡張または制限することは、一般に受け入れられている。たとえば、ビジネスプロセス X には、業界が「自動車」である場合は XY、業界が「小売」である場合は XX、という制限（拡張）された動作がそれぞれ割り当てられるかもしれない。ビジネスプロセスのコンテキストは、ビジネス地域、プロセス地域、業界、役割、取引行為/取引財貨（場合による）などのモデリング要素の値によって定義される。この点は、コア構成要素および文書仕様に適用されるため、コンテキストの概念に類似している。コンテキストおよびコア構成要素についての詳細は、『ebXML The role of context in the re-usability of Core Components and Business Processes』を参照。

分析サポート:ワークシートおよびツール

分析およびモデリングに関する専門的知識がない場合は、UMMを参照マニュアルとして使用すると有用である。その場合、ビジネスプロセスの分析中に、UMM準拠の手法や代替手法を使用することになる。実際の経験から考えると、電子ビジネスコミュニティでは、分析およびモデリングの専門的知識を必要としない手法の方が有用である。つまり、事業家自身が適用できる手法が最も有用である。ビジネスプロセス分析ワークシートおよびガイドラインは、そのようなアプローチを提供している。

1.22 分析ワークシートおよびガイドライン

『ebXML Business Process Analysis Worksheets [BPAWAG]』は、UMMと共に参照先として使用されるビジネスプロセス分析の設計支援フォームである。このワークシートを使用すると、ユーザは、ビジネスプロセスを完全に記述するのに必要なすべての情報を取得できる。この記述は、ソフトウェアを起動するために使用したり、登録、分類、発見、および再利用したりできる。

1.22.1 分析ワークシートおよびエディタ

ワークシートの構築には、ブラウザベースのフォームが使用されることが意図されている。ユーザは、ビジネスライブラリ（ビジネスプロセス仕様のカタログを含むレジストリまたはレポジトリ）で既に定義済みのアイテムを探索する過程でワークシートを作成できる。この関係を、図 10.1.1-1 に示す。図 10.1-1 項目（ビジネスプロセス、取引コラボレーション、文書スキーマなど）がワークシートに参照（再利用）またはコピーされ、必要に応じて変更される。時間の経過と共に、ビジネスプロセスライブラリが十分に多数のビジネスプロセスで満たされていく。その際多くの場合、分析プロセスは、要件に対して前もって定義されたビジネスプロセスを単に有効化するだけである。

Enablement:Analysis Worksheets and Editor

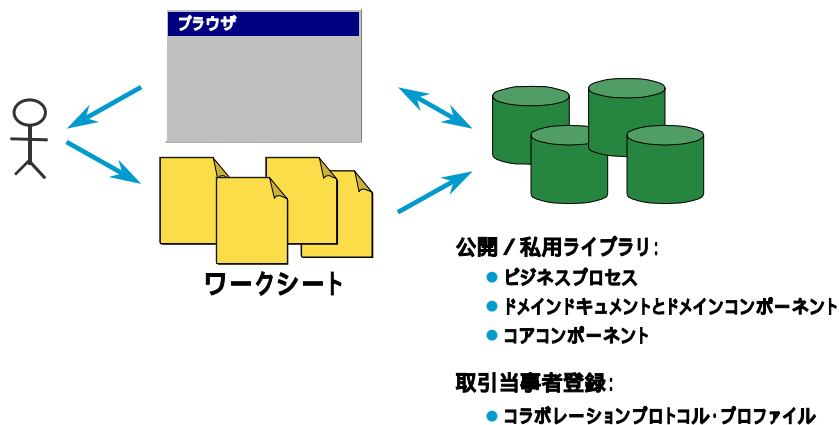


図 1.22.1-1 ビジネスプロセス分析ワークシートの使用

図 10.1-1 項目（ビジネスプロセス、取引コラボレーション、文書スキーマなど）がワークシートに参照（再利用）またはコピーされ、必要に応じて変更される。時間の経過と共に、ビジネスプロセスライブラリが十分に多数のビジネスプロセスで満たされていく。その際多くの場合、分析プロセスは、要件に対して前もって定義されたビジネスプロセスを単に有効化するだけである。

1.22.2 ビジネスプロセスエディタおよび文書エディタ

ビジネスプロセス分析ワークシート、ビジネスプロセスとコンポーネントのモデリングまたは分析の作成および保守は、ビジネスプロセスエディタ、文書コンポーネントエディタなどのアプリケーションツールによって、ビジネス専門家に優しい方法で提供される。これらのツールは、ビジネスライブラリおよび取引当事者のディレクトリに直接接続できるので、ビジネスプロセスおよび情報のモデリングの効果的な方法を提供する。図 10.2-1 を参照。このツールは、発見、ユーザーフレンドリなフォームベースのモデリング、ビジネスプロセスとビジネス情報の比較、分析プロセスの文書作成とヘルプ、ビジネスライブラリのコントローラに仕様を送信する機能をサポートしている。ビジネスプロセスエディタ、文書&コンポーネントエディタ、および CPP/CPA のツールスイートは、ebXML ベースの電子商取引の手段となる。

Business Process and Document Editor

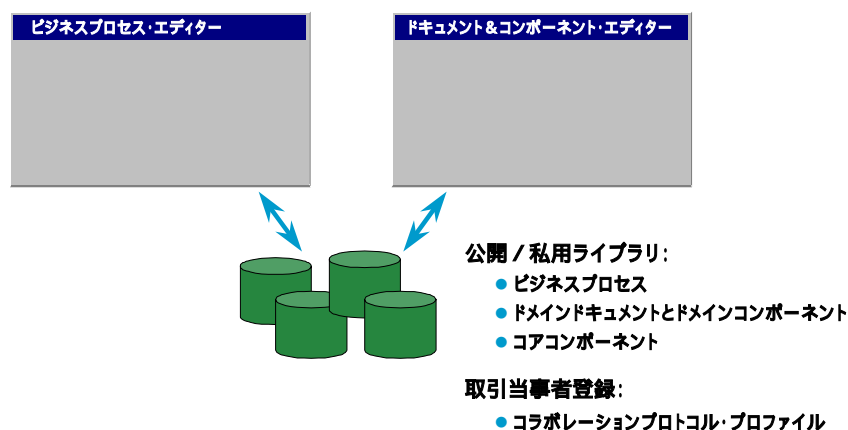


図 10.1.2-1 ツールの相互作用

参考資料

- [Bra97] Bradner, S., "Key words for use in RFCs to Indicate Requirement Level", BCP 14, RFC 2119, March 1997.
- [bpWS] ebXML Business Process Analysis Worksheets and Guidelines. V1.0, May 11 2001. ebXML Business Process Project Team.
- [ebBPSS] ebXML Business Process Specification Schema. Version 1.0 May 11 2001. *Context/Meta Model* Group of the CC/BP Joint Delivery Team.
- [ebPROC] ebXML Catalog of Common Business Processes. Version 1.0, May 11 2001. ebXML CC/BP Analysis Team.
- [bpPATT] ebXML Business Process and Simple Negotiation Patterns. Version 1.0, May 11 2001. ebXML Business Process Project Team.
- [IDEF0] Integration Definition For Function Modeling (IDEF0). Federal Information Processing Standards Publication 183.1993 December 21.
- [ISO14662] Information Technologies - Open-EDI Reference Model. ISO/IEC 14662:1997(E). International Organization for Standardization (ISO) and International Electrotechnical Commission (IEC).
- [ebCCD&A] ebXML Methodology for the Discovery and Analysis of Core Components. V1.0, May 11 2001. ebXML Core Components Project Team.
- [ebCNTXT] The role of context in the re-usability of Core Components and Business Processes. Version 1.0, May 11 2001. ebXML Core Components Project Team.
- [ebGLOSS] ebXML. TA Glossary. Version 1.0, May 11 2001 . Technical Architecture Project Team.
- [ebTA] ebXML Technical Architecture Specification. Version 1.0.4. 16 February 2001. ebXML Technical Architecture Project Team.
- [ebCCDOC] ebXML specification for the application of XML based assembly and context rules. Version 1.0, 11 May 2001. ebXML Core Components.
- [UMM] UN/CEFACT Modeling Methodology. CEFACT/TMWG/N090R9. February 2001. UN/CEFACT Technical Modeling Working Group.

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

Business Process Project チーム
Business Process/Core Components (BP/CC) Analysis チームリーダー

名前	Brian Hayes
所属	Commerce One
番地	4440 Rosewood Drive
州名、都市名など	Pleasanton, CA
国名	USA
電話	+1 (925) 788-6304
電子メール	brian.hayes@UCLAumni.net

編集

名前	Randy W. Clark
所属	Baker Hughes, Inc.
番地	3900 Essex Lane, Suite 800
州名、都市名など	Houston, TX 77027
電話	+1 (713) 439-8143
電子メール	randy.clark@bakerhughes.com

付録 A コンテキストカテゴリ - メタモデルの相互参照

次の表に、コア構成要素のコンテキストカテゴリとメタモデル要素との相互参照を示す。

コンテキストカテゴリ	定義	メタモデル要素	資源のソース	備考
業界	情報交換を行う業界または副業界。	BusinessOperationalMap.industry	UN/CEFACT など。	階層的な値 BOMによって、プロセスの集合の論理カテゴリが得られる。これらのプロセスは、複数の方法（スキーム）で、または業界を含む複数の視点から組織化できる。 ドメインと業界は同じではない。業界はドメインの一種であるが、ドメインは必ずしも特定の業界に対応していない。
ビジネスプロセス	情報交換によって実現するビジネスプロセス。	BusinessProcess	ebXML Catalog of Common Business Processes UN 業界クラス ロゼッタネット BPAWG（UN/Cefact プロセスグループ） ビジネスプロセスパターン	階層的な値。 標準分類を使用するかまたは独自の分類を定義する。メタモデルではこれが許されている。さまざまな業界フォーラムで定義されている可能性が高い。 商品の種類によって商品情報が異なる。
製品	情報交換によって記述または実現される商品またはサービス。	EconomicResource	UN/SPCP UN の一般的な分類およびドメインの一般的な分類。	階層的な値。 標準分類を使用するかまたは独自の分類を定義する。メタモデルではこれが許されている。さまざまな業界フォーラムで定義されている可能性が高い。 商品の種類によって商品情報

コンテキストカテゴリ	定義	メタモデル要素	資源のソース	備考
				が異なる。
地理的要素/ 条件/地域	地理的要素および条件（天候、標高、気候）、情報交換の地理的コンテキスト（地政学ではない）。	地理および地域カテゴリは、BOMのカテゴリスキーマによって定義できる。	GPS、航空、ISO	階層的な値。 条件の範囲は、カテゴリ要素に対する拘束条件として指定される。
時間	情報交換の時間ベースのコンテキスト。	EconomicCommitment.due	条件の多様性について評価される可能性のある条件付き表現。	階層的ではない。 日付の範囲でもよい。
地政学的な法律/取締/文化	政府による規制（通常は地理的要素ごとに定義される）および規制元。 注意：ビジネス会話への外的影響	地理および地域カテゴリは、BOMのカテゴリスキーマによって定義できる。	ATA, DOD, FAA, AECMA, UN/Cefact, ISO	階層的な値 – 高レベル（管区、州、または市のレベル）まででよい – 規制の主体は指定しない。
アプリケーション処理	情報交換のアプリケーション、システムコンテキスト、またはその両方。 サポートについては同意されたレベルがある。	ビジネスサービス	UN 経済アクティビティ、OAG、またはその両方（階層的な値）。（アプリケーション内のアプリケーション）。 - *広義*の「アプリケーション」。外的本体により自己登録される。	ベンダーおよび業界のサブ標準値をサポートしている。
ビジネスの目的/ドメイン	ビジネスプロセスには無関係なビジネス目的コンテキスト。ビジネス情報の送	BOM		ビジネスの目的およびドメインは、BOMカテゴリ化スキーマによって定義および範囲規定できる。

コンテキストカテゴリ	定義	メタモデル要素	資源のソース	備考
	付先の「目的」。			
当事者役割	当事者がプロセス内で果たす特定の役割。	当事者役割		階層的ではない。 商取引コラボレーションによって定義される。
サービスレベル(プロフィール-優先度情報ではない)	製品の提供者または荷受業者の契約書に付属しているサービスレベル。	契約書	OTA、クレジット代理店	階層的。
仮想市場	ビジネスを行う環境。	市場カテゴリは、BOMのカテゴリスキーマによって定義できる。		市場(マーケットプレイス)とコミュニティは同義語である。
情報構造コンテキスト	[XMLの意味での情報の「要素」コンテキスト]	取引文書、InformationEntity	自己参照、おそらく階層的。	
契約/契約書		契約書、商契約。		

著作権について

Copyright © UN/CEFACT and OASIS, 2001. All Rights Reserved

本書および本書の翻訳版は、上記の著作権通知およびこの段落を含めることを要件とし、自由にその一部または全部をコピーして配布したり、その解説や実施を支援する説明の作成、コピー、刊行、配布などを行ったりしてよい。ただし、英語以外の言語に翻訳する際に必要な場合を除き、著作権通知や ebXML、UN/CEFACT、OASIS などへの参照を取り除くなど、本書自体を変更することは一切してはならない。

上述の制約付き許可は永続的なものであり、ebXML やその継承者や譲受者によって破棄されることはない。本書および本書に含まれる情報は「無保証」で提供されており、ebXML は、明示、暗示の別を問わず、いかなる保証もしない。これには、本書の情報の使用が他の権利を侵害しないこと、暗示される商品性の保証、特定の目的の適合性などが含まれるが、これらに限定されない。

Copyright Statement

Copyright © ebXML 2001. All Rights Reserved.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this paragraph are included on all such copies and derivative works. However, this document itself may not be modified in any way, such as by removing the copyright notice or references to ebXML, UN/CEFACT, or OASIS, except as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by ebXML or its successors or assigns. This document and the information contained herein is provided on an "AS IS" basis and ebXML DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.



Creating A Single Global Electronic Market

ebXML ビジネスプロセス仕様スキーマ 改訂 1.01

ビジネスプロセスプロジェクトチーム

2001年5月11日

技術検証 ECOM XML/EDI 標準化専門委員会

1 本書の位置づけ

本技術仕様文書は ebXML 本会議において承認を受けており、ebXML 要件文書に記載された要件を満たしている。本書はインターネットソサエティの標準 RFC 形式に準拠している。

本バージョン:

www.ebxml.org/specs/ebBPSS.pdf

最新バージョン:

www.ebxml.org/specs/ebBPSS.pdf

2 ebXML ビジネスプロセス/コア構成要素メタモデル参加者

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。

チームリーダー :

Paul Levine, Telcordia

編集 :

Jim Clark, E2Open - previously Edifecs: (Transaction Semantics)
Cory Casanave, Data Access Technologies: (UML model)
Kurt Kanaskie, Lucent Technologies: (DTD and Examples)
Betty Harvey, Electronic Commerce Connection: (DTD documentation)
Jamie Clark, McLure-Moynihan, Inc.: (Legal aspects)
Neal Smith, Chevron (Issues Lists, and W3C schema)
John Yunker, Edifecs: (Signal structures)
Karsten Riemer, Sun Microsystems: (Overall Document)

協力 :

Antoine Lonjon, Mega

J.J. Dubray, Excelon
Bob Haugen, Logistical Software
Bill McCarthy, Michigan State University
Paul Levine, Telcordia
Brian Hayes, CommerceOne
Nita Sharma, Netfish
David Welsh, Nordstrom
Christopher Ferris, Sun Microsystems
Antonio Carrasco, Data Access Technologies

3 目次

1 本書の位置づけ	1
2 ebXML ビジネスプロセス/コア構成要素メタモデル参加者	1
3 目次	1
4 はじめに	4
4.1 対象読者	4
4.2 関連文書	5

4.3	前提条件	5
5	設計目標	5
5.1	目的 / 目標 / 要件 / 問題記述	5
5.2	警告と前提条件	6
5.2.1	ebXML ビジネスプロセス仕様スキーマと UMM の関係	6
6	システム概要	9
	ビジネスプロセス仕様スキーマの UML バージョン	11
	生成規則	13
6.1	ebXML ビジネスプロセス仕様スキーマの重要概念	14
6.2	ebXML ビジネスプロセス仕様スキーマ の利用方法	17
6.3	ebXML ビジネスプロセス仕様スキーマと他の ebXML 仕様との関わり	17
6.4	設計時に再利用しながらコラボレーションとトランザクションを設計する方法	19
6.4.1	取引トランザクションとその取引文書フローを指定する	20
6.4.2	バイナリコラボレーションを指定する	25
6.4.3	複数当事者コラボレーションを指定する	29
6.4.4	振り付けを指定する	31
6.4.5	全体モデル	34
6.5	コア取引トランザクション意味情報	36
6.5.1	相互作用の予測の可能性	36
6.5.2	法的拘束力を持つ契約の作成	39
6.5.3	否認不可	40
6.5.4	認定セキュリティ	41
6.5.5	文書セキュリティ	42
6.5.6	信頼性	43
6.5.7	CPP/CPA の必須パラメータ	43
6.6	実行時トランザクションの意味情報	44
6.6.1	タイムアウト	44
6.6.2	例外	46
6.7	実行時コラボレーション意味情報	48
6.8	ebXML ビジネスプロセス仕様スキーマが実装される状況	49
7	UML 要素指定	49
7.1	取引コラボレーション	49
7.1.1	MultiPartyCollaboration	49
7.1.2	BusinessPartnerRole	50
7.1.3	Performs	50
7.1.4	AuthorizedRole	51
7.1.5	BinaryCollaboration	51
7.1.6	業務アクティビティ	53
7.1.7	BusinessTransactionActivity	53
7.1.8	CollaborationActivity	54
7.2	取引トランザクション	55

7.2.1	BusinessTransaction	55
7.2.2	BusinessActivity	56
7.2.3	Requesting BusinessActivity	57
7.2.4	Responding BusinessActivity	57
7.3	文書フロー	58
7.3.1	文書セキュリティ	58
7.3.2	文書エンベロープ	59
7.3.3	BusinessDocument	60
7.3.4	Attachment	60
7.4	コラボレーションの中の振り付け	61
7.4.1	BusinessState	61
7.4.2	Transition	61
7.4.3	Start	62
7.4.4	TerminalState	63
7.4.5	Success	63
7.4.6	Failure	64
7.4.7	Fork	64
7.4.8	Join	64
7.5	定義と範囲	65
7.6	コラボレーション指定規則	65
8	ebXML ビジネスプロセス仕様スキーマ – (DTD)	68
8.1	DTD 用文書	68
8.2	XML から UML への相互参照	99
8.3	有範囲名前参照	101
8.4	置換セット	102
8.5	上記 DTD に照らし合わせたサンプル XML 文書	102
9	ビジネスシグナルの構造	103
9.1.1	ReceiptAcknowledgment の DTD	103
9.1.2	AcceptanceAcknowledgement の DTD	105
9.1.3	例外シグナルの DTD	107
10	生成規則	108
	付録 A : XML Business Process Specification のサンプル	110
	付録 B : ビジネスプロセス仕様スキーマの DTD	114
	付録 C : ビジネスプロセス仕様スキーマの XML	120
11	References	131
12	免責	131
13	連絡先	132
	著作権について	133

4 はじめに

総括

『ebXML ビジネスプロセス仕様スキーマ』は取引トランザクションを構成している取引コラボレーションの実行を支援するためにビジネスシステムを構成する標準フレームワークを提供する。これは、すでに行われた UN/CEFACT 作業に基づいている。具体的には N090R9.1 仕様が定める UN/CEFACT モデリング手法 (UMM) のメタモデルに基づいている。

この仕様スキーマは、取引トランザクションの仕様と、取引トランザクションから取引コラボレーションへの振り付けを支援する。取引トランザクションは、多くの標準パターンを使って実装できる。これらのパターンを使用して、必要な電子商取引トランザクションを達成し、取引当事者間で行われるメッセージとビジネスシグナルの具体的なやり取りを決定する。

仕様スキーマの現行版は、2 者間のコラボレーションを扱う (バイナリコラボレーション)。

今後の版では、経済交換・契約の意味情報、より複雑な複数当事者振り付け、コンテキストに基づく内容など、追加の要素を扱うことが見込まれる。本書の概要本書では、『ebXML ビジネスプロセス仕様スキーマ』について論じる。

本書では、UML 形式と DTD 形式の両方で仕様スキーマを論じる。

まず最初に一般的概念および意味情報の概説をしてから、これらの意味情報のモデルの各部分への適用に関して詳細に説明する。次に、UML 形式のすべての要素を指定してから、XML 形式のすべての要素を特定する。

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。

4.1 対象読者

主な対象読者はビジネスプロセスの分析者である。本書で定義するビジネスプロセス分析者とは、実業者と面談したうえで、ビジネスプロセスを明確な構文で成文化する者である。

このほかに、ビジネスプロセス定義ツールにおけるユーザ入力から仕様スキーマの XML 表現への変換を指定するビジネスプロセス定義ツールの設計者も対象読者である。

ビジネスアプリケーション開発者は対象読者ではない。

4.2 関連文書

上述の通り、『ebXML ビジネスプロセス仕様スキーマ』の構成要素とその相互関係を詳しく定義する文書が他にある。それらの『ebXML 仕様』は以下の論題を扱っている：

- ebXML Technical Architecture Specification, version 1.04
- ebXML Core Components Dictionary, version 1.04
- ebXML Naming Convention for Core Components, version 1.04
- ebXML Collaboration-Protocol Profile and Agreement Specification V1.0
- ebXML Business Process and Business Information Analysis Overview, version 0.7
- ebXML Business Process Analysis Worksheets & Guidelines, version 0.10
- ebXML E-Commerce Patterns, version 0.99
- ebXML Catalog of Common Business Processes, version 0.99
- ebXML メッセージ取扱サービス Specification V0.99
- UN/CEFACT Modeling Methodology (UMM) as defined in the N090R9.1 specification

4.3 前提条件

対象読者としては、下記のテクノロジーとテクニックをすでに理解している者、もしくは、今後精通する者が想定される。

- ビジネスプロセスのモデリング技術・原理
- UML の構文と意味情報
- 拡張マークアップ言語 (XML)

5 設計目標

5.1 目的 / 目標 / 要件 / 問題記述

ビジネスプロセスモデルは、取引当事者のコラボレーションを可能にする相互運用可能なビジネスプロセスを明示するものである。電子ビジネスのためのビジネスプロセスモデルは、取引当事者に代替するソフトウェアコンポーネントの形に還元されなければならない。

『ebXML ビジネスプロセス仕様スキーマ』の目的は、電子ビジネスプロセスモデリングと電子ビジネスソフトウェアコンポーネント仕様を橋渡しをすることである。

『ebXML ビジネスプロセス仕様スキーマ』は、電子ビジネスソフトウェアコンポーネント間のコラボレーションを実現するために、取引当事者間のコラボレーションを指定し、取引当事者の実行時システムに設定パラメータを提供するのに必要な仕様要素を用意している。

ebXML ビジネスプロセス仕様スキーマを使って作成された仕様は、『ebXML ビジネスプロセス仕様スキーマ』と呼ばれる。

『ebXML ビジネスプロセス仕様スキーマ』は、UML バージョンおよび XML バージョンの 2 つの独立した形式で提供されている。

『ebXML ビジネスプロセス仕様スキーマ』の UML バージョンは UML クラス図のみを提供しているため、ebXML ビジネスプロセス仕様の作成に直接使用することはできない。このバージョンはむしろ、ebXML 準拠のビジネスプロセス仕様の作成に必要な仕様要素とそれらの関係すべてを含んだ独立した一覧である。ebXML に準拠したビジネスプロセス仕様の作成に使用される手法またはメタモデル、あるいはその両方は、これらの要素および関係を最低限サポートしていなければならない。

『ebXML ビジネスプロセス仕様スキーマ』の XML バージョンは、ebXML ビジネスプロセス仕様の XML をベースにしたインスタンスの仕様を、他の記述方法に対する生成規則の目標として提供している。DTD および W3C スキーマの両方が提供されている。

『ebXML ビジネスプロセス仕様スキーマ』の UML ベースのバージョンおよび XML ベースのバージョンの両方は、明確な形で相互にマッピングされている。

5.2 警告と前提条件

本書では、取引コラボレーションの実行時の側面を明記する。

本書において手法を編入するつもりはなく、手法の使用を直接的に規定することはしない。ただし手法を用いる場合は、UN/CEFACT モデリング手法 (UMM) が推奨される。

『ebXML ビジネスプロセス仕様スキーマ』そのものが「取引文書構造」を定義することはない。既存の「取引文書」定義や、「ebXML コア構成要素」仕様に含まれる「取引文書」メタモデルとともに機能することが意図される。

5.2.1 ebXML ビジネスプロセス仕様スキーマと UMM の関係

UN/CEFACT モデリング手法(UMM)とは、ビジネスプロセスおよび情報モデリングに使用する手法である。

本節では、UMM および『ebXML ビジネスプロセス仕様スキーマ』の関係を解説する。

UMM メタモデルはビジネス意味情報の記述であり、この記述を通じて「トランザクション取引当事者」は、それぞれのビジネスシナリオの中で一貫性のあるモデリング手法を用いてビジネスプロセスを具体化できる。ビジネスプロセスは、「トランザクション取引当事者」がいかにして割り当てられた役割と関係と責任を引き受けながら相手方とのやり取りを行うかを詳しく記述している。役割間の相互作用は、振り付けされた取引トランザクション群として発生する。各々の取引トランザクションは、電子「取引文書」の交換で表わされる。交換の順序はビジネスプロセスによって決まり、さらにメッセージングやセキュリティも配慮している。「取引文書」は、再利用可能なビジネス情報構成要素から生成される。より低いレベルでは、ビジネスプロセスを再利用可能な「共通ビジネスプロセス」で構成し、「ビジネス情報オブジェクト」を再利用可能なコア構成要素で構成することが可能である。共通ビジネスプロセスとビジネス情報オブジェクトは UMM ビジネスライブラリに格納されている。

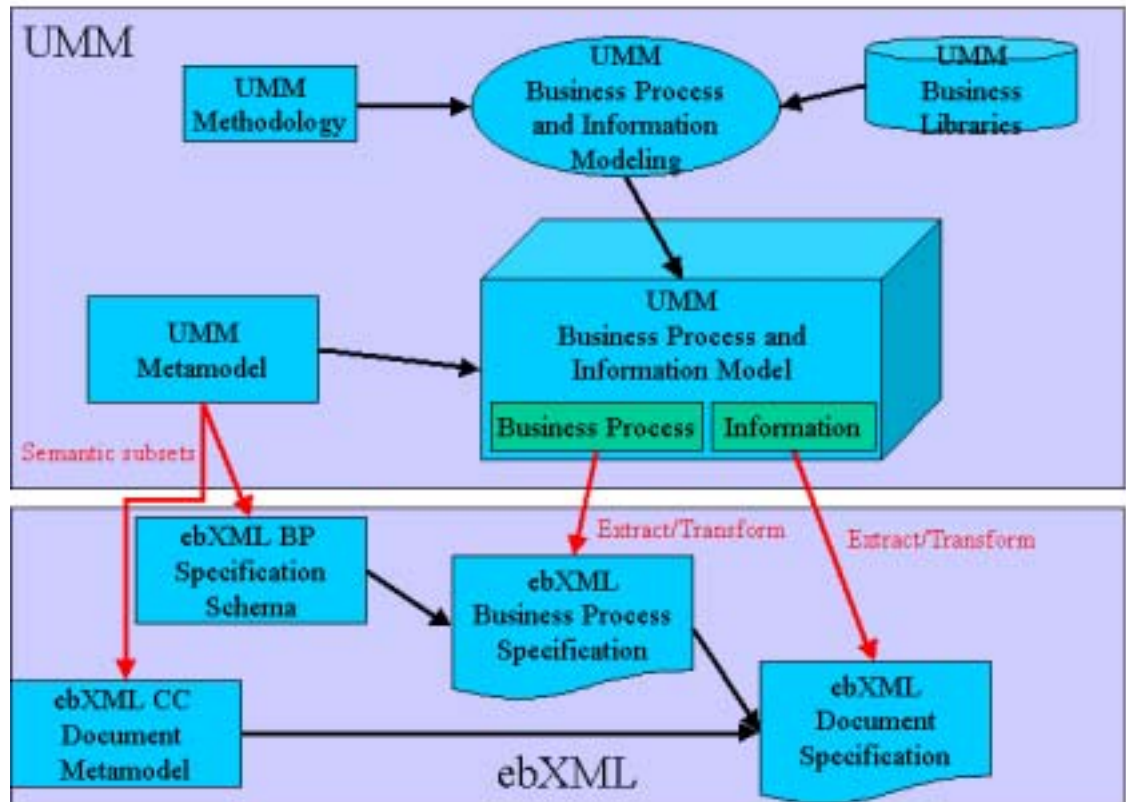
UMM メタモデルでは、各ビューポイントに対して一連の意味情報（語彙）を提供するビジネスプロセスのビューポイントがサポートされている。これらのビューポイント群は、ビジネスプロセスおよび情報の統合と相互運用性のために必要な意味情報と生成物の仕様の基盤となる。ユーザは、UMM 手法と UMM メタモデルを用いて、完全なビジネスプロセスと情報モデルを作成することができる。このモデルには、ebXML 準拠のソフトウェアの設定に必要なとされる情報が含まれている。また、このモデルは構文から独立しており、ebXML 準拠のソフトウェアから直接読み取ることはできない。

『ebXML ビジネスプロセス仕様スキーマ』では、UMM メタモデルのもう 1 つのビューも提供されている。このサブセットにより、一連の名目要素の直接的仕様がサポートされる。これらの要素は、ebXML 取引トランザクションを実行するシステムの設定には必要不可欠である。他の複数のビューをモデリング要素と結びつけることによって、『ebXML ビジネスプロセス仕様スキーマ』は、UMM メタモデルの意味情報サブセットを形成する。このように『ebXML ビジネスプロセス仕様スキーマ』を用いることで、ユーザは、ebXML 準拠のソフトウェアの設定に必要な情報のみを含むビジネスプロセス仕様を作成することができる。

『ebXML ビジネスプロセス仕様スキーマ』は、UML バージョンおよび XML バージョンという 2 つの独立した形式で提供されている。XML バージョンは ebXML 準拠のソフトウェアで読み取ることが可能である。

UMM メタモデルと『ebXML ビジネスプロセス仕様スキーマ』の関係を以下の図 1 に示す。

図 1. UMM メタモデルと『ebXML ビジネスプロセス仕様スキーマ』



UMM 手法と UMM ビジネスライブラリの内容を用いて、ユーザは、UMM メタモデルに準拠した完全なビジネスプロセスおよび情報モデルを作成することができる。

『ebXML ビジネスプロセス仕様スキーマ』は UMM メタモデルの意味情報のサブセットであるため、ユーザは自動的に、必要とされる一連の要素および関係をビジネスプロセスおよび情報モデルから抽出し、それらを『ebXML ビジネスプロセス仕様スキーマ』に準拠した ebXML ビジネスプロセス仕様に変換することができる。

これと同様に、ebXML CC 文書メタモデルは UMM メタモデルと密接な関係にあるため、ユーザは自動的に、必要とされる要素および関係をビジネスプロセスおよび情報モデルから抽出し、それらを ebXML コア構成要素の仕様に準拠した ebXML 文書モデルに変換することができる。

UMM 手法は、正式な ebXML 仕様に含まれていない。

これと同様に、UMM メタモデルはすべて、正式な ebXML 仕様に含まれない。『ebXML ビジネスプロセス仕様スキーマ』およびコア・コンポーネントによって示される意味情報のサブセットのみが、正式な ebXML 仕様に含まれる。

本書の後半では、『ebXML ビジネスプロセス仕様スキーマ』および、それを基にして作成されたビジネスプロセス仕様を中心に解説する。適切なビジネスプロセスおよび情報モデリングが、ビジネスプロセス仕様の作成開始より以前に作成される場合もある。

6 システム概要

『ebXML ビジネスプロセス仕様スキーマ』は、ビジネスプロセス仕様の標準的なフレームワークを提供する。このスキーマを、ebXML コラボレーションプロトコルプロファイル(CPP)仕様およびコラボレーションプロトコル合意書 (CPA)仕様と併用することで、ビジネスプロセスモデリングと ebXML に準拠した電子商取引ソフトウェア(例：図 2 で示すような取引サービスインタフェース)の設定を橋渡しする。

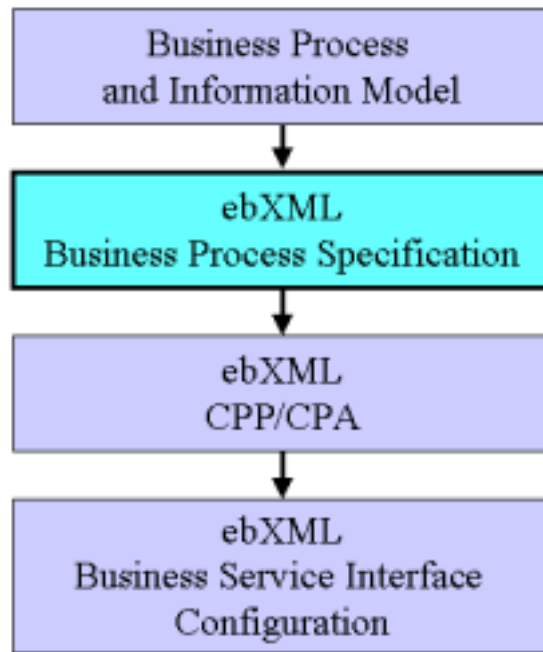


図 2：ビジネスプロセス仕様および取引サービスインタフェースの設定

ビジネスプロセスモデリングを用いて、ユーザは、完全なビジネスプロセスおよび情報モデルを作成することができる。

このようなビジネスプロセスおよび情報モデルを基にして、ユーザは『ebXML ビジネスプロセス仕様スキーマ』を用いて、一連の ebXML 取引トランザクションを実行するために ebXML 実行時システムの設定に必要な名目上の一連の要素を抽出、フォーマットすることができる。この結果、ebXML ビジネスプロセス仕様が得られる。

また、ebXML ビジネスプロセス仕様を、ビジネスプロセスモデリングを行わずに直接作成する方法もある。

ebXML ビジネスプロセス仕様には、取引トランザクションの仕様、および取引トランザクションの取引コラボレーションへの振り付けが含まれる。

この ebXML ビジネスプロセス仕様は、ebXML 取引当事者の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書(CPA)の形成に利用される。

次に、ebXML 取引当事者の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書は、ebXML 取引サービスインタフェース用ソフトウェアの設定ファイルとして使用される。

『ebXML ビジネスプロセス仕様スキーマ』のアーキテクチャは、以下に示す機能コンポーネントで構成される。

- 『ビジネスプロセス仕様スキーマ』の UML バージョン
- 『ビジネスプロセス仕様スキーマ』の XML バージョン
- 『ビジネスプロセス仕様スキーマ』の UML バージョンから XML バージョンへのマッピングを定義する生成規則
- ビジネスシグナルの定義

これらのコンポーネントを併用することで、ビジネスプロセスモデルの実行時の設定事項をすべて指定することができる。

これらのコンポーネントを以下の図 3(点線の枠内)で示す。

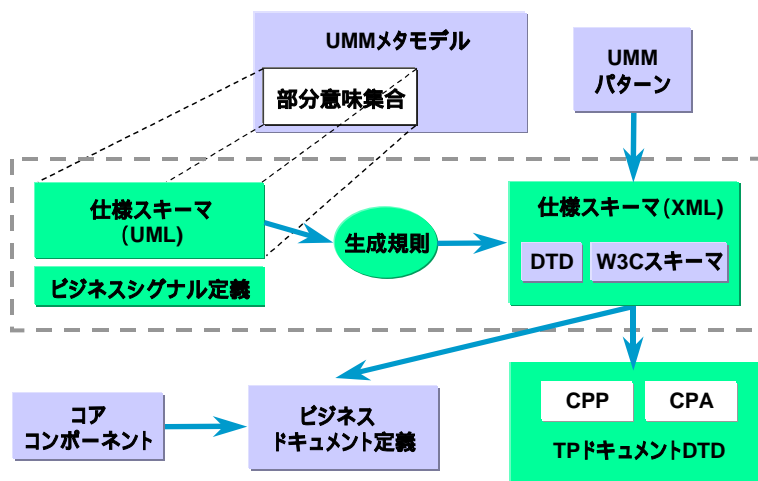


図 3：『ebXML ビジネスプロセス仕様スキーマ』の UMM、CPP/CPA およびコア構成要素に対する関係

『ebXML ビジネスプロセス仕様スキーマ』の各コンポーネントの説明と、それらの UMM、ebXML CC、および CPP/CPA に対する関係の説明を次に行う。

ビジネスプロセス仕様スキーマの UML バージョン

『ebXML ビジネスプロセス仕様スキーマ』の UML バージョンは、UMM の背後で使用されるメタモデルの意味情報のサブセットであり、UN/CEFACT TMWG の N090R9.1 においてもそのように指定されている。

N090R9.1 は、本書の作成時において、まだ UN/CEFACT によって承認を受けていない。そのため、『ebXML ビジネスプロセス仕様スキーマ』と TMWG の N090 を意味情報的に密接に関連付けることが意図されている。

『ebXML ビジネスプロセス仕様スキーマ』の UML バージョンは UML クラス図のみを提供しているため、ebXML ビジネスプロセス仕様の作成に直接使用することはできない。このバージョンはむしろ、ebXML 準拠のビジネスプロセス仕様の作成に必要な仕様要素とそれらの関係すべてを含んだ独立した一覧である。

ビジネスプロセス仕様スキーマのXMLバージョン

『ebXML ビジネスプロセス仕様スキーマ』のXMLバージョンは、ebXML ビジネスプロセス仕様のXMLベースのインスタンスの仕様を、他の記述方法に使用される生成規則の目標として提供している。このようにして、ユーザは、ビジネスプロセス仕様をXML文書として直接作成するか、または最初に別の仕様指定方法を用いてから、生成規則に従ってXML文書のバージョンが使用できる。

ebXML 準拠のビジネスプロセス仕様の作成に使用する手法またはメタモデル、あるいはその両方は、『ebXML ビジネスプロセス仕様スキーマ』のXMLバージョンに含まれる要素および関係の作成を最低限サポートしていなければならない。

DTD および W3C スキーマの両方が提供されており、それぞれは『ebXML ビジネスプロセス仕様スキーマ』のUMLバージョンと同一構造の定義である。

UMM ビジネスプロセス相互作用のパターン

ebXML 取引サービスインタフェースは、ビジネスプロセス仕様で指定されたビジネスプロセスを実行するように設定されている。ビジネスプロセスの実行は、ebXML メッセージとビジネスシグナルを交換することで行われる。

各取引トランザクションは、使用可能な多くの標準パターンのうちの1つを使って実装することができる。これらのパターンにより、取引当事者間でのメッセージとビジネスシグナルの実際の交換方法が決定され、要求された電子商取引トランザクションを実行することができる。

第8章の UMM N090R9.1 文書において言及されている取引トランザクション相互作用パターンでは、推奨されたメッセージシーケンスが説明されている。このメッセージシーケンスは、定義された取引トランザクションのタイプおよびトランザクション内で指定されたタイミングポリシーによって決定される。

UMM パターン自身は ebXML 仕様には含まれないが、パターンのプロパティの表示に必要なセキュリティおよびタイミングのパラメータすべては、『ebXML ビジネスプロセス仕様スキーマ』における要素の属性として提供されている。

ビジネスシグナルの定義

ビジネスシグナルは、取引トランザクションの最新の状態を‘知らせる’アプリケーションレベルの文書である。これらのビジネスシグナルは特定のビジネス目的を持っており、低レベルのプロトコルおよびトランスポートシグナルとは別のものである。

しかし、ebXML ビジネスシグナルの構造は‘一般的’であるため、トランザクションごとに変化しない。そのためビジネスシグナルは『ebXML ビジネスプロセス仕様スキーマ』の一部として1度定義するだけで良い。

『ebXML ビジネスプロセス仕様スキーマ』は、ビジネスシグナルの振り付けと、ビジネスシグナルのビジネス搬送内容の構造定義の両方を提供する。ebXML メッセージ取扱サービス仕様のシグナル構造は、取引サービス状態の関連付け基盤を提供する。この基盤には、基本的なプロセス関連付けの要件を満たすために使用される独自のメッセージ識別子およびダイジェストが含まれる。ここで提示されるビジネスシグナルの搬送内容構造は、オプションおよび規準であり、ビジネスシグナルに対してビジネス意味情報および正しい意味情報を提供することを目的とする。

使用可能な各ビジネスシグナルにつき、1つの DTD が提供される。

生成規則

『ebXML ビジネスプロセス仕様スキーマ』の UML バージョンの XML バージョンへのマッピングを定義するために、一連の生成規則が提供されている。

これらの生成規則を使用する目的は、『ebXML ビジネスプロセス仕様スキーマ』の UML クラス図から DTD を生成することである。

『ビジネスプロセス仕様スキーマ』のクラス図は、ebXML ビジネスプロセス仕様を直接作成するために使用することはできない。しかし、ビジネスプロセス仕様がこのクラス図のインスタンスとして(プログラムを使って)実際に作成された場合には、この仕様を DTD に準拠した XML 文書に変換する際にも生成規則が適用される。

これとは別に、UML 図を基にした、『ebXML ビジネスプロセス仕様スキーマ』の XML の作成に使用される生成規則が構築される。

『ebXML ビジネスプロセス仕様スキーマ』の UML クラス図の1つのインスタンスからは、該当の生成規則を適用することで、XML 仕様文書が生成される。この XML 仕様文書は、同じサブセットを UMM およびそれに関連する生成規則を使ってモデリングした場合に得られる文書と、分析的に見ても、そして意味情報および機能の点で同等である。

CPP/CPA との関係

ビジネスプロセス仕様は、ebXML 取引サービスインタフェースに必要とされる、機械による解釈が可能な実行時ビジネスプロセス仕様である。そのためビジネスプロセス仕様は、ebXML 取引当事者の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)に統合されたり、あるいは CPP および CPA に参照される。各 CPP はビジネスプロセス仕様内の1つ以上の役割をサポートすることを宣言する。次に、これらの CPP プロファイルおよび CPA 合意内に技術的パラメータをさらに追加することで、各取引当事者に対する実行時ソフトウェアの完全な仕様が完成される。

ebXML コア構成要素との関係

『ビジネスプロセス仕様スキーマ』自身が取引文書定義をサポートしているわけではなく、ビジネスプロセス仕様が取引文書定義を単に指しているだけである。このような文書定義の構造は、ebXML ベース、アタッチメント、その他の構造の場合や、または全く構造を持たない場合もある。XML ベースの取引文書仕様は、ebXML コア構成要素仕様に基づいている場合もある。

ebXML メッセージ取扱サービス仕様との関係

『ビジネスプロセス仕様スキーマ』は、ビジネスメッセージおよびビジネスシグナルの振り付けを行う。ebXML メッセージ取扱サービス仕様は、振り付け段階において、あらゆるメッセージを他のメッセージとの関連を考慮して順に配置するとともに、メッセージおよびシグナルの識別、タイピング、および統合のための基盤を提供する。

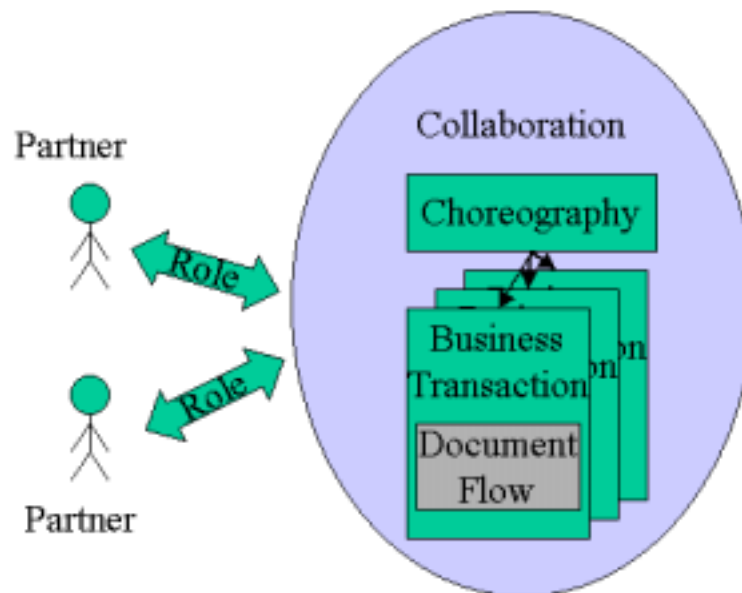
6.1 ebXML ビジネスプロセス仕様スキーマの重要概念

『ebXML ビジネスプロセス仕様スキーマ』は、取引コラボレーションを定義するための意味情報と要素とプロパティを提供する。

取引コラボレーションを構成する役割群は、振り付けされたトランザクションを通じて取引文書をやり取りしながら協力する。

取引コラボレーションの基本意味情報は図 4 に示す通りである。

図 4 : 取引コラボレーションの基本意味情報



複数の取引当事者が役割を通じて取引コラボレーションに関与する。役割は、取引トランザクションを通じて互いに働きかける。「振り付け」は、取引トランザクションに相対的な順序を与える。各取引トランザクションは、あらかじめ定義された1つまたは2つの「取引文書フロー」からなる。1つの取引トランザクションは、さらに1つ以上のビジネスシグナルによってサポートされる場合もある。

以下の節では、取引コラボレーション、取引トランザクション、「取引文書フロー」、および「振り付け」の概念を論じる。

1. 取引コラボレーション

取引コラボレーションとは、取引当事者間で行われる取引トランザクション群である。コラボレーションの中では、各取引当事者が1つまたは複数の役割を果たす。

『ebXML ビジネスプロセス仕様スキーマ』は、バイナリコラボレーションと複数当事者コラボレーションという2つの水準の取引コラボレーションをサポートする。

バイナリコラボレーションは、2つの役割の間で行われる。

複数の当事者コラボレーションは3つ以上の役割の間で行われるが、複数当事者コラボレーションは常に2つ以上のバイナリコラボレーションから合成される。たとえば、A、B、およびCの役割が協力し、すべての当事者が互いに働きかける場合には、AとBの間に単独のバイナリコラボレーションが存在し、BとCの間、AとCの間にも単独のバイナリコラボレーションが存在する。この複数当事者コラボレーションは、3つのバイナリコラボレーションを合成している。

バイナリコラボレーションは、2つの役割間の業務アクティビティ群として表わされる。各業務アクティビティはコラボレーションの中の一状態を反映する。「業務アクティビティ」には、'原子的'なもの、すなわち原子的取引トランザクションを遂行するアクティビティが、'複合的'なもの、すなわち別のバイナリコラボレーションを遂行するアクティビティがある。いずれにせよアクティビティは、以下のとおりに振り付けることができる。

1つのバイナリコラボレーションの中のアクティビティとして、実質的に他のバイナリコラボレーションを実行できることは、バイナリコラボレーションの再帰的組成とバイナリコラボレーションの再利用にとって重要である。

バイナリコラボレーションは本質的に、2役割間における再利用可能なプロトコルである。

2. 取引トランザクション

取引トランザクションは、2取引当事者間のトランザクション取決めにおけるタスクの原子的単位である。取引トランザクションは、トランザ

クションの中で対極の役割を果たす 2 者間で行われる。それらの役割は常に、要求する役割と応答する役割である。

「バイナリコラボレーション」と同様、取引トランザクションは 2 役割間における再利用可能なプロトコルである。取引トランザクションを再利用するときには、「バイナリコラボレーション」から取引トランザクションを参照する。基本的には、取引トランザクションの実行に「バイナリコラボレーション」の役割が割り当てられる。

ただし「バイナリコラボレーション」と違って取引トランザクションは原子的であり、より低位の取引トランザクションに分解することはできない。

厳密で強制可能なトランザクション意味情報を達成するため、取引トランザクションは特化した拘束されるプロトコルである。この意味情報は、トランザクションを管理するソフトウェア、すなわち「取引サービスインタフェース」(BSI)によって強制される場合がある。

取引トランザクションは常に成功か失敗のいずれかになる。成功の取引トランザクションは、2 当事者にとって法的拘束力を持つ場合がある。失敗の取引トランザクションは無意味かつ無効であり、各取引当事者は、そのトランザクションによって成立した相互要求を放棄しなければならない。これは、失敗時にトランザクションを「復帰」(ロールバック)させる行為と見なすことができる。

3. 取引文書フロー

取引トランザクションは、要求アクティビティと応答アクティビティとの間の「取引文書フロー」として実現する。要求「取引文書」は常に存在し、2 方向対話の対極にあたる 1 方向通知のように、所望のトランザクション意味情報に応じてオプションとして、応答「取引文書」が存在する。

実際の文書定義は ebXML コア構成要素仕様を使って定義するか、もしくは ebXML 以外の何らかの方法を用いるが、結果としては、仕様スキーマインスタンスから指示できる DTD が「スキーマ」が導かれる。

4. 振り付け

「取引トランザクション振り付け」は、取引トランザクション間か 1 バイナリコラボレーション内のサブコラボレーション間における配列と遷移を記述する。UML ツールでは、UML アクティビティ図を使ってこれを行う。仕様スキーマでは、開始状態、終了状態、アクティビティ、同期、アクティビティ間の遷移、遷移のガードなど、アクティビティ図の概念を使って振り付けを記述する。

5. パターン

仕様スキーマは、トランザクションとコラボレーションを指定するための明確な意味情報群を提供する。これらの意味情報を利用するコミュニティは、多様なトランザクションとコラボレーションを柔軟に指定でき

る。パターンを利用すれば、この柔軟性に一貫性が加わり、迅速な設計・実装が促進され、包括的な処理が可能となる。

前もって定義されたトランザクション相互作用のパターンは、UMMに含まれるトランザクション相互作用パラメータ設定の一般的な組み合わせを定義する。

UMM トランザクション相互作用パターンは ebXML 仕様には含まれないが、パターンのプロパティの表示に必要なセキュリティおよびタイミングのパラメータすべてが、『ビジネスプロセス仕様スキーマ』の要素の属性として提供されている。

また、コラボレーションの振り付けのためのパターンが将来作成されることが期待されている。このようなパターンの一例が、「ebXML 電子商取引パターン」である。

再利用、反復、およびパターンなどが、『ebXML ビジネスプロセス仕様スキーマ』の主な概念である。次の節でこれらの概念について詳細に論じる。

6.2 ebXML ビジネスプロセス仕様スキーマ の利用方法

取引コラボレーションに向けてソフトウェアを指定するときは必ず、『ebXML ビジネスプロセス仕様スキーマ』を使用する必要がある。

このようなソフトウェアは一般的に「取引サービスインタフェース(BSI)」と呼ばれる。

『ebXML ビジネスプロセス仕様スキーマ』を使用すると、これらのコラボレーション実行のための BSI 設定に使用する設定パラメータに関連するビジネスプロセスを指定することができる。

本節では次のことについて論じる

- 『ebXML ビジネスプロセス仕様スキーマ』と他の ebXML 仕様との関わり。
- まったく新しいコラボレーションとトランザクションを指定する、あるいは既存のコラボレーションとトランザクションを再利用する際の、設計時に『ebXML ビジネスプロセス仕様スキーマ』を利用する方法。
- 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書(CPP/CPA)に必要であるコアトランザクションの意味情報およびパラメータの指定方法。
- 『ebXML ビジネスプロセス仕様スキーマ』によって指定され、「取引サービスインタフェース」(BSI)によって管理されることが見込まれる、実行時のトランザクションおよびコラボレーションの意味情報。

6.3 ebXML ビジネスプロセス仕様スキーマと他の ebXML 仕様との関わり

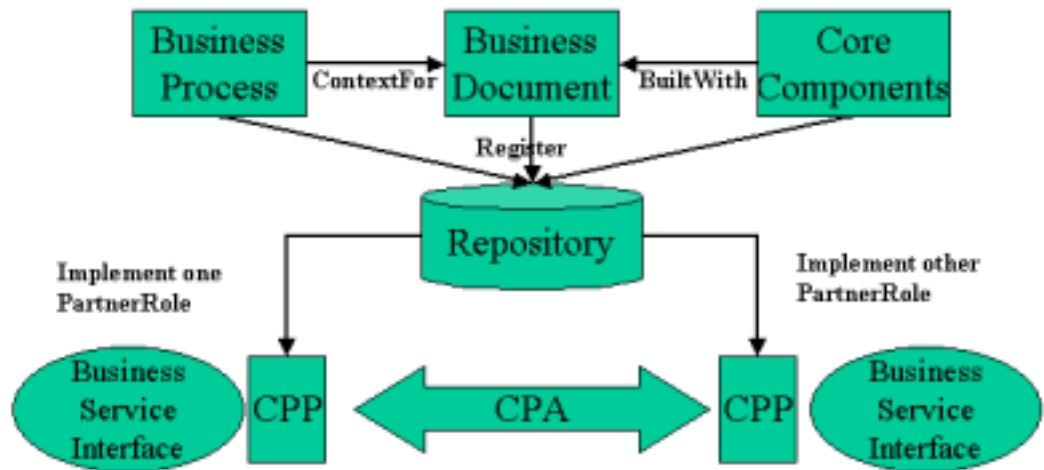
『ebXML ビジネスプロセス仕様スキーマ』は、取引コラボレーションを定義するための意味情報と要素とプロパティを提供する。

コラボレーションを構成する役割は、振り付けされたトランザクションを通じて取引文書をやり取りを行う。

取引文書は、図5で示すように、ビジネスプロセス仕様および ebXML コア構成要素仕様の両方を使用して定義される。ビジネスプロセス仕様は、一連の要求された取引文書を参照するが、定義はしない。ebXML 取引文書は、外部文書仕様によって定義されるか、あるいはコア構成要素と呼ばれる低いレベルの情報構造から直接または間接的にアセンブリされる。このアセンブリは一連のコンテキストに基づいて実行される。これらのコンテキストの多くは、例えば文書フローで該当の文書を使用するコラボレーションのようなビジネスプロセス群から取得できる。

ビジネスプロセスモデルと情報モデルの組み合わせは、取引当事者が電子ビジネスの進め方について取決めを行う際の基礎となる。

図5：『ebXML ビジネスプロセス仕様スキーマ』とその他の ebXML 仕様



ユーザは、必要情報を既存のビジネスプロセスおよび情報モデルから取得し、変換する。関連した生成規則を用いて、ビジネスプロセス仕様の XML バージョンの作成を支援することができる。

また、ユーザは XML ベースのツールを利用して XML 版をダイレクトに作ることもできる。そして、生成規則を用いて XMI への変換を行い、UML ツールに適宜ロードできる状態にする。

どちらの方法を使用した場合も、ビジネスプロセス仕様の XML バージョンは ebXML リポジトリに格納され、今後の抽出が可能になるように ebXML レジストリに登録される。ビジネスプロセス仕様の登録には、この設計時に派生したクラス識別子が使用される。

実装者が取引当事者の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書を作成する場合、XML 文書またはその一部が、CPP および CPA XML 文書に埋め込まれるか、あるいは CPP および CPA XML 文書によって参照される。ebXML CPP および CPA 文書は、ebXML ビジネスプロセス仕様の XML バージョンのみを参照することができる。

CPP/CPA 仕様に沿ってできあがった XML 文書は、「取引サービスインタフェース」(BSI)、すなわち取引当事者によるコラボレーションへの関与を実際に管理するソフトウェアのための設定ファイルとなる。

6.4 設計時に再利用しながらコラボレーションとトランザクションを設計する方法

本節では、完全な「複数当事者コラボレーション」を下記のとおりに一から構築する方法を示すことによって、仕様スキーマのモデリング関係を説明する。

1. 取引トランザクションを指定する
2. 取引トランザクションの「文書フロー」を指定する
3. 取引トランザクションを再利用しながら「バイナリコラボレーション」を指定する
4. 「バイナリコラボレーション」の振り付けを指定する
5. 低位「バイナリコラボレーション」を再利用しながら上位「バイナリコラボレーション」を指定する
6. 「バイナリコラボレーション」を再利用しながら「複数当事者コラボレーション」を指定する

本節では道案内のため、モデルを 1 から説明するが、仕様スキーマは多分に、既存の低位内容を可能な限り利用するトップダウン式モデリングのためのものである。

上述された構築方法では、非常に複雑な複数当事者コラボレーションの仕様がサポートされる。しかし ebXML 準拠のビジネスプロセス仕様は、1 つのバイナリコラボレーションが 1 つの取引トランザクションを参照しているというように単純な場合がある。この場合、仕様には数値 1 以上のみが使用されるため、『ebXML Business Process Specification』には、高いレベルのバイナリコラボレーション、複数当事者コラボレーション、および振り付け表示は必要ないことになる。

6.4.1 取引トランザクションとその取引文書フローを指定する

取引トランザクションを図6で示す。

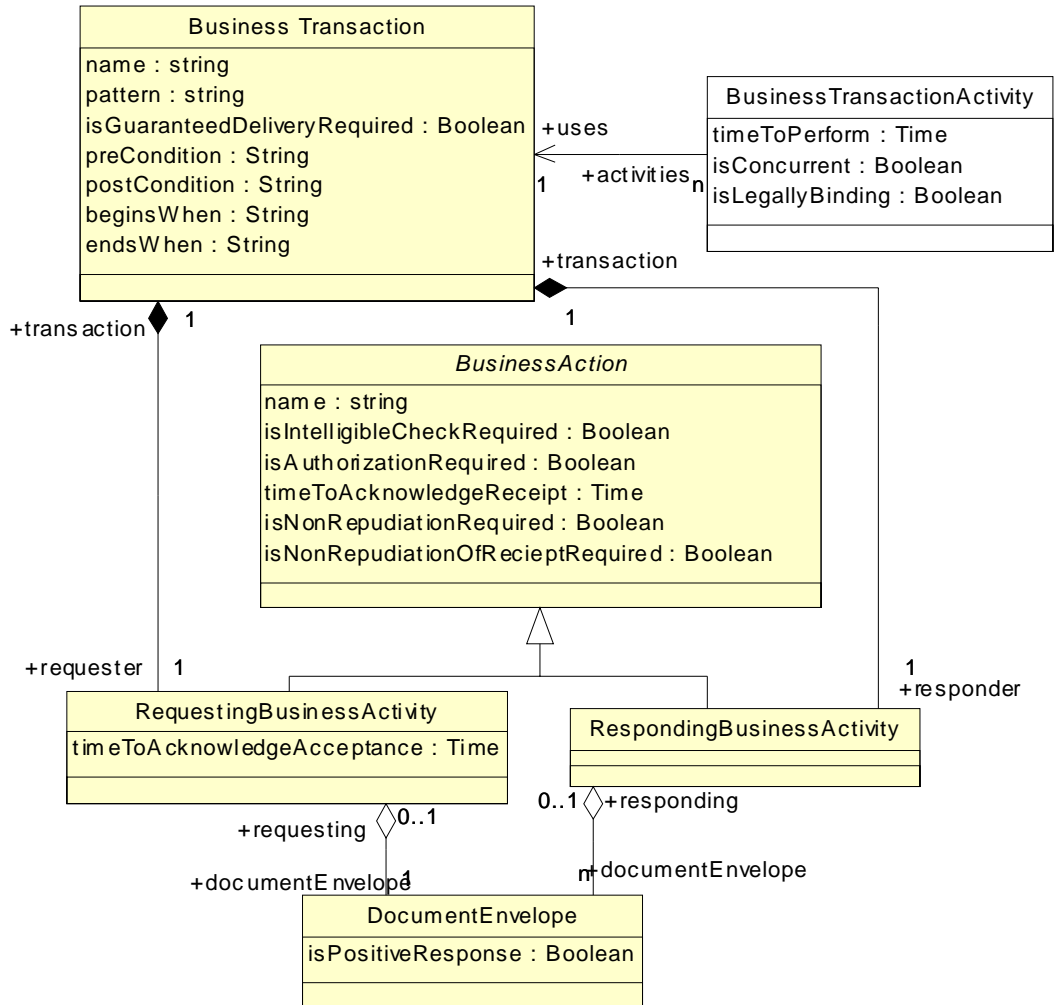


図6：取引トランザクションのUML意味情報

6.4.1.1 取引トランザクションの主要意味情報

取引トランザクションは、2者の取引当事者間のトランザクション取決めにおけるタスクの原子的単位である。

取引トランザクションは、「要求される業務アクティビティ」と「応答業務アクティビティ」、ならびに前記2アクティビティ間における1つまたは複数の「文書フロー」からなる。1つの取引トランザクションは、

さらに1つ以上のビジネスシグナルによってサポートされている場合がある。この場合のビジネスシグナルは、トランザクションの確認および関連事項の意味とその使用を統制する。

「要求される業務アクティビティ」を実行する要求役割と、「応答業務アクティビティ」を実行する応答役割は暗黙的に存在する。これらの役割は、「バイナリコラボレーション」の中の「取引トランザクションアクティビティ」の中でトランザクションが行われるときに明確になる。

「要求文書フロー」は常に存在する。

「応答文書フロー」が要求されるか否かは、取引トランザクション定義に含まれる。通常なら契約や合意を成立させるため、ある種の取引トランザクションにはこの種の要求と応答が必要となる。それ以外の取引トランザクションは通知に近いものであり、「要求文書フロー」しかない。

抽象スーパークラスにあたる「業務アクティビティ」は、「要求される業務アクティビティ」と「応答業務アクティビティ」の双方に共通の属性を保持する。

6.4.1.2 サンプル構文

これは、文書フローが1つしかない簡単な通知トランザクションである。

```
<BusinessTransaction name="Notify of advancesshipment">
  <RequestingBusinessActivity name="">
    <DocumentFlow isSuccess="true"
      documentType name="ASN"/>
  </RequestingBusinessActivity>
  <RespondingBusinessActivity name="">
  </RespondingBusinessActivity>
</BusinessTransaction>
```

「文書フロー」は、「文書フロー」を確認する1つまたは複数のビジネスシグナルを伴うことがある。確認シグナルは明示的にモデル化されないが、シグナルが要求されるか否かはトランザクションのパラメータで指定する。

取引トランザクションの中の「文書フロー」とビジネスシグナルは図7に示す通りである。

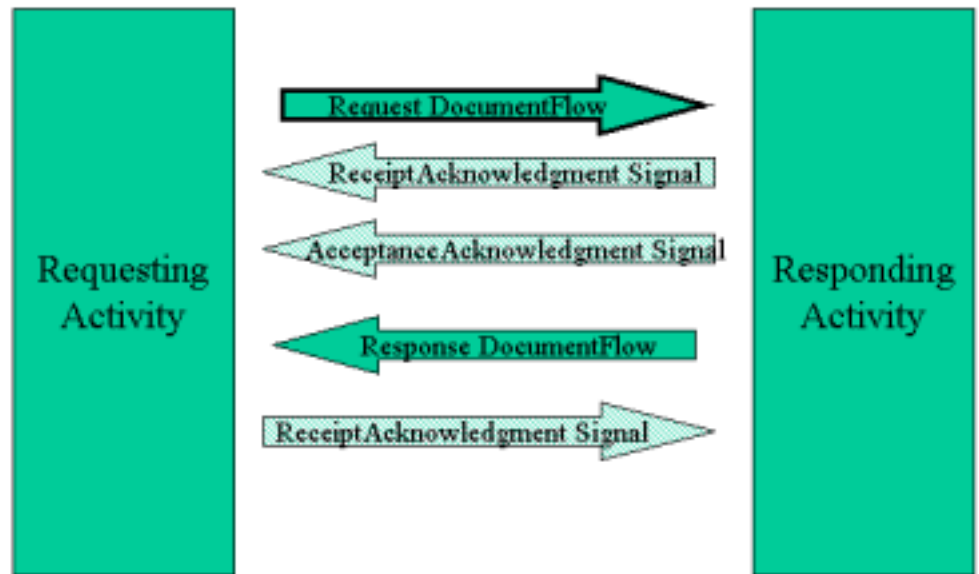


図7: 文書フローとシグナルとその順序

確認シグナル（「ビジネスシグナル」）は、取引トランザクションの現状を「合図する」アプリケーションレベルの文書である。

receiptAcknowledgement シグナルや acceptanceAcknowledgement シグナルを送信するか否かは、取引トランザクションの相互作用パターンを選ぶ際に選ぶ。ビジネスシグナルは固有のビジネス目的を持ち、低位のプロトコルと伝送シグナルから分離している。

受領確認ビジネスシグナルを使用すると、メッセージが正しく受信されたことが通知される。*isIntelligibleCheckRequired* プロパティを用いて、読み取りが可能であるメッセージの受領確認が実行されることに関し、取引当事者らの合意を取り付けることができる。「読み取り可能である」というのは、構造およびスキーマの妥当性が確認済みのメッセージを意味する。メッセージが確実に受領されたか、そして（メッセージ受領確認後に）そのメッセージが読み取り可能であるかどうかの確認が実行されるのは、取引規則が適用される前、あるいはメッセージの取引文書または文書エンベロープ内の表現やガード表示の評価が行われる前である。

受付確認ビジネスシグナルを使用すると、メッセージが受領され、ビジネスプロセスとして処理されるよう受付が行われたことが通知される。

この通知がされるのは、このメッセージの取引文書および文書エンベロープの取引規則の妥当性が確認されたことが前提となる。

要求された(timeToAcknowledgeReceipt あるいは timeToAcknowledgeAcceptance においてタイムアウト値を指定することによって要求された)どちらのシグナルも送信されていない場合、トランザクションは null として無効になり、関連づけられた timeToPerform を満たす取引文書の受領に依存したであろう ' 成功した ' 終了状態が実現できない。

6.4.1.3 サンプル構文

これは、2つの「文書フロー」と3つのビジネスシグナルを伴うやや複雑なトランザクションである。

この要求は受領確認と受付確認の両方を要求し、応答は受領確認のみを要求する。P2D は ISO 8601 に準拠した W3C スキーマ構文であり、期間=2日を意味する。P3D は期間=3日を意味し、P5D は期間=5日を意味する。これらの期間はどれも、要求を最初に送信するときから測る。

```
<BusinessTransaction name="Create Order">
  <RequestingBusinessActivity name=""
    isNonRepudiationRequired="true"
    timeToAcknowledgeReceipt="P2D"
    timeToAcknowledgeAcceptance="P3D">
    <DocumentFlow isSuccess="true"
      documentType="Purchase Order"/>
  </RequestingBusinessActivity>
  <RespondingBusinessActivity name=""
    isNonRepudiationRequired="true"
    timeToAcknowledgeReceipt="P5D">
    <DocumentFlow isSuccess="true"
      documentType="PO Acknowledgement"/>
  </DocumentFlow>
</RespondingBusinessActivity>
</BusinessTransaction>
```

6.4.1.4 文書フローを指定する

要求文書

「要求文書フロー」と「応答文書フロー」の中には、取引トランザクションに関係のある「取引文書」がある。そのモデルを図8に示す。取引文書には様々な構造が見られるが、ビジネスシグナルにはいつも同じ構造が使用される。この構造は、『ebXML ビジネスプロセス仕様スキーマ』内で1回のみ定義されている。

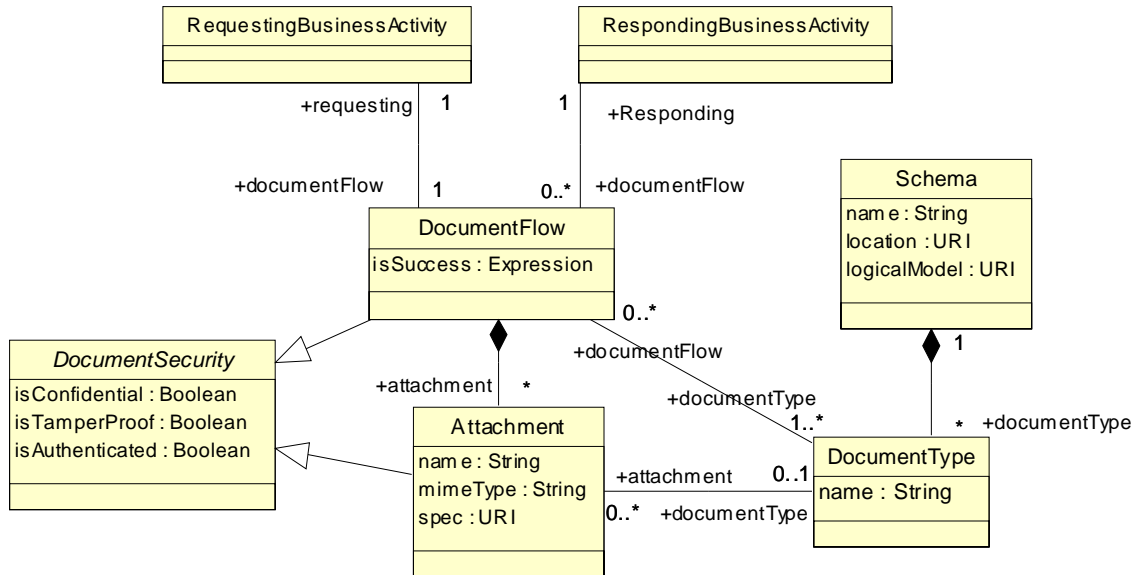


図 8 : 文書フローの UML 意味情報

文書フローは、直接モデル化されることはなく、1つの役割から送信され別の役割に受領される文書エンベロープとして間接的にモデル化される。文書エンベロープに、1つの「要求される業務アクティビティ」と1つの「応答業務アクティビティ」にいつも付随させることにより、フローのモデル化が実現される。

文書エンベロープには名前がつけられる。1つの「要求アクティビティ」ごとに、1つの名前を持つ文書エンベロープが常に存在する。名前が付けられた文書フローエンベロープは、1つの「応答アクティビティ」に対して全く存在しない場合もあり、1つ、あるいは相互排他的に多数存在する場合もある。たとえば、買付注文トランザクションの「応答文書エンベロープ」なら、PurchaseOrderAcceptance、PurchaseOrderDenial、PartialPurchaseOrderAcceptance などの名前が考えられる。ただし、買付注文トランザクションを実際に行うときは、定義された応答のうち1つだけが送信される。

「文書エンベロープ」は、アクティビティ間の文書の流れを表わす。各「文書フロー」は、ただ1つの「取引文書」を運ぶ。

「文書エンベロープ」はオプションとして、一次「取引文書」に関係のある1つまたは複数の添付文書を伴うことがある。基本的にはこの文書とその添付文書によって、ebXML メッセージ取扱サービスメッセージ構造の搬送内容で1つのトランザクションが成立する。

6.4.1.5 サンプル構文

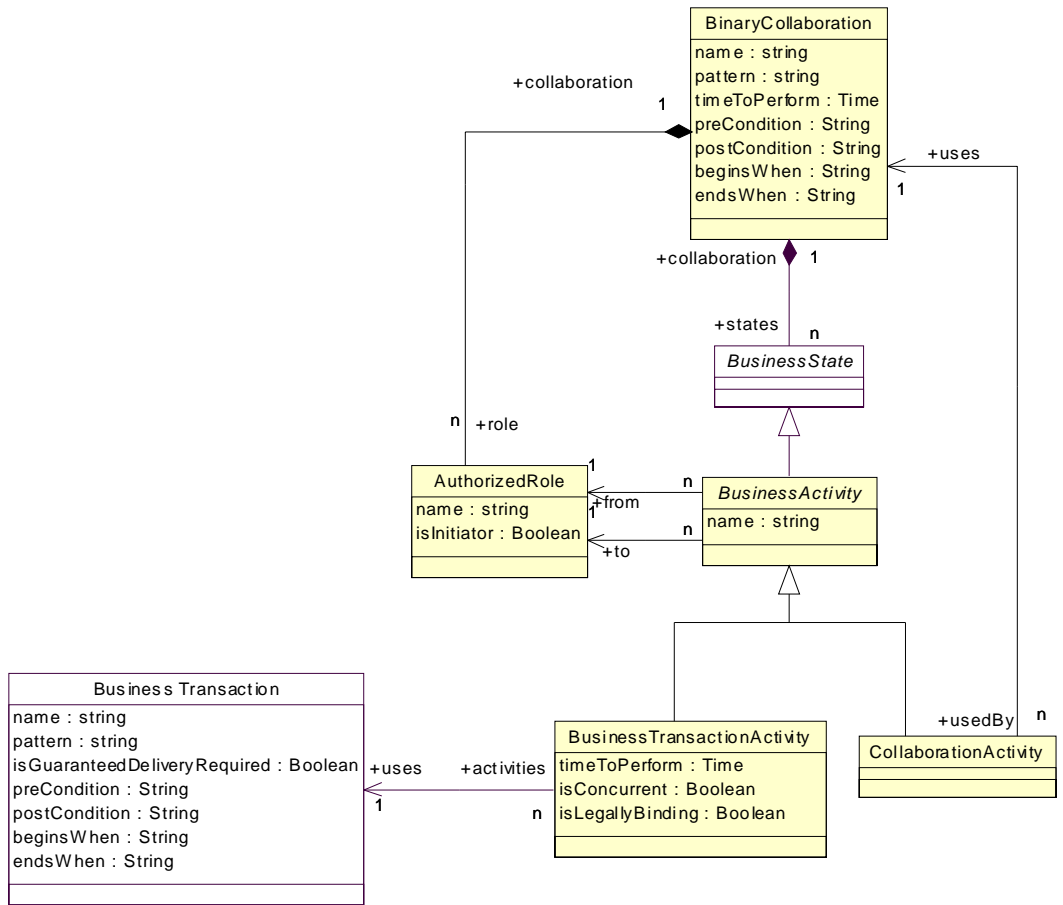
この例は、1つの要求と、成功および失敗の2応答を伴う取引トランザクションを示している。要求には添付文書がある。すべての文書は、スキーマ名で完全修飾されている。

```
<Schema name="ebXML1.0" location="someplace"
      logicalModel="someplaceAlso">
  <DocumentType name=" Purchase Order "/>
  <DocumentType name=" PO Acknowledgement "/>
  <DocumentType name=" PO Rejection "/>
  <DocumentType name="Delivery Instructions"/>
</Schema>

<BusinessTransaction name="Create Order">
  <RequestingBusinessActivity name=""
    <DocumentFlow isSuccess="true"
      documentType="ebXML1.0/PO Acknowledgement">
      <Attachment
        name="DeliveryNotes"
        mimeType="XML"
        documentType=
          "ebXML1.0/Delivery Instructions"
        specification=""
        isConfidential="true"
        isTamperProof="true"
        isAuthenticated="true">
      </Attachment>
    </DocumentFlow>
  </RequestingBusinessActivity>
  <RespondingBusinessActivity name=""
    <DocumentFlow isSuccess="true"
      documentType="ebXML1.0/PO Acknowledgement"/>
    </DocumentFlow>
    <DocumentFlow isSuccess="false"
      documentType=" ebXML1.0/PO Rejection"/>
    </DocumentFlow>
  </RespondingBusinessActivity>
</BusinessTransaction>
```

6.4.2 バイナリコラボレーションを指定する

バイナリコラボレーションを図9で示す。



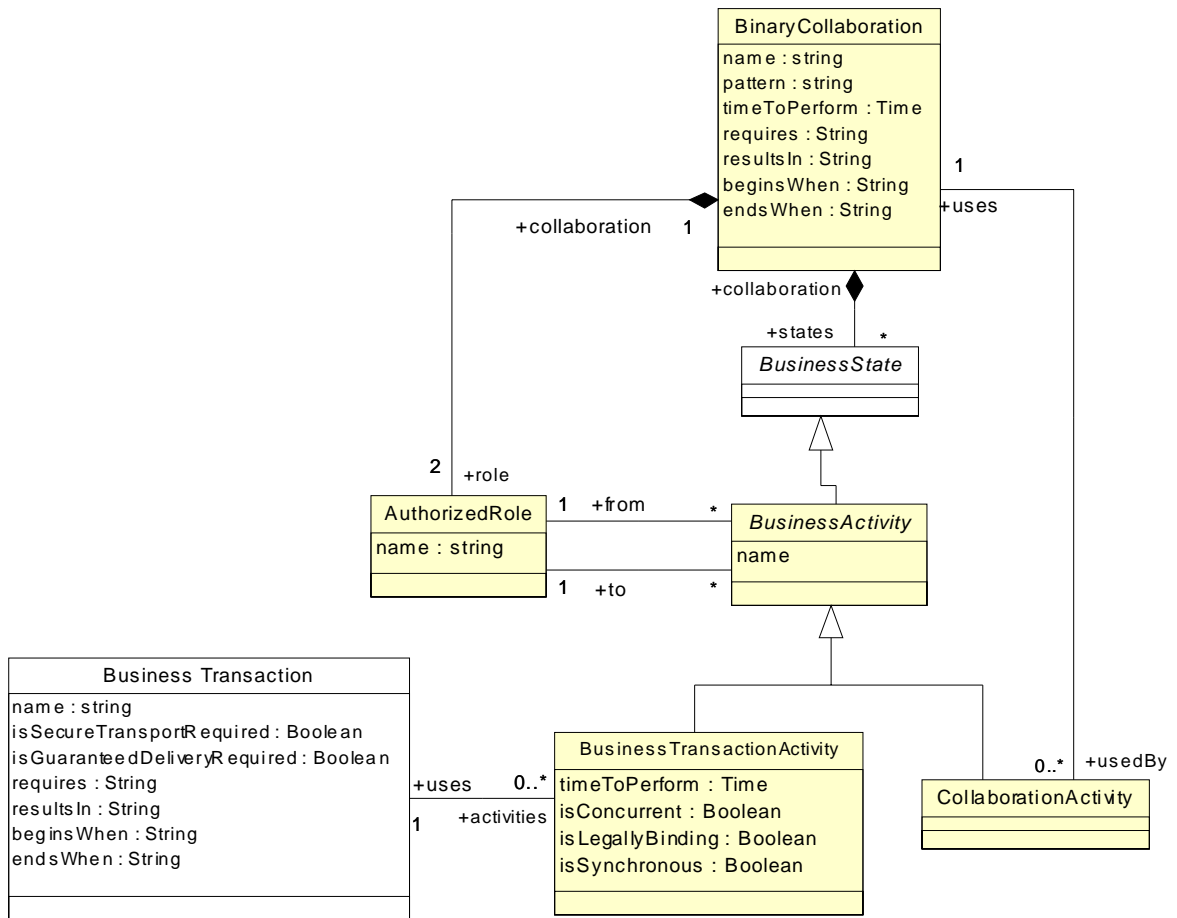


図9：バイナリコラボレーションのUML意味情報

6.4.2.1 バイナリコラボレーションの主要意味情報

「バイナリコラボレーション」は常に2つの役割の間で行われる。2つの役割は、コラボレーションへの関与を許可された作用主（アクター）だから、「認定役割」と呼ばれる。

「バイナリコラボレーション」は1つまたは複数の「業務アクティビティ」からなる。「業務アクティビティ」は常に、「バイナリコラボレーション」の2「認定役割」の間で行われる。各アクティビティにつき、2役割のうち的一方には InitiatingRole (from) が割り当てられ、他方には RespondingRole (to) が割り当てられる。

「業務アクティビティ」は「取引トランザクションアクティビティ」か「コラボレーションアクティビティ」のいずれかになる。

「取引トランザクションアクティビティ」は取引トランザクションを履行することである。取引トランザクションは再利用可能である。同じ取引トランザクションが別々の「バイナリコラボレーション」の中で複数の「取引トランザクションアクティビティ」を通じて履行されることがあり、同じ「バイナリコラボレーション」の中で複数の「取引トランザクションアクティビティ」を通じて履行されることさえある。

「コラボレーションアクティビティ」は、1つの「バイナリコラボレーション」を別の「バイナリコラボレーション」の中で履行することである。「コラボレーションアクティビティ」に関して、「バイナリコラボレーション」は再利用可能である。同じ「バイナリコラボレーション」が別々の「バイナリコラボレーション」の中で複数の「コラボレーションアクティビティ」を通じて履行されることがあり、同じ「バイナリコラボレーション」の中で複数の「コラボレーションアクティビティ」を通じて履行されることさえある。

ある1つの「バイナリコラボレーション」の中で「バイナリコラボレーション」を実行するときには、2つのレベルで役割間に暗黙的な関係が存在する。「バイナリコラボレーション」Xが「コラボレーションアクティビティ」Qを通じて「バイナリコラボレーション」Yを実行すると仮定する。「コラボレーションアクティビティ」Qでは、「顧客」を開始者とし、「業者」を応答者とする。「バイナリコラボレーション」Xには、「買手」と「売手」という「認定」役割と「取引トランザクションアクティビティ」があり、その「取引トランザクションアクティビティ」の中では「買手」は開始者にあたり、「売手」は応答者にあたる。「顧客」と「買手」はいずれも、履行する「バイナリコラボレーション」と履行される「バイナリコラボレーション」のアクティビティの中で開始者にあたるので、役割「顧客」および「買手」の間には役割関係が成立する。

取引トランザクションは本質的に原子的であるため、「取引トランザクションアクティビティ」を通じて1つの取引トランザクションを履行することも本質的に原子的である。所望の意味情報が原子的でなければ、そのタスクを複数のトランザクションにわたって分割する必要がある。たとえば、1つの要求について数個の部分受付を表現することが望ましいなら、その要求をバイナリコラボレーション内の1トランザクションとして表現する必要があり、部分受付は個別のトランザクションとして表現する必要がある。

「CPA/CPP仕様」によると、ビジネスを実行するにあたって当事者は「コラボレーションプロトコル合意書」(CPA)について合意する必要がある。CPAは特定の「バイナリコラボレーション」に連合する。したがって、2者間で行われる取引トランザクションはどれも、「バイナリコラボレーション」の中に含まれる「取引トランザクションアクティビティ」を通じて参照されなければならない。

6.4.2.2 サンプル構文

これは、先程定義した取引トランザクションの1つを使用する簡単な「バイナリコラボレーション」である。

```
<BinaryCollaboration name="Firm Order"
timeToPerform="P2D">
  <Documentation>
    timeToPerform =
      Period: 2 days from start of transaction
  </Documentation>
  <AuthorizedRole name="buyer"/>
  <AuthorizedRole name="seller"/>
  <BusinessTransactionActivity name="Create Order"
    businessTransaction="Create Order"
    fromAuthorizedRole="buyer"
    toAuthorizedRole="seller"/>
</BinaryCollaboration>
```

今度はやや複雑な「バイナリコラボレーション」であり、先程の「バイナリコラボレーション」と同じ取引トランザクションを再利用し、別の取引トランザクションを加えている。

```
<BinaryCollaboration name="Product Fulfillment"
timeToPerform="P5D">
  <Documentation>
    timeToPerform =
      Period: 5 days from start of transaction
  </Documentation>
  <AuthorizedRole name="buyer"/>
  <AuthorizedRole name="seller"/>
  <BusinessTransactionActivity name="Create Order"
    businessTransaction="Create Order"
    fromAuthorizedRole="buyer"
    toAuthorizedRole="seller"
    isLegallyBinding="true" />
  <BusinessTransactionActivity
    name="Notify shipment"
    businessTransaction="Notify of advance
    shipment"
    fromAuthorizedRole="buyer"
    toAuthorizedRole="seller"/>
</BinaryCollaboration>
```

6.4.3 複数当事者コラボレーションを指定する 複数当事者コラボレーションを図 10 で示す。

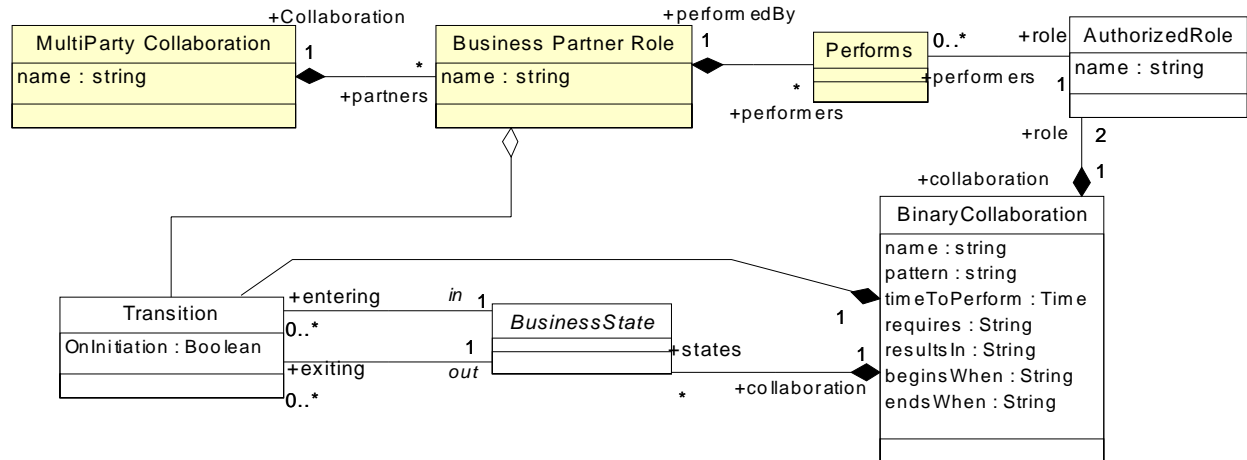


図 10：複数当事者コラボレーションの UML 図

6.4.3.1 複数当事者コラボレーションの主要意味情報

「複数当事者コラボレーション」は「バイナリコラボレーション」の合成である。

「複数当事者コラボレーション」は、数々の「取引当事者役割」を伴う。各々の「取引当事者役割」は、1つのバイナリコラボレーションの中で1つの「認定役割」を実行し、場合によっては、数個のバイナリコラボレーションの中で1つの「認定役割」を実行する。これは Performs 要素を使って表現する。

「取引当事者役割」と「認定役割」との間の'Performs'リンクによって、「バイナリコラボレーション」は「複数当事者コラボレーション」に合成される。暗黙的には、「複数当事者コラボレーション」を構成するすべての「バイナリコラボレーション」の中で、「取引当事者役割」が「認定役割」を果たす。

トランザクション取引当事者からなる二項対の各々は、1つまたは複数の個別 CPA の対象になる。

以下に記すように、「複数当事者コラボレーション」の中では、「バイナリコラボレーション」の中で「取引トランザクションアクティビティ」間の遷移を振り付けする場合がある。

6.4.3.2 サンプル構文

これは、上で定義した「バイナリコラボレーション」を使用する簡単な「複数当事者コラボレーション」である。

```
<MultiPartyCollaboration name="DropShip">
```

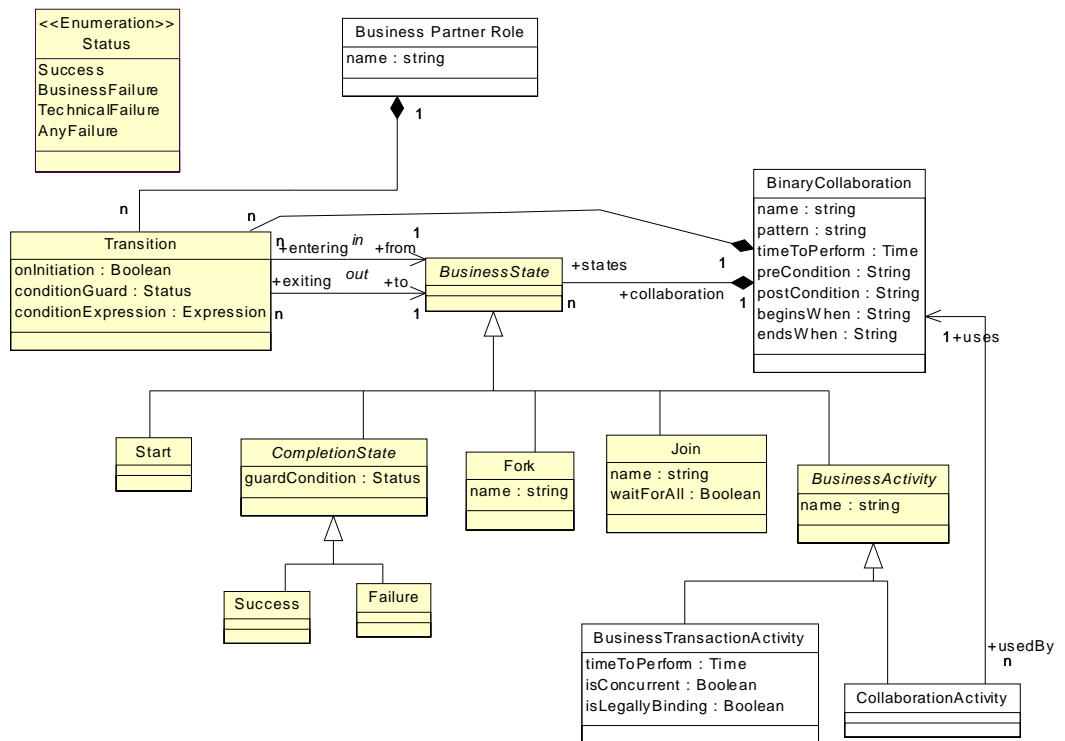
```

<BusinessPartnerRole name="Customer">
  <Performs
    binaryCollaboration="Firm Order"
    authorizedRole="buyer" />
</BusinessPartnerRole>
<BusinessPartnerRole name="Retailer">
  <Performs
    binaryCollaboration="Firm Order"
    authorizedRole="seller" />
  <Performs
    binaryCollaboration="Product Fulfillment"
    authorizedRole="buyer" />
</BusinessPartnerRole>
<BusinessPartnerRole name="DropShip Vendor">
  <Performs
    binaryCollaboration="Product Fulfillment"
    authorizedRole="seller" />
</BusinessPartnerRole>
</MultiPartyCollaboration>

```

6.4.4 振り付けを指定する

振り付けを図 11 で示す。



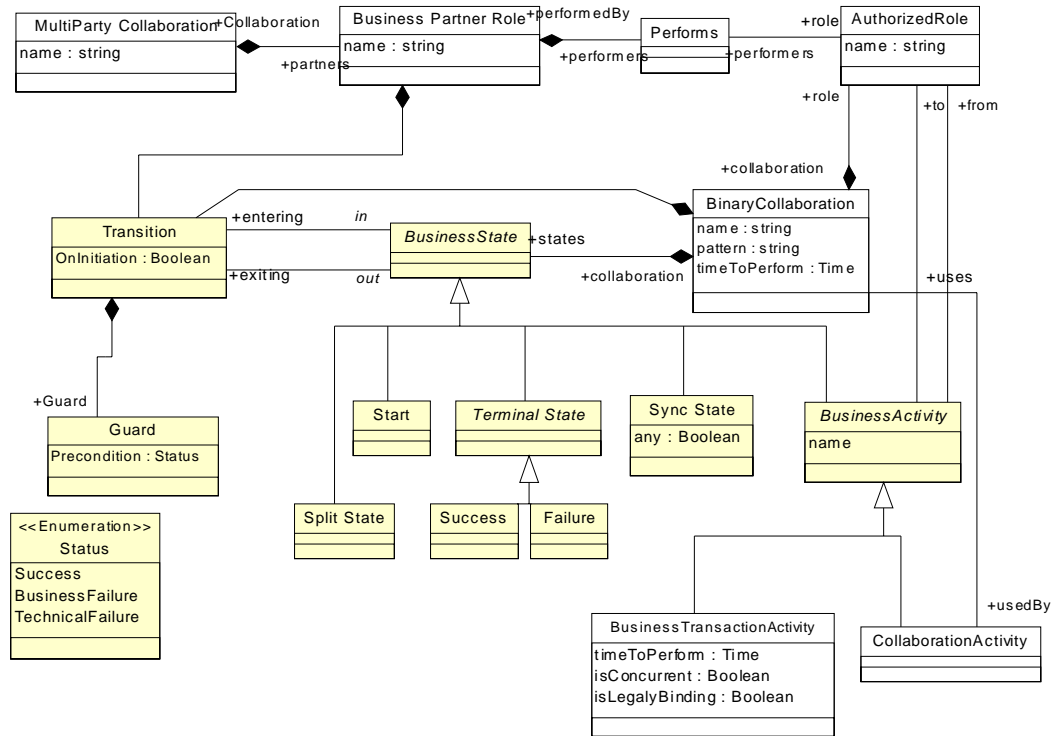


図 11：振り付けのUML図

6.4.4.1 Choreography 振り付けの主要意味情報

振り付けとは、「バイナリコラボレーション」の中で「業務アクティビティ」を整列・配列するものである。

振り付けは、「ビジネス状態」と、「ビジネス状態」間の遷移として指定される。

「業務アクティビティ」は抽象的な「ビジネス状態」である。そのサブタイプにあたる「取引トランザクションアクティビティ」と「コラボレーションアクティビティ」は具象的な「ビジネス状態」である。振り付けの目的は、1つの取引コラボレーションの中で「取引トランザクションアクティビティ」や「コラボレーションアクティビティ」を整理・配列したり、「複数当事者コラボレーション」内の複数の「バイナリコラボレーション」にまたがって「取引トランザクションアクティビティ」や「コラボレーションアクティビティ」を整理・配列することである。

「ビジネス状態」の振り付けを促進する補助的な「ビジネス状態」がいくつか存在する。「開始」状態、「終了」状態（これは「成功」型か「失敗」型に帰結する）、「分割」状態、「同期化」状態である。これらはどれも、UML アクティビティ図上で生成物を図式化することに等しい。

遷移は「ビジネス状態」間に起こる。遷移の進行は「ガード」を使って封鎖できる。「ガード」は、遷移を引き起こした「文書フロー」のステータス、送信された「文書」のタイプ、文書の内容、または先行状態に関する事後条件を照会できる。

「遷移」を利用すれば、入れ子の BusinessTransactionActivity を作ることもできる。入れ子の BusinessTransactionActivity では、最初のトランザクションで要求が受信された後に最初の遷移が生じ、それから2番目のトランザクションを全部実行し、その後に最初のトランザクションに戻って当初の要求者に応答を返信する。それには、BusinessTransactionActivity で 'onInitiation' フラグを使用する。入れ子の BusinessTransactionActivity は普通、複数当事者コラボレーションの中に存在する。基本的には、ある1つの「バイナリコラボレーション」で「認定役割」が要求を受け取り、次に方向転換して別の「バイナリコラボレーション」内の要求者に転じ、それからもとに戻って最初の「バイナリコラボレーション」の中で応答を送信する。

isConcurrent は、トランザクションの流れを取り仕切るパラメータである。セキュリティやタイミングのパラメータと違って、併発性は内的なトランザクションの流れを取り仕切るのではなく、むしろ当該トランザクション型の複数のインスタンスを、同一取引トランザクションアクティビティの一部として同時に '開放' できるかどうかを判断する。

IsConcurrent は、このような機能を統制するパラメータであり、取引トランザクションアクティビティのレベルに位置する。

6.4.4.2 サンプル構文

これは、先程と同じ「バイナリコラボレーション」の末尾に振り付けを加えたものである。2つの間の遷移、開始、および2つのコラボレーションの結果、すなわち成功と失敗がある。

```
<BinaryCollaboration name="Product Fulfillment"
timeToPerform="P5D">
  <Documentation>
    timeToPerform =
      Period: 5 days from start of transaction
  </Documentation>
  <AuthorizedRole name="buyer"/>
  <AuthorizedRole name="seller"/>
  <BusinessTransactionActivity name="Create Order"
    businessTransaction="Create Order"
    fromAuthorizedRole="buyer"
    toAuthorizedRole="seller"/>
  <BusinessTransactionActivity
    name="Notify shipment"
    businessTransaction="Notify of advance
shipment"
    fromAuthorizedRole="buyer"
```

```

        toAuthorizedRole="seller"/>
<Start toBusinessState="Create Order"/>
<Transition
    fromBusinessState="Create Order"
    toBusinessState="Notify shipment"/>
<Success fromBusinessState="Notify shipment"
    guard="Success"/>
<Failure fromBusinessState="Notify shipment"
    guard="BusinessFailure"/>
</BinaryCollaboration>

```

これは以前に定義したのと同じ「複数当事者コラボレーション」だが、2つの「バイナリコラボレーション」にまたがって簡単な振り付け（遷移）がある。

```

<MultiPartyCollaboration name="DropShip">
  <BusinessPartnerRole name="Customer">
    <Performs
      binaryCollaboration="Firm Order"
      authorizedRole="buyer"/>
    </BusinessPartnerRole>
  <BusinessPartnerRole name="Retailer">
    <Performs
      binaryCollaboration="Firm Order"
      authorizedRole="seller"/>
    <Performs
      binaryCollaboration="Product Fulfillment"
      authorizedRole="buyer"/>
    <Transition
      fromBinaryCollaboration"Firm Order"
      fromBusinessState="Create Order"
      topBinaryCollaboration"Product Fulfillment"
      toBusinessState="Create Order"/>
    </BusinessPartnerRole>
  <BusinessPartnerRole name="DropShip Vendor">
    <Performs
      binaryCollaboration="Product Fulfillment"
      authorizedRole="seller"/>
    </BusinessPartnerRole>
</MultiPartyCollaboration>

```

6.4.5 全体モデル

図 12 は、以上の意味情報を 1つの UML クラス図にまとめたものである。完全な状態の UML 版『ebXML ビジネスプロセス仕様スキーマ』がここにある。

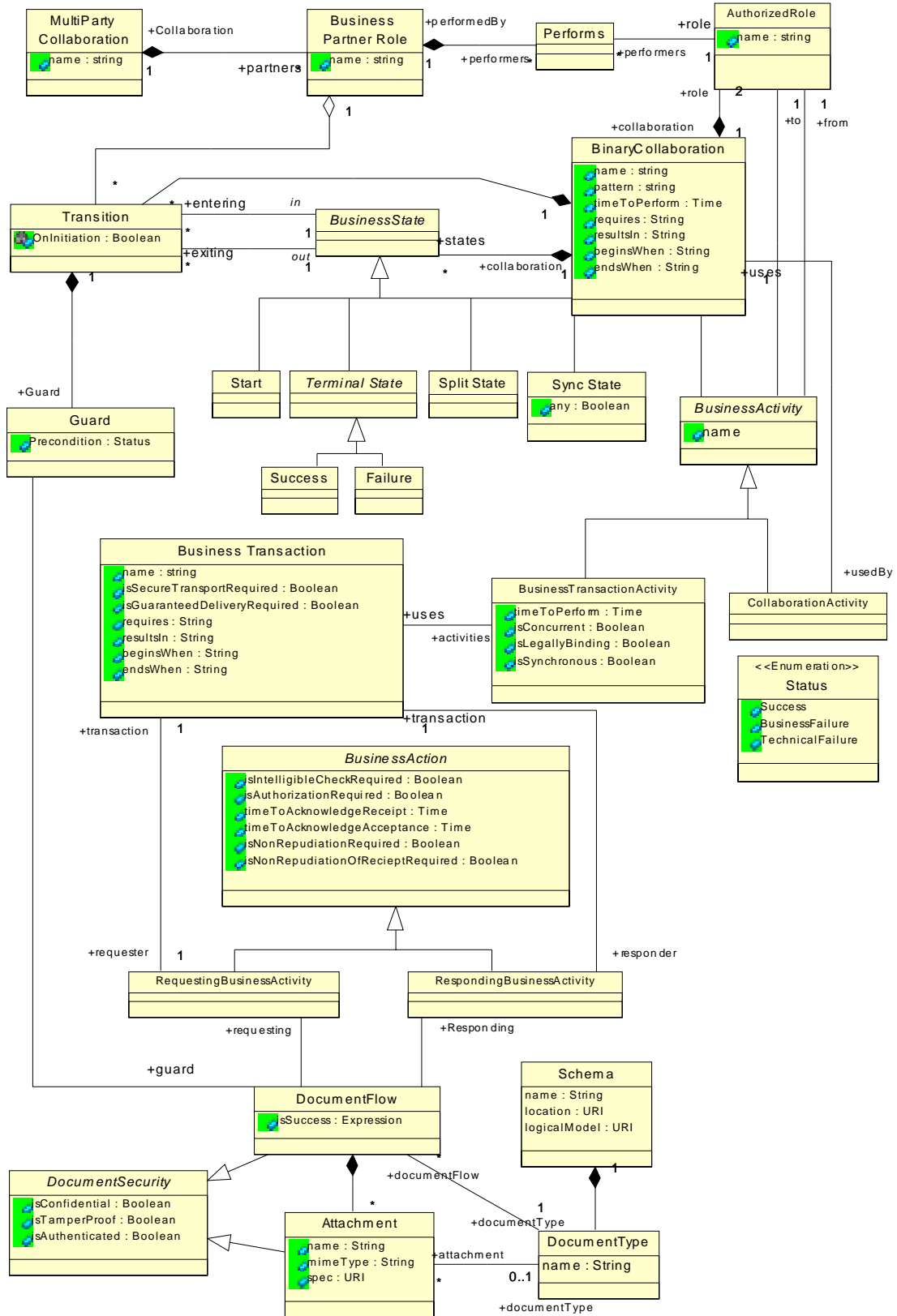


図 12 : UML クラス図の形をとる全体仕様スキーマ

6.5 コア取引トランザクション意味情報

ebXML の取引トランザクション概念とその背後にある意味情報は、予測可能で実行可能の商業の中枢をなすものである。どの「取引サービスインタフェース」(BSI)も、これらの意味情報に従ってトランザクションを処理できることが見込まれる。

ebXML 取引トランザクション意味情報により、以下の事項を提供する電子商取引トランザクションを指定することができる。

- 相互作用の予測の可能性。これには、それぞれ明確な役割、トランザクション範囲、時間範囲、ビジネス情報意味情報、成功または失敗の判断などが含まれる。
- 「法的拘束力を持った契約」を作成する機能。例えば、関係当事者に対して拘束力を持つ取引トランザクションを指定する機能。
- 否認不可。法的強制力を強化するために、生成物の保持を指定することができる。
- 承認セキュリティ。役割を持つ関係当事者の承認を要求するよう指定することができる。
- 文書セキュリティ。承認や認証の実行、機密事項としての扱い、不正操作を不可能にする、などの指定をすることができる。
- 信頼性。取引文書およびビジネスシグナルの信頼できる配信を指定することができる。
- 実行時取引トランザクション意味情報。取引サービスインタフェース用ソフトウェアが、ebXML 取引トランザクションを予想可能で実施可能にするために必要とする規則と設定パラメータなどが含まれる。

ebXML 取引トランザクション意味情報の、上述した各特長を以下で詳細に論じる。

6.5.1 相互作用の予測の可能性

すべての取引トランザクションは、正確に定められたフロー、つまり詳細に定義されたサブセットに完全に従う。このフローの概要を以下の図で示す。ここでは、2つの取引当事者をまたがる状態マシンとした例を挙げる。UMM メタモデルの N090R9.1 の章において、各取引当事者の状態に関する詳細の図表が参照できる。

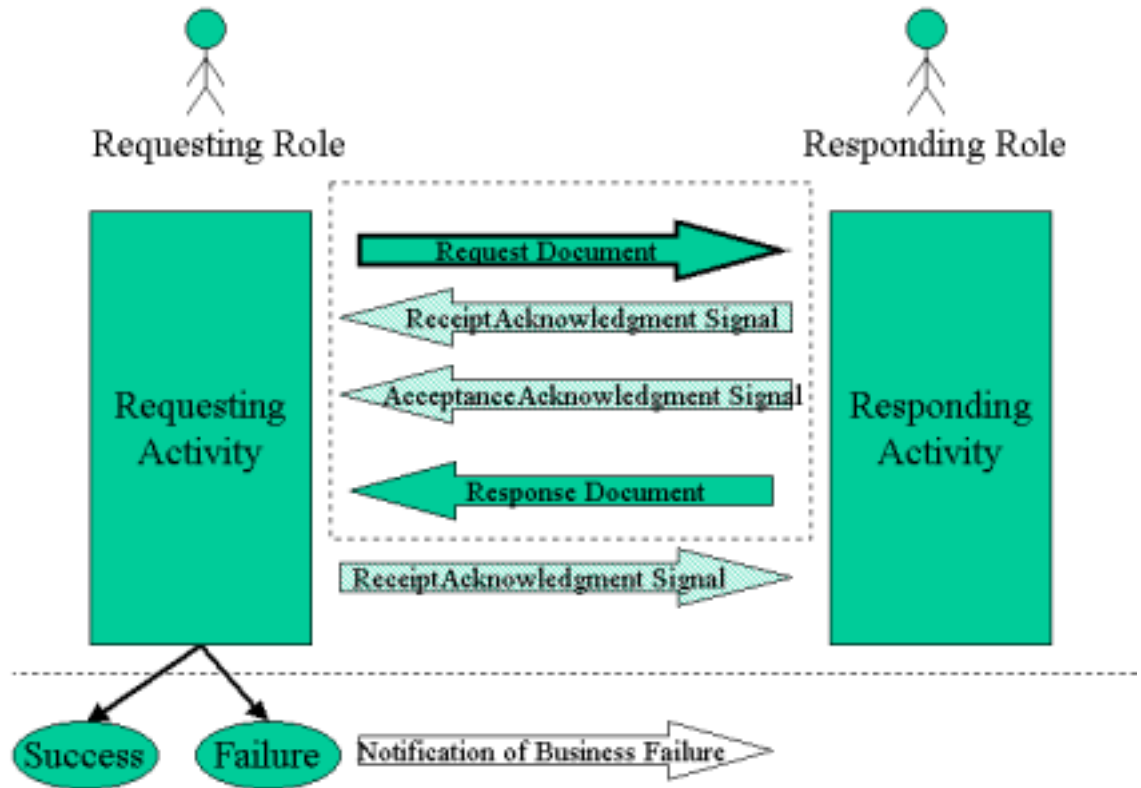


図 13：コア取引トランザクション意味情報の構造図

ebXML モデルの中では、取引トランザクションは常に以下の意味情報を持つ。

1. 取引トランザクションは仕事の単位である。取引トランザクションにおける相互作用はどれも成功しなければならないし、さもなくばトランザクションが始まる前の所定の状態までトランザクションを復帰（ロールバック）させなければならない。
2. 取引トランザクションは、トランザクションの中で対極の役割を果たす 2 取引当事者の間で行われる。それらの役割は常に「要求役割」と「応答役割」である。
3. 取引トランザクションの定義では、「要求アクティビティ」が制御される状況と、「応答アクティビティ」が制御される状況と、ある状態から別の状態にかけて制御が遷移する状況を明確に指定する。あらゆる取引トランザクションにおいて、制御は「要求アクティビティ」からスタートし、それから「応答アクティビティ」に移り、それから「要求アクティビティ」に戻る。
4. 取引トランザクションは常に、要求アクティビティによって送出される要求から始まる。
5. その要求によって制御が応答役割に移る。

6. 「要求文書フロー」の受信後、応答アクティビティは要求役割に向けて receiptAcknowledgement シグナルか acceptanceAcknowledgement シグナルを送信する場合がある。
7. 次に応答役割は応答アクティビティにとりかかる。応答アクティビティの最中、または応答アクティビティの完了時には、0個または1個の応答が送信される。
8. receiptAcknowledgement、acceptanceAcknowledgement、または応答が要求どおりに指定されている場合、あるいはこれらすべてが指定されている場合、制御は要求アクティビティに返される。
receiptAcknowledgement(要求された場合)はいつも acceptanceAcknowledgement(要求された場合)の前に使用されなければならない。acceptanceAcknowledgement(要求された場合)は応答の前に使用されなければならない。制御は、これら3つの中(要求されていれば)で最後に要求されたものに基づいて、要求アクティビティに返される。どれも要求されていない場合、制御は応答アクティビティを維持する。
9. あらゆる取引トランザクションは成功するか失敗する。成功と失敗は以下の要因に左右される：
 - a. 要求、応答、ビジネスシグナルのうちどれか、またはこれらすべての受領、あるいは未受領
 - b. 時間切れ(タイムアウト)の発生
 - c. ビジネス例外の発生
 - d. 制御例外の発生
 - e. 受領した応答および遷移するガード表示を、成功あるいは失敗のどちらかで解釈する作業
10. 取引トランザクションの成功または失敗の決定は、上記の成功または失敗の要因に基づいて要求当事者が行う。成功か失敗の決定がされれば、この取引トランザクションは、両当事者に関して終了したものと認識される。
11. 応答の受領後、要求アクティビティは receiptAcknowledgement シグナルを応答役割に送り返す場合がある。このシグナルは単に通知を目的としたもので、制御を応答アクティビティに返すことや、応答の受領に基づいた取引トランザクションの成功または失敗の結果を変更するものではない。
12. 取引トランザクションの処理中、およびその後のトランザクションの終了時にタイムアウトまたは例外が識別されると、要求当事者は応答当事者に対して失敗の通知を送信する場合がある。これは新しい取引トランザクションとして認識され、すでに確立された取引トランザクションの終了を変更するものではない。

6.5.1.1 トランザクション相互作用のパターン

取引トランザクション仕様を用いて、「成功」の最終状態を実現するために、要求文書に応答用の実質的文書が必要であるかどうかを指定することができる。さらに、トランザクションで timeToPerform に 0 以外の適切な時間を指定することで、実質的応答の制限時間を設定することができる。

また、取引トランザクションの仕様により、要求に関して receiptAcknowledgement または acceptanceAcknowledgement のどちらか、あるいはこれら両方が必要とされているか、そして応答に関して receiptAcknowledgement が必要とされているかを示すことができる。

receiptAcknowledgement が要求されるよう指定するには、timeToAcknowledgeReceipt パラメータを 0 以外の適切な時間に設定する。このパラメータがすでに適切な 0 以外の時間に設定されている場合、オプションで isIntelligibleCheckRequired パラメータまたは isNonrepudiationOfReceiptRequired パラメータのどちらか一方を、あるいはこれら両方を 'Yes' に設定することもできる。

acceptanceAcknowledgement が要求されるよう指定するには、timeToAcknowledgeAcceptance パラメータを 0 以外の適切な時間に設定する。

これら 2 つの認証関連のパラメータは、シグナルがトランザクションの一部として要求されるべきかを指定する Boolean フラグとして、そしてシグナルを受信しなかった場合のトランザクションのタイムアウト値として倍に拡張される。UMM において、これらの組み合わせを表示するパターンが、再利用の目的で図示されている。

6.5.2 法的拘束力を持つ契約の作成

取引当事者は、ebXML に従って進められた取引トランザクションが法的拘束力を持つかどうかを表示されるよう希望している。法的拘束力を宣言することは、強制力を持つ署名入り物的書類と同等に法的な主要な要素である。当事者らが、このような法的拘束力を持たせる意図を明示的な文書として作成するには、(1) 『ebXML ビジネスプロセス仕様スキーマ』の標準規定に準拠すること、(2) 同標準規定においてこのような意図を示すよう指定されたパラメータ("isLegallyBinding")を操作すること、によって行われる。

初期の電子商取引用アプリケーションのいくつかでは、取引当事者らは電子署名(XML-DSIG 標準の使用などによる)の有無によってこのような意図が示されていた。しかし文書の意味情報内に、該当の契約が草案であることや、法的拘束力を持たないこと、あるいはその他の条件が含まれている場合、署名の有無だけで文書を正確に解釈することできない場合

もある。ebXML においては、XML-DSIG 署名は送信者の識別およびメッセージの整合性の確認などの他の目的で使用されるため、電子署名の有無だけでは法的拘束力があるかどうかの判断はできない。

isLegallyBinding は BusinessTransactionActivity レベルのパラメータであり、このパラメータを使用すると、バイナリコラボレーション内で実行される BusinessTransaction が法的拘束力を持つかどうかを指定することができる。

この標準に基づいて取引を行う場合、当事者は、条件に合意する旨(例：提供および認定など)を示した拘束力を持つメッセージを交換することで、拘束力のある契約を結ぶ。"isLegallyBinding"パラメータは Boolean で、デフォルト値は"true"である。この標準に基づいて、業務アクティビティが拘束力を持たないことを示すための排他的方法は、該当のトランザクションアクティビティの"isLegallyBinding"パラメータに"false"値を適用することである。EDI においてと同様 ebXML 標準では、特別の指定がされていない限り、取引トランザクションは拘束力を持つと取引当事者に認識される。

非規約的な問題として、当事者は、テストや、契約条件の合意まで行われる取引条件の提案と、それに対する提案のやりとりなどの様々な理由で、拘束力を持たないトランザクションを実行したい場合がある。署名された実際の書面が使用される場合は、当事者は手書きの署名を差し控えることや、"DRAFT(草案)"のスタンプを用いて、これを実現することができる。ebXML においては、取引当事者は"isLegallyBinding"パラメータを用いて同じ効果が得られる。「ebXML 電子商取引パターン」において示された単純交渉パターンの図解を参照のこと。

6.5.3 否認不可

トランザクション取引当事者は ebXML で、法律的に強制できる取引トランザクションの遂行を望む場合がある。相手方が後になって ebXML 取引文書とメッセージを否認しようとする場合に、裁判所において契約義務の主張に役立つ文書を生成するため、当事者は否認不可プロトコルの使用を選ぶ場合がある。

否認とは一般的に、トランザクションに関する不変の生成物を証拠に、取引当事者がトランザクションに合意していないことを後に主張することである。この主張の内容は、契約条件の作成時に、応答文書が送信されていないこと、適切な当事者によって送信されなかったこと、不正確に解釈された(適用可能な標準または取引当事者の取引規則に基づいて)などである。

本書のもとで、利用できる否認不可プロトコルは 2 種類ある。各プロトコルを用いて、後の否認紛争で役に立つ追加の生成物を作成および要求できるため、ユーザにとっての証拠確保が強化される。しかしどちらのプロトコルも、絶対的な証拠確保を保証する

ことはできない。書面文書を使用する場合、取引当事者はいつでも新たな主張を考案し、明白に強制可能な記述事項は無視される。これらのパラメータを用いて、このような行いを困難にすることができる。

一方の Protokol を使用すると、各当事者がすべての取引文書と文書エンベロープのコピーを保存するよう義務付けられる。これらの文書およびエンベロープは、当事者各自のトランザクションで構成され、例えば、要求者は自らが行った要求を、応答者は自らが行った応答を保存する。これを行うために、要求アクティビティおよび応答アクティビティにおいて

isNonRepudiationRequired パラメータが使用される。これは、相手である取引当事者が監査証跡を保持するよう要求するのと論理的には同じである。しかし、この要求に従わなかった事実は、実行時に操作によって検知されるとは限らず、また「成功」または「失敗」の最終状態の決定を上書きするとも限らない。

もう一つの Protokol は、署名された受取りのコピーを送信することを応答者に要求し、要求者は送信されたコピーを保存することである。これを行うには、要求される業務アクティビティにおいて isNonRepudiationOfReceiptRequired パラメータが使用される。

NonRepudiationOfReceipt は、ReceiptAcknowledgement と結び付けられており、ReceiptAcknowledgement にはデジタル署名が必要とされる。そのため ReceiptAcknowledgement が要求されなければ、NonRepudiationOfReceipt は意味を持たないことになる。NonRepudiationOfReceipt に従わない場合は、実行時に操作的に検知され、「失敗」の最終状態が上書きされる。

timeToAcknowledgeReceipt が要求メッセージに適用されおり、NonRepudiationOfReceipt が true である場合、デジタル署名されている受取りのみがタイムアウト時間制限に間に合うことになる。このように、timeToAcknowledgeReceipt の制限時間内に署名入りの受取りが送信されないと、トランザクションは null として無効になる。

パラメータ	BSI 要件
isNonRepudiationRequired	送信するメッセージの監査証跡を保存しなければならない
isNonRepudiationOfReceiptRequired	receiptAcknowledgements にはデジタル署名が必要

6.5.4 認定セキュリティ

要求や応答は、取引当事者に関係するさまざまな個人、代表、または自動システムによって送信される場合がある。トランザクション取引当事者が、種々の権限レベルに相当する複数の ebXML 対応取引サービスインタフェースを持つ場合がある。当事者はかかる場合に、企業の代表として確実に信頼できるインタフェース / オペレータがどれなのかについて規則を確立する場合がある。

そのような規則に頼るため、当事者は要求アクティビティや応答アクティビティでそれ相応に `IsAuthorizationRequired` を指定する場合があります、その場合の[アクティビティ]は、それを解釈する当事者が、アクティビティの[認定役割]の身元陳述を、その当事者によって事前に供給される許容値リストに照らして照合できる場合に限り、有効として処理される。

パラメータ	BSI 要件
<code>IsAuthorizationRequired</code>	認定発起者リストに照らして発起者の身元を確認しなければならない

`IsAuthorizationRequired` は、要求アクティビティおよび応答アクティビティに対して指定される。

6.5.5 文書セキュリティ

転送される各取引文書に使用される、以下に示したセキュリティの特性は、たとえ多くの文書が同じメッセージに集約されていたとしても、文書エンベロープ内において個別および集約的に指定することができる。

パラメータ	デリバリーチャネル要件
<code>isConfidential.</code>	情報エンティティを暗号化することによって、許可されない者が情報を閲覧できないようにする
<code>isTamperProof.</code>	情報エンティティに暗号化されたメッセージダイジェストを持たせることによって、メッセージが不正に変更されたかどうかをチェックできるようにする。それには、文書エンティティにデジタル署名（送信者のデジタル証明と暗号化されたメッセージダイジェスト）を付随させる必要がある。
<code>isAuthenticated.</code>	文書エンティティにデジタル証明が付随する。これは署名者の身元の証明になる。

文書エンベロープ内の *isConfidential*、*isTamperProof*、および *isAuthenticated* の値は、いつも主要な取引文書に適用できる。これらの値はまた、添付文書レベルで特に上書きされなければ、各添付文書に適用することもできる。

これらのパラメータが YES(あるいは TRUE)に設定されると、対応するセキュリティ特性が有効になり、持続性を発揮する。このセキュリティ特性に準拠すると、文書の指定された特性が、取引サービスインタフェースで受取り内容を保持し、この文書がアーカイブされるか転送されるまで持続されることになる。

6.5.6 信頼性

取引トランザクションレベルでのこのパラメータは、トランザクションの取引文書の保証された引渡しが必要であることを示す。

パラメータ	デリバリーチャンネル要件
IsGuaranteedDeliveryRequired	これは、転送「取引文書」の引渡しが保証されることを意味する。(デリバリーチャンネルまたは取引当事者以外の当事者による引渡し)。

これは、取引当事者が、関連するトランザクションにおける取引文書を送信するには、第3者による引渡しを保証するデリバリーチャンネルのみを使用しなければならないと宣言していることになる。

6.5.7 CPP/CPA の必須パラメータ

『ebXML ビジネスプロセス仕様スキーマ』が提供するパラメータを利用すれば、ある程度のセキュリティと信頼性を指定できる。仕様スキーマは、それらのパラメータを一般的なビジネス表現で提供する。

これらのパラメータはビジネスプロセスのための一般的要件であるが、ebXML 実装では特に、所定のサービス水準を達成するための能力を、CPP と CPA を通じて BSI やデリバリーチャンネルに要求するためにこれらのパラメータを使用する。

CPP と CPA はこれらを 2 種類のパラメータに翻訳する。

1 つのパラメータにより、デリバリーチャンネルに使用される転送方法および転送技術に適用可能な、特定のセキュリティおよび信頼性のパラメータが選択される。文書セキュリティ、および上述の信頼性が、デリバリーチャンネルを決定する。

もう 1 つのパラメータにより、特定のサービスレベルまたは BSI の機能が決定され、以下に示した実行時の取引トランザクション意味情報がサポートされるようになる。

6.6 実行時トランザクションの意味情報

ebXML の取引トランザクション概念とその背後にある意味情報は、予測可能で実施可能な商業の中枢をなすものである。どの「取引サービスインタフェース」(BSI)も、これらの意味情報に従ってトランザクションを処理できることが見込まれる。

そこで、「取引サービスインタフェース」(BSI)や、ebXML コラボレーションの中で1つの役割を実施するソフトウェアは、最低でも以下のトランザクション意味情報をサポートする必要がある：

1. トランザクションの開始を検知すること
2. 制御の転送を検知すること
3. トランザクションの成功を検知すること
4. トランザクションの失敗を検知すること
 - a. タイムアウトの検知
 - b. 例外の検知
5. 失敗を通知すること
6. 失敗通知を受け取ること
7. 失敗のときにロールバックすること(これは各役割の独立した責任であり、相互調整ロールバックであり、ebXML には2段階コミットがない)

これらのトランザクション意味情報をいかにして実装するかは ebXML で指定されていないが、「取引サービスインタフェース」(BSI)は実行時にこれらの初歩的トランザクション意味情報をサポートできるようになるだろう。いずれかの当事者が完全サポートを提供できない場合は、CPP/CPA において無効として要件が緩和される場合がある。

以下の節では、タイムアウトと例外という2つの失敗原因について論じる。いずれか一方が生じた場合に、必要とされるロールバックを実行してトランザクションを終了することは2役割の責任である。2つの役割の責任は若干異なり、以下の節で説明する。概して、応答役割側で失敗が発生する場合は、応答役割が応答役割に向けて例外シグナルを送り、両当事者が現行のトランザクションを終了する。要求役割側で失敗が発生すると、要求役割が現行のトランザクションを終了させ、別のトランザクションの中で応答役割に失敗を通知する。このように、トランザクションの中での制御の流れは常に明確で有限である。

6.6.1 タイムアウト

すべての取引トランザクションには明確な時間制限が必要であるため、応答に付随するタイムアウトパラメータ、およびそれぞれの

theacknowledgement シグナルが用意されている。必須の応答が到着する前にタイムアウトが生じると、そのトランザクションは無意味であり無効である。

3種の応答に関係のあるタイムアウトパラメータ：

必須の応答	パラメータ名	タイムアウトの意味
受領確認	timeToAcknowledgeReceipt	応答役割が取引文書の受領を確認するのにかかる時間
受付確認（形式的）	timeToAcknowledgeAcceptance	応答役割が取引文書の事務受付を形式的に確認するのにかかる時間
実質的応答	TimeToPerform	応答役割が取引文書の事務受付を実質的に確認するのにかかる時間

要求する側の取引当事者が取引文書要求に対する応答を期待する場合は必ず、タイムアウトパラメータを指定しなければならない。要求する側の取引当事者は、無限の待ち状態に留まってはならない。

タイムアウトパラメータのタイムアウト値は絶対値であり、相対値ではない。すべてのタイマーは、要求取引文書が送信されるときにスタートする。タイマーの値は、タイマー値に関するウェルフォームドルールに準拠しなければならない。

BSI は、上述のパラメータに準拠して正しくタイムアウトを検知する必要がある。取引トランザクションの原子的意味情報を維持するために、要求役割と応答役割はタイムアウトに基づいて異なった処理を行う。

タイムアウトが発せられると、応答する側の取引当事者は終結する。そうすることで、応答取引トランザクションが際限なく停滞する事態を防ぐことができる。

要求する側の取引当事者はタイムアウトが発せられると終結し、別のトランザクションの中で応答者に失敗通知を送る。

アクティビティを実行する時間が、受領確認時間か事務受付確認時間に等しい場合に、発起者がそもそもの取引文書提供を取りやめる理由を提示する際には、優先度が最も高いタイムアウト例外を使用しなければならない。例外を実行する時間は、受領確認時間と事務受付確認時間のいずれにも優先度が劣る。

6.6.2 例外

あらゆる正常な状況下において、取引トランザクションの成功と失敗は、応答メッセージやタイムアウトによって決定付けられる。ただし、応答役割が要求役割のいずれかで、トランザクションの事務処理に異変が起こることはある。

6.6.2.1 ControlException

ControlException は業務アクティビティ管理の中でエラー状況を合図する。このビジネスシグナルは、要求を発起した開始アクティビティに非同期的に返信される。この例外によって、取引トランザクションは終結しなければならない。これらのエラーは、確認、検証、認証、認可などのメッセージ交換機構を扱うものであり、メッセージの受付に応じて発生する。一般に、メッセージに適用される規則と拘束条件は、構造と構文とメッセージ要素値だけを扱う。

6.6.2.2 ビジネスプロトコル例外 (ProcessException)

ビジネスプロトコル例外(または ProcessException)は、業務アクティビティの中でエラー状況を合図する。このビジネスシグナルは、要求を発起した開始アクティビティに非同期的に返信される。この例外によって、取引トランザクションは終結しなければならない。これらのエラーは取引トランザクションの処理機構を扱うものであり、メッセージの確認・検証の後に発生する。一般に、メッセージに適用される規則と拘束条件は、メッセージ要素の意味情報と要求そのものの妥当性を扱うものであり、内容は応答役割の取引規則からは有効でない。この種の例外は普通、AcceptanceAcknowledgement が返された後に生成される。

ビジネスプロトコル例外によって取引トランザクションは終結する。ビジネスプロトコル例外は以下のとおりである。

- 受領の否定確認。メッセージの構造 / スキーマが無効。
- 受付の否定確認。取引規則の違反。
- 履行例外。要求された業務アクティビティは履行できない。

- 順序例外。取引文書やビジネスシグナルの順序またはタイプが不正。
- 構文例外。取引文書やビジネスシグナルの中に無効な句読法、語彙、または文法が存在する。
- 認可例外。取引トランザクションへの役割の関与が認可されていない。
- ビジネスプロセスコントロール例外。否認不可のための署名が要求される状況で、否認不可のための署名が取引文書にない。

取引トランザクションは、非常に原子的で決定論的な方法で定義される。取引トランザクションはいつも要求役割で開始され、そして要求役割で終結される。必要な応答またはシグナル、あるいはその両方を受領することで、またはタイムアウトを受け取ることで、要求役割は取引トランザクションが成功あるいは失敗したかを明確に判断することができる。この意味情報を維持するために、要求役割と応答役割は、制御失敗とビジネス失敗を以下のように別々に処理する。

ビジネスプロトコル例外の発生を認識した応答役割は、その例外を要求役割に返し、取引トランザクションを終結させる。ビジネス例外(受領および受付の否定確認を含む)が通知された場合、取引トランザクションは終結されなければならない。

ビジネスプロトコル例外の発生を認識した要求役割は、トランザクションを終結させるが、そのビジネス例外を応答役割に通知しない。その替わり要求役割は、障害の原因である取引文書要求を無効にする通知を別の取引トランザクションとして送信する。この新しいトランザクションは、最新のバイナリコラボレーションの延長として定義される場合や、失敗の通知を処理するよう明確に定義された新しいバイナリコラボレーションを開始する場合もある。

BSIは、関連した特別の例外を作成することで、以下のパラメータに明確に準拠する必要がある。要求役割と応答役割は、以下に示すように異なった処理を行う。

IsAuthorizationRequired

取引当事者が業務アクティビティを要求するための認可や業務アクティビティに応答するための認可を必要とする場合、送信取引当事者役割は交換される取引文書に署名しなければならないし、受信取引当事者役割はそのビジネスコントロールを確認し認可者を証明しなければならない。送信取引当事者役割による業務アクティビティの実行が許可されない場合には要求取引当事者が認可例外を発信しなければならない。要求取引当事者による応答業務アクティビティの実行が許可されない場合には、送信取引当事者が認可失敗通知を送信しなければならない。

IsNonRepudiationRequired

起源と内容の否認不可が要求される場合、業務アクティビティは、トランザクション取引当事者合意の中で相互に合意された期間にわたって、取引文書を原形で保管しなければならない。応答取引当事者は、送信取引当事者役割が取引文書を正しく引き渡さなかった場合に、ビジネスコントロール例外を発信しなければならない。要求取引当事者は、応答取引当事者が取引文書を正しく引き渡さなかった場合に、失敗ビジネスコントロール通知を発信しなければならない。

isNonRepudiationOfReceiptRequired.

両当事者は、要求取引文書の受領を相互に確認すること、そして受領が否認不能でなければならないことについて合意する。要求取引当事者は、応答取引当事者が取引文書を正しく引き渡さなかった場合に、失敗ビジネスコントロール通知（ことによると契約申込みを取り消す）を送信しなければならない。受領の否認不可に関する詳細は、『ebXML E-Commerce and Simple Negotiation Patterns』を参照のこと。

受領の否認不可には、以下の監査制御のためのデータが含まれる。
Verify responding role identity (認証する) – 要求取引文書を受け取った応答役割（個人または団体）の身元を確認する。
Verify content integrity – 取引文書要求の本来の内容の健全性を確認する。

isPositiveResponse

評価結果を TRUE あるいは FALSE で示す。TRUE の場合、この DocumentEnvelope は、要求に肯定的に応答する意図があると認識される。DocumentEnvelope のこのパラメータに使用される値は、DocumentEnvelope の送信者が持つ、関連するトランザクションに対する意図を示す。しかしこの値は受信者に対する拘束力を持たず、またガード表現を使ってのトランザクションの成功または失敗の操作を上書きすることもしない。

要求役割がこれらの表現内容を評価して失敗と判断した場合、要求役割は取引トランザクションを「ロールバック」し、失敗の通知を送信する。

6.7 実行時コラボレーション意味情報

ebXML のコラボレーション意味情報は、複数当事者コラボレーションとバイナリコラボレーションとの関係、バイナリコラボレーションの再帰層間の関係、バイナリコラボレーションにおけるトランザクション間の振り付けを含む。今日 BSI にトランザクションの状態を管理することが期待されるように、BSI ソフトウェアは時が経つにつれ、コラボレーションの状態を監視・管理する状態にまで

進化することが予想される。そのような能力が必要とされるには、しばらく時間がかかるだろう。

6.8 ebXML ビジネスプロセス仕様スキーマが実装される状況

ebXML 取引コラボレーションの中で役割を実行する形にソフトウェアを指定するときは必ず、『ebXML ビジネスプロセス仕様スキーマ』を使用する必要がある。具体的には、取引当事者のコラボレーションプロトコルプロファイルとコラボレーションプロトコル合意書を作成するためのビジネスプロセスと文書仕様が『ebXML ビジネスプロセス仕様スキーマ』で提供される。

ただし仕様スキーマは、電子商取引コラボレーションを指定するために利用される場合がある。たとえば非営利団体間や企業の内部で行われるトランザクションコラボレーションを定義するなど、非営利コラボレーションにも利用される場合がある。

7 UML 要素指定

これより、「UML 仕様スキーマ」の全仕様要素を以下のとおりに分類して考察する。

- 取引コラボレーション
 - 複数当事者
 - バイナリ
- 取引トランザクション
- 文書フロー
- 振り付け

7.1 取引コラボレーション

7.1.1 MultiPartyCollaboration

「複数当事者コラボレーション」は「バイナリコラボレーション」の合成である。

「複数当事者コラボレーション」を構成する「取引当事者役割」は、バイナリコラボレーションの中で互いに役割を実行する。

タグ付値：

name. MultiPartyCollaboration の名前を定義する。

Figure 1. 関連：

partners 複数当事者コラボレーションは複数の BusinessPartnerRole を持つ。

Figure 2. ウェルフォームドルール :

あらゆる複数当事者コラボレーションは、バイナリコラボレーションから合成されなければならない。

7.1.2 BusinessPartnerRole

BusinessPartnerRole は、MultiPartyCollaboration の中で取引当事者によって実行される役割である。「複数当事者コラボレーション」を構成する「バイナリコラボレーション」の各々につき、BusinessPartnerRole は最高 1 個の「認定役割」を実行する。

Figure 3. タグ付値 :

name. 顧客や供給者など、複数当事者取引コラボレーションの全体の中で取引当事者によって実行される役割の名前

Figure 4. 関連 :

performers. バイナリ取引コラボレーションの中で取引当事者によって実行される「認定役割」。

transitions 複数のバイナリコラボレーションにまたがるアクティビティ間の遷移 (この BusinessPartnerRole によって管理される遷移)

collaboration 「取引当事者役割」は、1 つの複数当事者コラボレーションに使用される。

Figure 5. ウェルフォームドルール :

1 取引当事者は 1 つの業務アクティビティの中で両方の役割を実行してはならない。

7.1.3 Performs

Performs は、BusinessPartnerRole とそれが実行する Role との関係を明示的に表現するものである。これは、複数当事者コラボレーション内での「認定役割」の使用を指定する。

Figure 6. タグ付値 :

なし

Figure 7. 関連 :

performedBy Performs のインスタンスは、ただ 1 つの BusinessPartnerRole によって実行される。

authorizedRole Business PartnerRole によって実行される「認定役割」。

Figure 8. ウェルフォームドルール :

AuthorizedRole を実行する Performs ごとに、対極の AuthorizedRole を実行する Performs が存在しなければならないし、存在しなければ「複数当事者コラボレーション」は不完全である。

7.1.4 AuthorizedRole

「認定役割」とは、たとえば買付注文の要求を送ることを買手に許可したり、買付注文の承諾を送ることを売手に許可するように、要求 / 応答の送信を許可された役割である。

Figure 9. タグ付値 :

<i>name</i>	AuthorizedRole の名前を、バイナリコラボレーション内で独自に定義する。
<i>isInitiator</i>	この認可を受けた役割が、関連するバイナリコラボレーションの開始者であるかを判断する Boolean。

Figure 10. 関連 :

<i>performers</i>	AuthorizedRole は 1 つまたは複数の実行者、すなわち複数当事者コラボレーションにおける「取引当事者役割」によって使用される場合がある。
<i>from</i>	AuthorizedRole は業務アクティビティの中で開始者になる場合がある。
<i>to</i>	AuthorizedRole は業務アクティビティの中で応答者になる場合がある。
<i>collaboration</i>	AuthorizedRole はただ 1 つの BinaryCollaboration の中で存在する。

Figure 11. ウェルフォームドルール :

AuthorizedRole は、1 つの取引トランザクションの中で要求者と応答者の両方にはなりえない。

AuthorizedRole は、1 つのバイナリ取引コラボレーションの中で開始者と応答者の両方にはなりえない。

7.1.5 BinaryCollaboration

「バイナリコラボレーション」は、2 認定役割間の相互作用プロトコルを定義する。

「バイナリコラボレーション」は、コラボレーション役割間における振り付けされた状態群である。取引トランザクションやその他のコラボレーションを実行するアクティビティは、一種の状態である。

「バイナリコラボレーション」では、2 役割間で1つまたは複数の取引トランザクションアクティビティを振り付けする。

「バイナリコラボレーション」は原子的トランザクションではなく、トランザクションロールバックが要求される状況では使用しないほうがよい。

Figure 12. タグ付値 :

<i>name</i>	BinaryCollaboration の名前を定義する。
<i>timeToPerform</i>	最初のアクティビティの開始から、このコラボレーション全体が終結しなければならなくなるまでの時間。
<i>preCondition</i>	コラボレーションの開始前に必要とされる、このコラボレーション外の状態の記述。
<i>postCondition</i>	このコラボレーションの実行前には存在しないが、このコラボレーション実行後に存在する状態の記述。
<i>beginsWhen</i>	通常は当該コラボレーションを開始させる、当該コラボレーション外のイベントの記述。
<i>endsWhen</i>	通常は当該コラボレーションを終結させる、当該コラボレーション外のイベントの記述。
<i>Pattern</i>	当該バイナリコラボレーションの基礎となるパターンのオプション名

Figure 13. 関連 :

<i>role</i>	バイナリコラボレーションは2つの認定役割を伴う。1つは開始役割として、もう1つは応答役割として指定されている。
<i>states</i>	バイナリコラボレーションは1つまたは複数の状態を伴い、状態には '静的' なものと作用状態がある。
<i>usedBy</i>	コラボレーションアクティビティを通じて1つのバイナリコラボレーションを別のバイナリコラボレーションの中で使用する場合がある。
<i>transitions</i>	当該バイナリコラボレーションにおけるアクティビティ間の遷移

Figure 14. ウェルフォームドルール :

なし

7.1.6 業務アクティビティ

業務アクティビティは、バイナリコラボレーションの中の1つの作用状態である。トランザクションの実行アクティビティを指定する `BusinessTransactionActivity` や、別のバイナリコラボレーションを指定する `CollaborationActivity` のスーパータイプにあたる。

Figure 15. スーパータイプ :

`BusinessTransactionActivity`, `CollaborationActivity`

Figure 16. サブタイプ :

`BusinessState`

Figure 17.

Figure 18. タグ付値 :

name. アクティビティの名前を、バイナリコラボレーション内で独自に定義する。

Figure 19. 関連 :

from 親バイナリコラボレーション内の `AuthorizedRoles` の1つと一致しなければならず、`BinaryCollaboration` の開始者、あるいはこのアクティビティにより実行された `BusinessTransaction` となる。

to 親バイナリコラボレーション内の `AuthorizedRoles` の1つと一致しなければならず、`BinaryCollaboration` の応答者、あるいはこのアクティビティにより実行された `BusinessTransaction` となる。

Figure 20. ウェルフォームドルール :

なし

7.1.7 `BusinessTransactionActivity`

取引トランザクションアクティビティは、バイナリコラボレーションにおける取引トランザクションの使用を定義する。

取引トランザクションアクティビティは、所定の取引トランザクションを実行する業務アクティビティである。`isConcurrent` プロパティが `true` なら、取引トランザクションアクティビティを複数回にわたって実行できる。

Figure 21. サブタイプ :

BusinessActivity

Figure 22.

Figure 23. タグ付値 :

timeToPerform. 要求の送信から、両当事者がこの取引トランザクションアクティビティによって実行された取引トランザクションを終結することに合意するまでの時間。

isConcurrent. BusinessTransactionActivity が並行実行可能であれば、関連した BusinessTransaction の 1 つ以上のインスタンスを、この BusinessTransactionActivity の実行の一部として実行することができる。

isLegallyBinding 当該アクティビティによって実行される取引トランザクションは、トランザクション取引当事者によって拘束力があるものと見なされる。デフォルト値は True。

Figure 24. 関連 :

uses. 取引トランザクションアクティビティは、ただ 1 つの取引トランザクションを実行（使用）する。

Figure 25. ウェルフォームドルール :

なし

7.1.8 CollaborationActivity

コラボレーションアクティビティとは、別のバイナリコラボレーションの中でバイナリコラボレーションを実行するアクティビティである。

Figure 26. サブタイプ :

BusinessActivity

Figure 27.

Figure 28. タグ付値 :

なし（継承されるものを除く）

Figure 29. 関連 :

uses. コラボレーションアクティビティは、ただ1つのバイナリコラボレーションを使用する。

Figure 30. ウェルフォームドルール :

バイナリコラボレーションはそれ自身を再利用できない。

7.2 取引トランザクション

7.2.1 BusinessTransaction

取引トランザクションは2 営利取引当事者間で行われるビジネス情報・ビジネスシグナルのやり取りであり、そのやり取りは、合意された形式と順序と期間に従って行わなければならない。何らかの合意に違反する場合はトランザクションが終結し、すべてのビジネス情報・ビジネスシグナル交換を破棄しなければならない。取引トランザクションは、オンライン申込み / 承諾取引約定の作成などでは公式であり、製品発表の配布などでは非公式である。

Figure 31. タグ付値 :

name 取引トランザクションの名前を定義する。

isGuaranteedDeliveryRequired. 両当事者は、引渡が保証される伝送の使用に合意しなければならない。

preCondition このトランザクションの開始前に必要とされる、このトランザクション外の状態の記述。

postCondition このトランザクションの開始前には存在しないが、このトランザクションの実行後に存在する状態の記述。

beginsWhen 通常は当該トランザクションを開始させる、当該トランザクション外のイベントの記述。

endsWhen 通常は当該トランザクションを終結させる、当該トランザクション外のイベントの記述。

pattern 当該取引トランザクションの基礎となるパターンのオプション名

Figure 32. ウェルフォームドルール :

なし

7.2.2 BusinessActivity

「BusinessActivity」は抽象スーパークラスである。「要求される業務アクティビティ」と「応答業務アクティビティ」の双方に共通の属性を保持する。

Figure 33. スーパータイプ :

Requesting BusinessActivity, Responding BusinessActivity

Figure 34. タグ付値 :

name RequestingBusinessTransaction あるいは RespondingBusinessTransaction の名前をサブタイプに基づいて定義する。

IsAuthorizationRequired 受信当事者は、承認された発起者の一覧と照合して発起者の身元確認を行わなければならない。このパラメータは、送信者側で指定される(詳細は、「認定セキュリティ」の節を参照のこと)。

IsNonRepudiationRequired 受信当事者は受領確認を送信する前に、要求文書が不正に変更されていないか(読み取り不可能、または理解不可能であるなど)を確認しなければならない。このパラメータは、送信者側で指定される(詳細は、「コア取引トランザクション意味情報」の節を参照のこと)。

isNonRepudiationOfReceiptRequired.

受信当事者が署名された受取を返送し、元の送信者がこの受取のコピーを保存することを要求する。このパラメータは、送信者側で指定される(詳細は、「否認不可」の節を参照のこと)。

timeToAcknowledgeReceipt 受信役割が取引文書の受領を確認するための時間。このパラメータは、送信者側で指定される(詳細は、「コア取引トランザクション意味情報」の節を参照のこと)。

isIntelligibleCheckRequired 受信当事者は受領確認を送信する前に、要求文書が不正に変更されていないか(読み取り不能、または理解不能であるなど)を確認しなければならない。このパラメータは、送信者側で指定される(詳細は、「コア取引トランザクション意味情報」の節を参照のこと)。

Figure 35. 関連 :

なし

Figure 36. ウェルフォームドルール :

なし

7.2.3 Requesting BusinessActivity

Requesting BusinessActivity とは、別の役割に対して開始を要求する役割によって実行される業務アクティビティである。

Requesting BusinessActivity では、要求を運ぶ「文書フロー」を指定する。

Figure 37. サブタイプ :

BusinessAction

Figure 38.

Tagged Values:*timeToAcknowledgeAcceptance* 応答役割が取引文書の事務受付を形式的に確認するのにかかる時間。このパラメータは、要求者側で指定される(詳細は、「コア取引トランザクション意味情報」の節を参照のこと)。

Figure 39. 関連 :

transaction 要求アクティビティは、ただ1つの取引トランザクションの中で実行される

documentEnvelope 要求アクティビティは、ただ1つの文書エンベロープを送信する。

Figure 40. ウェルフォームドルール :

なし

7.2.4 Responding BusinessActivity

Responding BusinessActivity は、取引トランザクション内の応答役割によって実行される業務アクティビティであり、応答を行う文書エンベロープの指定をする。

複数の応答「文書フロー」を定義する場合があるが、実際のトランザクションインスタンスではその内のただ1つが発生する。

Figure 41. サブタイプ :

BusinessAction

Figure 42.

Figure 43. **タグ付値 :**

なし、「BusinessActivity」からの継承を除く

Figure 44. **関連 :**

transaction 応答アクティビティは、ただ1つの取引トランザクションの中で実行される

documentEnvelope 応答アクティビティは、最大1つの文書エンベロープを送信する。

Figure 45. **ウェルフォームドルール :**

なし

7.3 文書フロー

7.3.1 文書セキュリティ

DocumentSecurity は DocumentEnvelope、Attachment 属性に関連する抽象スーパークラスセキュリティを保持する。

Figure 46. **スーパータイプ :**

DocumentEnvelope and Attachment

Figure 47. **タグ付値 :**

IsAuthenticated 文書エンティティにデジタル証明が付随され、署名者の身元が証明される(詳細は、「文書セキュリティ」の節を参照のこと)。

IsConfidential 許可されない者が情報を閲覧できないように、情報エンティティを暗号化する(詳細は、「文書セキュリティ」の節を参照のこと)。

isTamperProof 情報エンティティには暗号化されたメッセージダイジェストが付随する。このメッセージダイジェストを利用すれば、メッセージに対する不正変更の有無をチェックすることができる。このチェックを実行するには、文書エンティティにデジタル署名(送信者のデジタル証明と暗号化されたメッセージダイジェスト)を付随させる必要がある。

Figure 48. **関連 :**

Figure 49. ウェルフォームドルール :

なし

7.3.2 文書エンベロープ

文書エンベロープにより、取引トランザクション内で2つの役割の間のビジネス情報交換が行われる。1つの文書エンベロープが要求役割から応答役割に要求を伝達し、もう1つの文書エンベロープが応答役割から要求役割に応答(応答があった場合)を伝達する。

Figure 50. サブタイプ :

DocumentSecurity

Figure 51. タグ付値 :

isPositiveResponse TRUE または FALSE が使用される。
TRUE の場合、この DocumentEnvelope は要求に対して肯定的な応答を意図していると認識される。このパラメータは、応答エンベロープにのみ使用可能である。この値は、受信者に対して拘束力を持たず、またトランザクションのガード表現を使ってのトランザクションの成功または失敗の操作を上書きすることもない。

Figure 52. 関連 :

schema 文書型は、最大 1 つのスキーマの中に存在する。
documentFlow 文書型は、複数の文書フローの中に存在できる。
attachment 1 つの文書型で多数の添付文書を指定できる。

Figure 53. ウェルフォームドルール :

1つの文書エンベロープは、1つの要求アクティビティと1つの応答アクティビティに付随する。
IsPositiveResponse は、要求アクティビティによって送信された DocumentEnvelope に使用可能なパラメータではない。

7.3.3 BusinessDocument

BusinessDocument は文書の一般名である。

Figure 54. タグ付値：

<i>name</i>	取引文書の一般名を、該当のビジネスプロセス仕様内で表示されているとおりに定義する。
<i>conditionExpression</i>	取引文書には1つの条件表現が含まれる場合がある。これにより、該当の取引文書がそのエンベロープにとって有効であるかどうかを判断することができる。

Figure 55. 関連：

<i>documentEnvelope</i>	1つの取引文書を複数の文書エンベロープに配置することができる。
<i>attachment</i>	取引文書を使用して、多くの添付文書のタイプを指定することができる。

Figure 56. ウェルフォームドルール：

なし

7.3.4 Attachment

Attachment とは、文書エンベロープ内の BusinessDocument にオプションとして添付された文書である。

Figure 57. サブタイプ：

DocumentSecurity

Figure 58.

Figure 59. タグ付値：

<i>name</i>	添付文書の名前を定義する。
<i>mimeType</i>	当該「添付文書」の有効 MIME（多目的インターネットメッセージ拡張）タイプを定義する
<i>specification</i>	当該添付文書の外部記述ソースへの参照
<i>version</i>	「添付文書」のバージョン

Figure 60. 関連：

<i>documentEnvelope</i>	1つの文書エンベロープにつき1つの「添付文書」が含まれる。
<i>businessDocument</i>	「添付文書」は BusinessDocument によって定義される。取引文書によって定義され

ていなければ、MIME タイプおよび仕様がこのタイプを示す。

Figure 61. ウェルフォームドルール :

なし

7.4 コラボレーションの中の振り付け

7.4.1 BusinessState

ビジネス状態はバイナリコラボレーションの状態である。Start および CompletionState は、あるアクティビティの直前または直後のスナップショットである。BusinessActivity はアクティビティ中の作用状態を示す。Fork および Join は、複数のアクティビティの分岐、またはそれらの結合を反映する。

Figure 62. スーパータイプ :

Start, CompletionState, Fork, Join, BusinessActivity

Figure 63.

Figure 64. タグ付値 :

なし

Figure 65. 関連 :

<i>collaboration</i>	ビジネス状態はただ 1 つの取引コラボレーションに属する
<i>entering</i>	この状態への進入を表わす遷移
<i>exiting</i>	この状態からの退出を表わす遷移

Figure 66. ウェルフォームドルール :

なし

7.4.2 Transition

遷移とは、バイナリコラボレーションにおける 2 ビジネス状態間の遷移である。

振り付けは、ビジネス状態間の遷移として表現される。

Figure 67. タグ付値 :

onInitiation これにより指定されるのは、これが入り子になった

BusinessTransactionActivity であることと、関連トランザクションからの要求を受け取ってから、該当トランザクションに戻る前に2つ目のアクティビティを実行することで、元の要求者に応答が送信されることである。

conditionGuard 前のトランザクションの状態への参照。
Success、BusinessFailure、TechnicalFailure、または AnyFailure の固定値である。

Figure 68. *conditionExpression* 遷移には1つの条件表現が含まれる場合がある。これにより、該当の遷移が実行される必要があるかどうか判断される。

Figure 69. **関連 :**

in この遷移が進入する先のビジネス状態
out この遷移が退出するところのビジネス状態

Figure 70. **ウェルフォームドルール :**

1つの遷移が同じ状態に出入することはできない。

7.4.3 Start

「バイナリコラボレーション」の開始状態。「バイナリコラボレーション」には少なくとも1つの開始アクティビティが存在する必要がある。まったく定義されない場合は、すべてのアクティビティが進入可能点として見なされる。

Figure 71. **サブタイプ :**

BusinessState

Figure 72.

Figure 73. **タグ付値 :**

なし

Figure 74. **関連 :**

なし

Figure 75. **ウェルフォームドルール :**

なし

7.4.4 TerminalState

成功と失敗によって細分化される、バイナリコラボレーションの終了状態

Figure 76. スーパータイプ :

Success, Failure

Figure 77. サブタイプ :

BusinessState

Figure 78.

Figure 79. タグ付値 :

なし

Figure 80. 関連 :

なし

Figure 81. ウェルフォームドルール :

なし

7.4.5 Success

アクティビティからの遷移としてバイナリコラボレーションの良好終了を定義する、TerminalState のサブタイプ。

Figure 82. サブタイプ :

CompletionState

Figure 83.

Figure 84. タグ付値 :

conditionExpression 成功の状態には、該当の遷移に対して 1 つの条件表現が含まれる場合がある。これにより、該当の遷移が実行される必要があるかどうか判断される。

Figure 85. 関連 :

継承されたもの以外はない

Figure 86. ウェルフォームドルール :

すべてのアクティビティ「バイナリコラボレーション」は、少なくとも 1 つの成功を持つ必要がある。

7.4.6 Failure

アクティビティからの遷移としてバイナリコラボレーションの異常終了を定義する、TerminalState のサブタイプ。

Figure 87. サブタイプ :

CompletionState

Figure 88.

Figure 89. タグ付値 :

conditionExpression 失敗の状態には、該当の遷移に対する 1 つの条件表現が含まれる場合がある。これにより、該当の遷移が実行される必要があるかどうか判断される。

Figure 90. 関連 :

継承されたもの以外はない

Figure 91. ウェルフォームドルール :

すべての「バイナリコラボレーション」は、少なくとも 1 つの失敗を持つ必要がある。

7.4.7 Fork

Fork は、1 つの受信遷移および複数の送信遷移を含んだ状態である。送信遷移によって指定されたすべてのアクティビティは、並行実行される。

Figure 92. サブタイプ :

BusinessState

Figure 93. タグ付値 :

Name Fork 状態の名前を定義する。

Figure 94. 関連 :

なし

Figure 95. ウェルフォームドルール :

なし

7.4.8 Join

あるアクティビティが、他の 1 つ以上のアクティビティの完了を待っているビジネス状態。前に分岐されたアクティビティを再び結合する時点を定義する。

Figure 96. サブタイプ :

BusinessState

Figure 97. タグ付値 :

<i>Name</i>	Join 状態の名前を定義する。
<i>waitForAll</i>	Join 状態が、すべての受信遷移が完了するのを待つ必要があるかどうかを示す Boolean 値。 TRUE の場合、すべての受信遷移の完了を待ち、 FALSE の場合、最初の受信遷移の処理を開始する。

Figure 98. 関連 :

なし

Figure 99. ウェルフォームドルール :

なし

7.5 定義と範囲

ebXML バイナリコラボレーションの中で役割を実行するようにソフトウェアを指定する時は必ず、『ebXML ビジネスプロセス仕様スキーマ』を使用する必要がある。具体的には、トランザクション取引当事者の「コラボレーションプロトコルプロファイル」と「コラボレーションプロトコル合意書」を作成するためのビジネスプロセスと文書仕様が「ebXML 仕様スキーマ」から提供される。ビジネスプロセスと情報モデルの表現を、トランザクション取引当事者の「コラボレーションプロトコルプロファイル」と「コラボレーションプロトコル合意書」に直接編入できる形に正しく拘束するため、指定規則群が確立された。

7.6 コラボレーション指定規則

以下の規則を用いて、標準解析の他に、『ebXML ビジネスプロセス仕様』における要素の属性値を適切に拘束することができる。

取引トランザクション

- [0] 否認不可が要求される場合、投入される取引文書や返却される取引文書は、不正操作耐性エンティティでなければならない。
- [1] 認可が要求される場合、投入される取引文書とビジネスシグナルは、認可された安全エンティティか不正操作耐性の安全エンティティでなければならない。
- [2] 受領確認時間と受付確認時間の両プロパティが値を持つ場合は、受付確認時間よりも受領確認時間を減らさなければならない。

timeToAcknowledgeReceipt < timeToAcknowledgeAcceptance

- [3] 受付確認時間が null の場合、アクティビティ実行時間は受領確認時間以上にしなければならない。
- [4] 受領確認時間か受付確認時間のいずれかが null でない場合、トランザクション実行時間は null にできない。
- [5] 受領の否認不可が要求される場合、受領確認時間は null にできない。
- [6] 受領確認時間と受付確認時間と実行時間のすべてを 0 にすることはできない。
- [7] 要求される業務アクティビティ側で否認不可が要求される場合は、応答取引文書が存在しなければならない。

Requesting BusinessActivity

- [8] ソース状態頂点が初期擬似状態となる入力遷移が 1 つ存在しなければならない。
- [9] ターゲット状態頂点が最終状態（アクティビティが正常に実行される場合のマシン状態を表わす）となる出力遷移が 1 つ存在しなければならない。
- [10] ターゲット状態頂点が最終状態（プロセスコントロール例外のためアクティビティが正常に実行されない場合のマシン状態を表わす）となる出力遷移が 1 つ存在しなければならない。
- [11] ターゲット状態頂点が最終状態（ビジネスプロセス例外のためアクティビティが正常に実行されない場合のマシン状態を羅和す）となる出力遷移が 1 つ存在しなければならない。
- [12] 転じて応答業務アクティビティへの入力となる、出力「文書フロー」が 1 つ存在しなければならない。
- [13] 転じて要求される業務アクティビティへの入力となる、要求からの出力「文書フロー」が 0 個または 1 個存在しなければならない。

Responding BusinessActivity

- [14] 転じて要求される業務アクティビティからの入力遷移 1 個を持つ、「文書フロー」からの入力遷移が 1 つ存在しなければならない。
- [15] 転じて要求される業務アクティビティに至る出力遷移 1 個を持つ、「文書フロー」からの出力遷移が 0 個または 1 個存在しなければならない。

取引コラボレーション

- [16] 1 つの「取引当事者役割」で、同じ取引トランザクションアクティビティの開始役割と応答役割の両方は提供できない。

8 ebXML ビジネスプロセス仕様スキーマ – (DTD)

本節では、『仕様スキーマ』の DTD および XML スキーマのバージョンを解説する。DTD と XML スキーマとの相違点は少ないため、要素は 1 度説明するにとどめ、必要に応じて相違点を指摘する。この解説には、以下の内容が含まれる。

- 付録 A に一覧表示された XML ビジネスプロセス仕様の一例。
- 付録 B の DTD 一覧、および付録 C の XML スキーマ。
- 定義と親子関係を含んだすべての要素の一覧表。
- 定義と親要素の関係を含んだすべての属性の一覧表。
- すべての要素の一覧表。各要素には、仕様スキーマの UML バージョンにおいて対応するクラスへの相互参照が含まれる。
- 名前空間および要素参照に関する規則。

8.1 DTD 用文書

本節では DTD について解説する。DTD は UML モデルから派生した。UML クラスと DTD 要素間の関係は、後に本書中で別に説明する。

属性定義以外の全体の構造を以下に示す。

[ProcessSpecification](#) (Documentation*, SubstitutionSet*, (Include | BusinessDocument | ProcessSpecification | Package | BinaryCollaboration | BusinessTransaction | MultiPartyCollaboration)*)

[Documentation](#)()

[SubstitutionSet](#) (DocumentSubstitution | AttributeSubstitution | Documentation)*

[DocumentSubstitution](#) (Documentation*)

[AttributeSubstitution](#) (Documentation*)

[Include](#)(Documentation*)

[BusinessDocument](#)(ConditionExpression? | Documentation*)

[ConditionExpression](#) (Documentation*)

[Package](#)(Documentation*, (Package | BinaryCollaboration | BusinessTransaction | MultiPartyCollaboration)*)

[BinaryCollaboration](#)(Documentation*, InitiatingRole, RespondingRole, (Documentation | Start | Transition | Success | Failure | BusinessTransactionActivity | CollaborationActivity | Fork | Join)*)

[InitiatingRole](#)(Documentation*)

[RespondingRole](#)(Documentation*)

[Start](#)(Documentation*)

[Transition](#)(ConditionExpression | Documentation)*

[Success](#)(ConditionExpression | Documentation)*

[Failure](#)(ConditionExpression | Documentation)*

[Fork](#)(Documentation*)

[Join](#)(Documentation*)

[BusinessTransactionActivity](#)(Documentation*)

[CollaborationActivity](#)(Documentation*)

[BusinessTransaction](#)(Documentation*, Requesting BusinessActivity, Responding BusinessActivity)

[Requesting BusinessActivity](#) (Documentation*, DocumentEnvelope)

[Responding BusinessActivity](#)(Documentation*, DocumentEnvelope*)
[MultiPartyCollaboration](#)(Documentation*, BusinessPartnerRole*)
[BusinessPartnerRole](#)(Documentation*, Performs*, Transition*)
[Performs](#)(Documentation*)
[Transition](#)(Documentation*)

a. Attachment

XML 要素名 : Attachment

DTD 宣言 :

```
<!ELEMENT Attachment (DocumentType?)>  
<!ATTLIST Attachment  
    name CDATA #REQUIRED  
    mimeType CDATA #REQUIRED  
    spec CDATA #REQUIRED  
    isConfidential (true | false) "false"  
    isTamperProof (true | false) "false"  
    isAuthenticated (true | false) "false">
```

定義 :

「文書エンベロープ」における DocumentType のオプション添付物。

親要素 :

- DocumentEnvelope

属性：

属性名	定義	デフォルト値
name	添付文書の名前を定義する。	必須入力
nameID	名前の XML ID バージョン。	オプション入力
businessDocument	添付文書のタイプは、BusinessDocument によって定義される。取引文書によって定義されていない場合は、MIME タイプあるいは仕様によって定義される。	必須入力
businessDocumentIDRef	businessDocument の XML IDREF バージョン。	オプション入力
isAuthenticated	文書エンティティにデジタル証明が付随し、署名者の身元を証明する。 (詳細は、「文書セキュリティ」の節を参照のこと。)	false {true, false}
isConfidential	許可されない者は情報を閲覧できなくするため、情報エンティティを暗号化する。 (詳細は、「文書セキュリティ」の節を参照のこと。)	false {true, false}
isTamperProof	暗号化されたメッセージダイジェストが情報エンティティに付随する。これを利用すれば、メッセージに対する不正変更の有無をチェックできる。それには、文書エンティティにデジタル署名（送信者のデジタル証明と暗号化されたメッセージダイジェスト）を付随させる必要がある。 (詳細は、「文書セキュリティ」の節を参照のこと。)	false 有効値 {true, false}
mimeType	当該「添付文書」の有効 MIME（多目的インターネットメール拡張）タイプを定義する	オプション入力 例： 'application/pdf'
specification	当該添付文書の外部記述ソースを指す参照	オプション入力
version	添付文書のバージョン。	オプション入力

b. AttributeSubstitution

要素名：AttributeSubstitution

DTD 宣言：

```
<!ELEMENT AttributeSubstitution (Documentation*)>
<!ATTLIST AttributeSubstitution
  attributeName CDATA #IMPLIED
  value CDATA #IMPLIED
>
```

定義：

AttributeSubstitution を使うことで、既存のプロセス仕様における一部の属性値を、別の 1 つの属性値で置き換えられるよう指定することができる。

親：

- SubstitutionSet

属性：

属性名	定義	デフォルト値
attributeName	置換セットの範囲内にある要素の属性名。	必須入力
value	現行の属性値を置き換えることになる値。	必須入力

c. Binary Collaboration

XML 要素名：BinaryCollaboration

DTD 宣言：

```
<!ELEMENT BinaryCollaboration (Documentation*,
AuthorizedRole, AuthorizedRole,
(Documentation* | Start | Transition | Success |
Failure | Split | Sync | BusinessTransactionActivity |
CollaborationActivity)*)>
<!ATTLIST BinaryCollaboration
  name ID #REQUIRED
  beginsWhen CDATA #IMPLIED
  endsWhen CDATA #IMPLIED
  requires CDATA #IMPLIED
  resultsIn CDATA #IMPLIED
  timeToPerform CDATA #IMPLIED
>
```

定義：

「バイナリコラボレーション」は、2 認定役割間の相互作用プロトコルを定義する。

「バイナリコラボレーション」は、コラボレーション役割間における振り付けされた状態群である。取引トランザクションやその他のコラボレーションを実行するアクティビティは、一種の状態である。

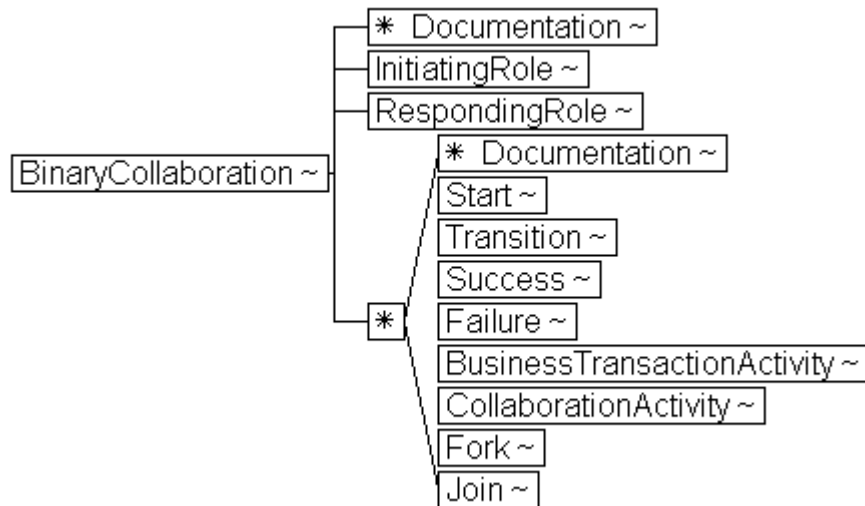
「バイナリコラボレーション」は、2 役割間における 1 つまたは複数の取引トランザクションアクティビティを振り付けする。

「バイナリコラボレーション」は原子的トランザクションではなく、トランザクションロールバックが要求される状況では使用しないほうがよい。

親：

- ProcessSpecification
- Package

階層モデル：

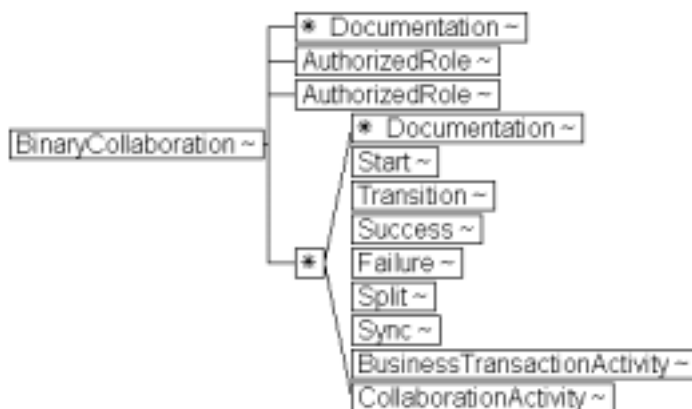


属性：

属性名	定義	デフォルト値
name	バイナリコラボレーションの名前を定義する。	必須入力
nameID	名前の XML ID バージョン。	オプション
beginsWhen	通常は当該コラボレーションを開始させる、当該コラボレーション外のイベントの記述。	オプション入力
endsWhen	通常は当該コラボレーションを終結させる、当該コラボレーション外のイベントの記述。	オプション入力
pattern	該当のバイナリコラボレーションの基盤となるパターンへのオプションの参照。XML スキーマのバージョンでは、データ型は xsd:anyURI である。	
preCondition	当該コラボレーションが終結する前に要求される、当該コラボレーション外	オプション入力

	の状態の記述。	
postCondition	当該コラボレーションを実行する前には存在しないが、当該コラボレーション/トランザクションを実行した結果として存在する状態の記述。	オプション入力
timeToPerform	最初のアクティビティの開始から始まる期間であり、その期間内に当該コラボレーションの全体が終結しなければならない。	オプション入力

階層モデル：



d. BusinessDocument

要素名： BusinessDocument

DTD 宣言：

```
<!ELEMENT BusinessDocument (ConditionExpression?,
Documentation*) >
<!ATTLIST BusinessDocument
    name CDATA #REQUIRED
    nameID ID #IMPLIED
    specificationLocation CDATA #IMPLIED
    specificationElement CDATA #IMPLIED>
```

定義：

BusinessDocument は文書の一般名である。

親：

- ProcessSpecification
- Attachment

属性：

属性名	定義	デフォルト値
name	取引文書の一般名を、該当のビジネスプロセス仕様内で表示されているとおりに定義する。	必須入力
nameID	名前の XML ID バージョン。	オプション
specificationLocation	外部スキーマ定義ソースへの参照。XML スキーマのバージョンにおいては、データ型は xsd:anyURI である。	オプション

specificati onElement	該当文書と定義するスキーマ定義内における要素への参照。	オプション
----------------------------------	-----------------------------	-------

e: Business Partner Role

要素名: BusinessPartnerRole

DTD 宣言:

```
<!ELEMENT BusinessPartnerRole (Documentation*, Performs*,
                                Transition*)>
<!ATTLIST BusinessPartnerRole
            name          CDATA          #REQUIRED>
```

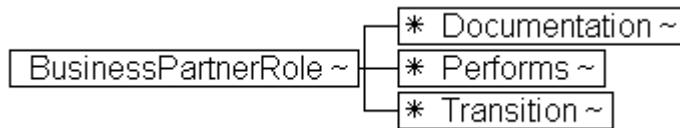
定義:

BusinessPartnerRole は、MultiPartyCollaboration の中で取引当事者によって実行される役割である。「複数当事者コラボレーション」を構成する「バイナリコラボレーション」の各々につき、BusinessPartnerRole は最高 1 個の「認定役割」を実行する。

親:

- MultiPartyCollaboration

階層モデル:



属性:

属性名	定義	デフォルト値
Name	複数当事者取引コラボレーション全体において取引当事者によって実行される役割の名前を定義する。例: 顧客または販売業者。	必須入力
nameID	名前の XML ID バージョン。	オプション

f. Business Transaction

要素名: BusinessTransaction

内容モデル:

```
<!ELEMENT BusinessTransaction (Documentation*,
                                RequestingBusinessActivity, RespondingBusinessActivity)>
<!ATTLIST BusinessTransaction
            name          CDATA          #REQUIRED
            nameID        ID            #IMPLIED
            pattern        CDATA          #IMPLIED
```

```

beginsWhen          CDATA #IMPLIED
endsWhen            CDATA #IMPLIED
isGuaranteedDeliveryRequired (true | false) false
precondition        CDATA #IMPLIED
postCondition       CDATA #IMPLIED>

```

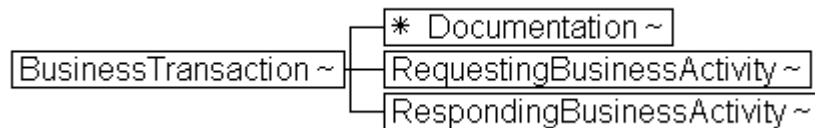
定義：

取引トランザクションは2営利取引当事者間で行われるビジネス情報・ビジネスシグナルのやり取りであり、そのやり取りは、合意された形式と順序と期間に従って行わなければならない。何らかの合意に違反する場合はトランザクションが終結し、すべてのビジネス情報・ビジネスシグナル交換を破棄しなければならない。「取引トランザクション」は、オンライン申込み/承諾取引約定の作成などでは公式であり、製品発表の配布などでは非公式である。

親：

- ProcessSpecification
- Package

階層モデル：



属性：

属性名	定義	デフォルト値
name	取引トランザクションの名前を定義する。	必須入力
nameID	名前のXML IDバージョン。	オプション
pattern	該当のトランザクションの基盤となるパターンへのオプションの参照。 XMLスキーマのバージョンにおけるデータ型はxsd:anyURIである。	オプション
beginsWhen	通常は当該トランザクションを開始させる、当該トランザクション外のイ	オプション入力

	イベントの記述。	
endsWhen	通常は当該トランザクションを終結させる、当該トランザクション外のイベントの記述。	オプション入力
isGuaranteedDeliveryRequired	両当事者は、引渡が保証される伝送の使用に合意しなければならない。	false 有効値： {true, false}
preCondition	当該トランザクションが終結する前に要求される、当該トランザクション外の状態の記述。	オプション入力
postCondition	当該トランザクションを実行する前には存在しないが、当該トランザクションを実行した結果として存在する状態の記述。	オプション入力

g. Business Transaction Activity

要素名： BusinessTransactionActivity

内容モデル：

```
<!ELEMENT BusinessTransactionActivity (Documentation*)>
<!ATTLIST BusinessTransactionActivity
    name                CDATA #REQUIRED
    nameID              ID #IMPLIED
    businessTransaction CDATA #REQUIRED
    businessTransactionIDRef IDREF #IMPLIED
    fromAuthorizedRole  CDATA #REQUIRED
    fromAuthorizedRoleIDRef IDREF #IMPLIED
    toAuthorizedRole    CDATA #REQUIRED
    toAuthorizedRoleIDRef IDREF #IMPLIED
    isConcurrent (true | false) "true"
    isLegallyBinding (true | false) "true"
    timeToPerform CDATA #IMPLIED>
```

定義：

取引トランザクションアクティビティは、バイナリコラボレーションにおける取引トランザクションの使用を定義する。

取引トランザクションアクティビティは、所定の取引トランザクションを実行する業務アクティビティである。**isConcurrent** プロパティが **true** なら、取引トランザクションアクティビティを複数回にわたって実行できる。

親：

- BinaryCollaboration

属性：

属性名	定義	デフォルト値
name	バイナリコラボレーション内で、アクティビティに対して独自の名前を定義する。	必須入力
nameID	名前の XML ID バージョン。	オプション入力
businessTransaction	当該取引トランザクションアクティビティによって使用される取引トランザクションを名前で参照する参照	必須入力
businessTransactionIDRef	businessTransaction の XML IDREF バージョン。	オプション入力
fromAuthorizedRole	取引トランザクションアクティビティにおける開始役割の名前。これは、該当バイナリコラボレーション内の AuthorizedRoles の 1 つに一致しなければならず、このアクティビティにより実行される BusinessTransaction において要求者になる。	必須入力
fromAuthorizedRoleIDRef	fromAuthorizedRole の XML IDREF バージョン。	オプション入力
toAuthorizedRole	取引トランザクションアクティビティにおける応答役割の名前。これは、該当のバイナリコラボレーション内の AuthorizedRole の 1 つに一致しなければならず、このアクティビティにより実行される BusinessTransaction において応答者になる。	必須入力
toAuthorizedRoleIDRef	toAuthorizedRole の XML IDREF バージョン。	オプション入力
timeToPerform	応答役割が取引文書の事務受付を確認するのにかかる時間	オプション入力
isLegallyBinding	このアクティビティによって実行された取引トランザクションが、拘束力を持つようトレーディングパーティによって意図されているかどうかを定義する。デフォルト値は True。	true 有効値： {true, false}
isConcurrent	BusinessTransactionActivity が	true

	<p>並行実行可能であれば、関連した BusinessTransaction の1つ以上のインスタンスを、この BusinessTransactionActivity の実行の一部として同時に開くことができる。</p>	<p>有効値： {true, false}</p>
--	---	-------------------------------

h. Collaboration Activity

要素名： CollaborationActivity

DTD 宣言：

```

<!ELEMENT CollaborationActivity (Documentation*)>
<!ATTLIST CollaborationActivity
    name                CDATA #REQUIRED
    nameID              ID    #IMPLIED
    fromAuthorizedRole  CDATA #REQUIRED
    fromAuthorizedRoleIDRef CDATA #IMPLIED
    toAuthorizedRole    CDATA #REQUIRED
    toAuthorizedRoleIDRef CDATA #IMPLIED
    binaryCollaboration CDATA #REQUIRED>
    binaryCollaborationIDRef CDATA #IMPLIED>

```

定義：

コラボレーションアクティビティとは、別のバイナリコラボレーションの中でバイナリコラボレーションを実行するアクティビティである。

親：

- BinaryCollaboration

属性：

属性名	定義	デフォルト値
name	バイナリコラボレーション内で、アクティビティの名前を独自に定義する。	必須入力
nameID	名前の XML ID バージョン。	オプション入力
fromAuthorizedRole	「コラボレーションアクティビティ」における開始役割の名前 これは、親バイナリコラボレーションにおける AuthorizedRoles の 1 つに一致しなくてはならず、このアクティビティによって実行される BinaryCollaboration の開始者となる。	必須入力
FromAuthorizedRoleID Ref	fromAuthorizedRole の XML IDREF バージョン。	オプション入力
toAuthorizedRole	コラボレーションアクティビティの応答役割の名前。これは、親バイナリコラボレーションにおける AuthorizedRoles の 1 つに一致しなくてはならず、このアクティビティによって実行される BinaryCollaboration の応答者となる。	必須入力
toAuthorizedRoleIDRef	toAuthorizedRole の XML IDREF バージョン。	オプション入力
binaryCollaboration	このコラボレーションアクティビティによって実行されるバイナリコラボレーションへの、名前を使っての参照。	必須入力
BinaryCollaborationID Ref	binaryCollaboration の XML IDREF バージョン。	オプション入力

i. Documentation

要素名： Documentation

DTD 宣言：

```
<!ELEMENT Documentation (#PCDATA)>
<!ATTLIST Documentation
    uri CDATA #IMPLIED>
```

定義：

何らかの要素について、ユーザドキュメンテーションを定義する。そのコンテナの最初の要素にしなければならない。ドキュメンテーションはインライン PCDATA か、より完全なドキュメンテーションの場所を示す URI のいずれかにできる。

親：

- AuthorizedRole
- BinaryCollaboration
- BusinessPartnerRole
- BusinessTransaction
- BusinessTransactionActivity
- CollaborationActivity
- DocumentEnvelope
- BusinessDocument
- ProcvcssSpecification
- MultiPartyCollaboration
- Package
- Performs
- RequestingBusinessActivity
- RespondingBusinessActivity
- Transition

属性：

属性名	定義	デフォルト値
uri	外部ドキュメンテーションの場所を示す URI (統一リソース識別子) を定義する。XMLスキーマのバージョンにおいては、データ型は xsd:anyURI である。	デフォルト値なし。有効な URI が要求される。

j. DocumentEnvelope

要素名： DocumentEnvelope

内容モデル：

```
<!ELEMENT DocumentEnvelope (Documentation*,
                             Attachment*)>
<!ATTLIST DocumentEnvelope
  businessDocument CDATA #REQUIRED
  businessDocumentIDRef IDREF #IMPLIED
  isPositiveResponse (true | false) "false"
  isAuthenticated (true | false) "false"
  isConfidential (true | false) "false"
  isTamperProof (true | false) "false">
```

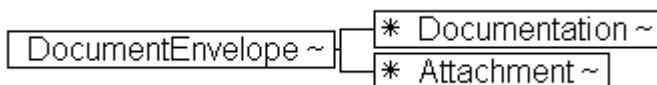
定義：

DocumentEnvelope とは、取引トランザクションにおける 2 役割間でビジネス情報を運ぶものである。1 つの DocumentEnvelope が要求役割から応答役割にかけて要求を運び、（応答がある場合は）もう 1 つの DocumentEnvelope が応答役割から要求役割にかけてその応答を運ぶ。

親：

- RequestingBusinessActivity
- RespondingBusinessActivity

階層モデル：



属性：

属性名	定義	デフォルト値
businessDocument	取引文書の名前。	必須入力
businessDocument IDRef	businessDocument の XML IDREF バージョン。	オプション入力
isPositiveResponse	TRUE または FALSE の値が使用される。TRUE の場合、この DocumentEnvelope は、要求に肯定的に回答する意図があると認識される。 DocumentEnvelope のこのパラメータに使用される値は、DocumentEnvelope の送信者が持つ、関連するトランザクションに対する意図を示す。しかしこの値は受信者に対する拘束力を持たず、またガード表現を使つてのトランザクションの成功または失敗の操作を上書きすることもしない。場合によっては、エンベロープ内の BusinessDocument に回答指令シグナルを送る XPath 表現を使用することもできる。 IsPositiveResponse は回答に対してのみ適用され、要求においては無視される。	オプション入力
isAuthenticated	文書エンティティにデジタル証明が付随し、署名者の身元を証	false 有効値：

	明する。 (詳細は、「文書セキュリティ」を参照のこと。)	{true, false}
isConfidential	許可されない者は情報を閲覧できなくするため、情報エンティティを暗号化する。 (詳細は、「文書セキュリティ」を参照のこと。)	false 有効値： {true, false}
isTamperProof	暗号化されたメッセージダイジェストが情報エンティティに付随する。これを利用すれば、メッセージに対する不正変更の有無をチェックできる。それには、文書エンティティにデジタル署名(送信者のデジタル証明と暗号化されたメッセージダイジェスト)を付随させる必要がある。 (詳細は、「文書セキュリティ」を参照のこと。)	false 有効値： {true, false}

I. DocumentSubstitution

要素名： DocumentSubstitution

DTD 宣言：

```
<!ELEMENT BusinessDocument (Documentation*) >
<!ATTLIST DocumentSubstitution
    originalBusinessDocument CDATA #IMPLIED
    originalBusinessDocumentID IDREF #IMPLIED
    substituteBusinessDocument CDATA #IMPLIED
    substituteBusinessDocumentId IDREF #IMPLIED
```

定義：

DocumentSubstitution を用いて、既存のプロセス仕様における文書を置き換えることになる文書を指定することができる。

親：

- SubstitutionSet

属性：

属性名	定義	デフォルト値
originalBusinessDoc	置換セットの範囲内の取引文書名。	必須入力

ument		
originalBusinessDocumentID	取引文書の ID。	オプション
substituteBusinessDocument	現行の文書を置き換えることになる文書。	必須入力
substituteBusinessDocumentID	置き換える文書の ID。	オプション

m. Failure

要素名： Failure

DTD 宣言：

```

<!ELEMENT Failure (ConditionExpression?,
Documentation*) >
<!ATTLIST Failure
    fromBusinessState CDATA #REQUIRED
    fromBusinessStateIDRef IDREF #IMPLIED
    conditionGuard (Success | BusinessFailure |
    TechnicalFailure | AnyFailure) #IMPLIED

```

定義：

アクティビティからの遷移として、バイナリコラボレーションの不良終結を定義する。

親：

- BinaryCollaboration

属性：

属性名	定義	デフォルト値
fromBusinessState	BusinessTransaction または BinaryCollaboration の不良終結への遷移を示すアクティビティの名前。	必須入力
fromBusinessStateIDRef	fromBusinessState の XML IDREF バージョン。	オプション
conditionGuard	この遷移をガードする条件。	オプション 有効値： {Success, BusinessFailure, TechnicalFailure, AnyFailure}

n. Fork

要素名： Fork

DTD 宣言：

```
<!ELEMENT Fork (Documentation*) >
<!ATTLIST Fork
    name CDATA #REQUIRED
    nameID ID #IMPLIED >
```

定義：

Fork は、1 つ以上の受信遷移および複数の送信遷移を含んだ状態である。送信遷移によって指されるすべてのアクティビティは、並行実行されることになっている。

親：

- BinaryCollaboration

属性：

属性名	定義	デフォルト値
name	Fork 状態の名前を定義する。	必須入力
nameID	名前の XML ID バージョン。	オプション

o. Include

要素名： Include

DTD 宣言：

```
<!ELEMENT Include (Documentation*) >
<!ATTLIST Include
    name CDATA #REQUIRED
    version CDATA #REQUIRED
    uuid CDATA #REQUIRED
    uri CDATA #REQUIRED >
```

定義：

別のプロセス仕様文書を編入し、その仕様を現行の仕様に併合する。同じ名前範囲の中にある同じ名前を持つ要素はどれも、まったく同じ仕様を持たなければならないが、パッケージは別の内容を持つ場合がある。

文書は、名前範囲に基づいて併合される。編入されるパッケージの中での名前は、基礎となる文書の中での名前と見分けがつかなくなる。

親：

- ProcessSpecification

階層モデル：

```
* Include ~ * Documentation ~
```

属性：

属性名	定義	デフォルト値
name	モデル要素の名前を定義する。この名前はモデル要素のコンテキストの中で一意でなければならない。この名前は、モデル内の他の場所から要素を参照するときに使われる。	必須入力
uri	統一リソース標識。XML スキーマにおけるデータ型は <code>xsd:anyURI</code> 。	必須
uuid	統一リソース標識。	必須
version	編入される仕様のバージョン	必須

o. Initiating Role

XML 要素名：InitiatingRole

DTD 宣言：

```
<!ELEMENT InitiatingRole (Documentation*)>
<!ATTLIST InitiatingRole
    name CDATA #REQUIRED
    nameID ID #IMPLIED>
```

定義：

Initiating Role は、例えば購買者が発注書の要求を送信することを認可されているというように、最初の要求を送信することを認可された役割で、バイナリコラボレーションを開始します。

親：

BinaryCollaboration

属性：

属性名	定義	デフォルト値
Name	開始役割の名前を定義します。	必須入力
nameID	名前の XML ID バージョン。	オプション

q. Join

要素名：Join

DTD 宣言：

```
<!ELEMENT Join (Documentation*) >
<!ATTLIST Join
    name CDATA #REQUIRED
    nameID ID #IMPLIED
    waitForAll (true | false) "true">
```

定義：

あるアクティビティが、1つ以上のアクティビティの完了を待っているビジネス状態である。前に分岐されたアクティビティを再び結合する時点を定義する。

親：

- BinaryCollaboration

属性：

属性名	定義	デフォルト値
name	Join 状態の名前を定義する。	必須入力
nameID	名前の XML ID バージョン。	オプション
waitForAll	Join 状態が、すべての受信遷移が完了するのを待つべきかを示す Boolean 値。TRUE の場合、すべての受信遷移を待ち、False の場合、最初の受信遷移から処理を開始する。	true 有効値： {true, false}

r. MultiParty Collaboration

要素名： MultiPartyCollaboration

DTD 宣言：

```
<!ELEMENT MultiPartyCollaboration (Documentation*,  
                                     BusinessPartnerRole+) >  
<!ATTLIST MultiPartyCollaboration  
           name      CDATA      #REQUIRED  
           nameID   ID         #IMPLIED   >
```

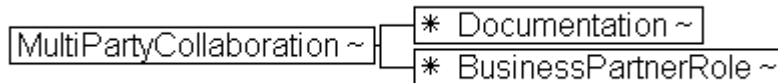
定義：

「複数当事者コラボレーション」は「バイナリコラボレーション」の合成である。
「複数当事者コラボレーション」を構成する数々の「取引当事者役割」は、バイナリコラボレーションの中で互いに役割を実行する。

親：

- ProcessSpecification
- Package

階層モデル：



属性：

属性名	定義	デフォルト値
name	MultiPartyCollaboration の名前を定義する。	必須入力
nameID	名前の XML ID バージョン。	オプション入力

s. Package

要素名： Package

DTD 宣言：

```
<!ELEMENT Package (Documentation*, (Package |
    BinaryCollaboration |
    MultiPartyCollaboration |
    BusinessTransaction)* ) >
<!ATTLIST Package
    name CDATA #REQUIRED
    nameID ID #IMPLIED >
```

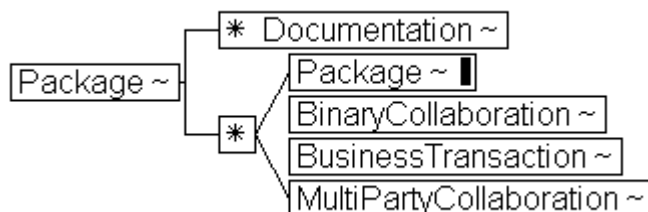
定義：

再利用可能要素を含む階層状の名前範囲を定義する。

親：

- ProcessSpecification
- package

階層モデル：



属性：

属性名	定義	デフォルト値
name	モデル要素の名前を定義する。この名前はモデル要素のコンテキストの中で一意でなければならない。この名前は、モデル内の他の場所から要素を参照するときに使われる。	必須入力
nameID	名前の XML ID バージョン。	オプション

t. Performs

要素名： Performs

DTD 宣言：

```
<!ELEMENT Performs (Documentation*) >
<!ATTLIST Performs
    initiatingRole CDATA #IMPLIED
    initiatingRoleIDRef IDREF #IMPLIED
    respondingRole CDATA #IMPLIED
    respondingRoleIDRef IDREF #IMPLIED >
```

定義：

Performs は、BusinessPartnerRole とそれが実行する Role との関係を明示的に表現するものである。これは、複数当事者コラボレーション内での「認定役割」の使用を指定する。認定された役割は、initiatingRole あるいは respondingRole

として示されなければならない。1つの認定された役割が、該当の Performs 要素において指定されなければならない。

親：

- BusinessPartnerRole

属性：

属性名	定義	デフォルト値
initiatingRole	Business PartnerRole によって実行される InitiatingRole であり、BinaryCollaboration の名前で修飾されている。	オプション入力
initiatingRoleIDRef	InitiatingRole の XML IDREF パージョン。	オプション入力
respondingRole	Business PartnerRole によって実行される RespondingRole であり、BinaryCollaboration の名前で修飾されている。	オプション入力
respondingRoleIDRef	RespondingRole の XML IDREF パージョン。	オプション入力

e. ProcessSpecification

要素名：ProcessSpecification

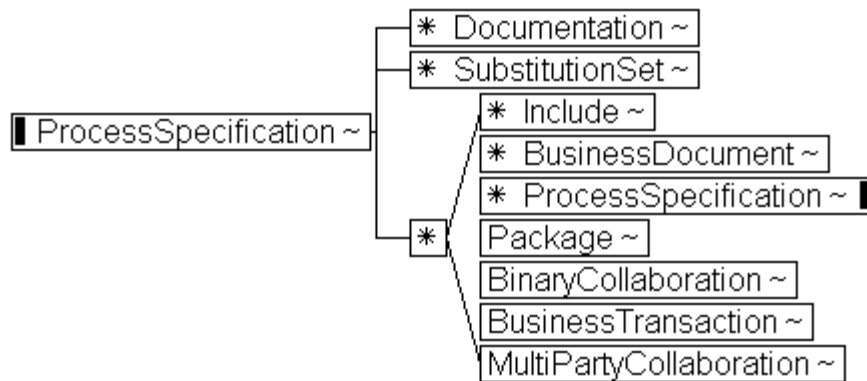
DTD 宣言：

```
<!ELEMENT ProcessSpecification (Documentation*,
SubstitutionSet*, (Include | BusinessDocument |
ProcessSpecification | Package | BinaryCollaboration |
BusinessTransaction | MultiPartyCollaboration)*)>
<!ATTLIST ProcessSpecification
    name          ID          #REQUIRED
    version       CDATA      #REQUIRED
    uuid          CDATA      #REQUIRED >
```

定義：

グローバルレベルで一意的な個性を持つプロセス仕様文書のルート要素。

階層モデル：



属性：

属性名	定義	デフォルト値
name	モデル要素の名前を定義する。この名前はモデル要素のコンテキストの中で一意でなければならない。この名前は、モデル内の他の場所から要素を参照するときに使われる。XML IDとして定義されている。	必須
uuid	普遍的固有識別子。	必須
version	仕様のバージョン。	必須

V. Requesting BusinessActivity

要素名： RequestingBusinessActivity

DTD 宣言：

```

<!ELEMENT RequestingBusinessActivity (Documentation*,
DocumentFlow) >
<!ATTLIST RequestingBusinessActivity
name CDATA #IMPLIED
namID ID #IMPLIED
isAuthorizationRequired (true | false) "false"
isIntelligibleCheckRequired (true | false) "false"
isNonRepudiationReceiptRequired (true | false) "false"
isNonRepudiationRequired (true | false) "false"
timeToAcknowledgeAcceptance CDATA #IMPLIED
timeToAcknowledgeReceipt CDATA #IMPLIED>
  
```

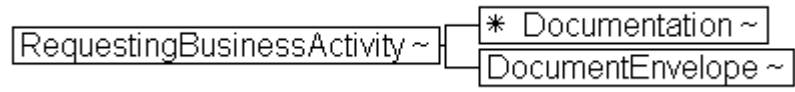
定義：

RequestingBusinessActivity は、取引トランザクション内で要求役割によって実行される業務アクティビティであり、要求を伝達する文書エンベロープを指定する。

親：

- BusinessTransaction

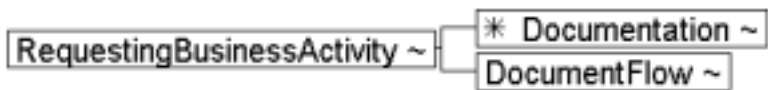
階層モデル：



属性：

属性名	定義	デフォルト値
name	RequestingBusinessTransaction の名前を定義する。	オプション入力
nameID	名前の XML ID バージョン。	オプション入力
isAuthorizationRequired	<p>受信当事者は、認定発起者の一覧と照合することで、発起者の身元を確認しなければならない。</p> <p>このパラメータは、送信者側で指定される。</p> <p>(詳細は、「認定セキュリティ」の節を参照のこと)</p>	<p>false</p> <p>有効値： {true, false}</p>
isIntelligibleCheckRequired	<p>受信者当事者は、受領確認を送信する前に、要求文書が不正に変更(読み取り不可能、または理解不可能であること)されていないかどうかを確認しなければならない。</p> <p>パラメータは、送信者側で指定される。</p> <p>(詳細は、「コア取引トランザクション意味情報」の節を参照のこと)</p>	<p>false</p> <p>有効値： {true, false}</p>
isNonRepudiationReceiptRequired	<p>受信当事者には署名付受取りの返送を要求し、原送信者には受取りコピーの保存を要求する</p> <p>(詳細は、「否認不可」の節を参照のこと)。</p>	<p>false</p> <p>有効値： {true, false}</p>
isNonRepudiationRequired	<p>トランザクションに使用された文書のコピーを送信する前に、送信当事者がそのコピーを保存することを要求する</p> <p>(詳細は、「否認不可」の節を参照のこと)。</p>	<p>false</p> <p>有効値： {true, false}</p>
timeToAcknowledgeAcceptance	<p>応答役割が、取引文書の事務受付けを形式的に確認するのにかかる時間。このパラメータは、要求者側で指</p>	<p>デフォルト値なし。</p>

	定される(詳細は、「コア取引トランザクション意味情報」を参照のこと)。	
timeToAcknowledgeReceipt	応答役割が、取引文書の受領を確認するのにかかる時間。このパラメータは、送信者側で指定される(詳細は、「コア取引トランザクション意味情報」の節を参照のこと)。	デフォルト値なし。



w. Responding BusinessActivity

要素名： RespondingBusinessActivity

DTD 宣言：

```
<!ELEMENT RespondingBusinessActivity (Documentation*,
                                       DocumentEnvelope) >
<!ATTLIST RespondingBusinessActivity
  name          CDATA          #IMPLIED
  nameID        ID            #IMPLIED
  isAuthorizationRequired (true | false) "false"
  isIntelligibleCheckRequired (true | false) "false"
  isNonRepudiationReceiptRequired (true | false) "false"
  isNonRepudiationRequired (true | false) "false"
  timeToAcknowledgeReceipt CDATA          #IMPLIED>
```

定義：

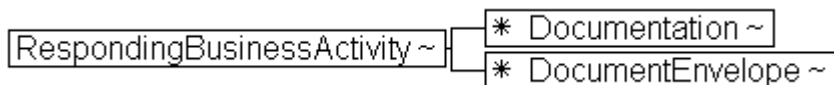
RespondingBusinessActivity は、取引トランザクション内で応答役割によって実行される業務アクティビティであり、応答を伝送する文書エンベロープを指定する。

複数の応答「文書フロー」を定義する場合があるが、実際のトランザクションインスタンスではその内のただ1つが発生する。

親：

- BusinessTransaction

階層モデル：



属性：

属性名	定義	デフォルト値
name	RespondingBusinessTrans アクティビティの要素名を定義し、この属性を用いて、モデル内の他の点から要素を参照することができる。	オプション入力
nameID	名前の XML ID バージョン。	オプション入力
isAuthorizationRequired	受信当事者は、認定された発起人の一覧と照合することで発起人の身元を確認し	false 有効値：

	なければならない。 パラメータは、送信者側で指定される。 (詳細は、「認定セキュリティ」の節を参照のこと)	{true, false}
isIntelligibleCheckRequired	受信当事者は受領確認を送信する前に、要求文書が不正に変更されていないか(読み取り不可能、または理解不可能であるなど)を確認しなければならない。このパラメータは、送信者側で指定される。 (詳細は、「コア取引トランザクション意味情報」の節を参照のこと)	false 有効値： {true, false}
isNonRepudiationReceiptRequired	受信当事者には署名付受取りの返送を要求し、原送信者には受取りコピーの保存を要求する。このパラメータは、送信者側で指定される。 (詳細は、「否認不可」の節を参照のこと)	false 有効値： {true, false}
isNonRepudiationRequired	トランザクションに使用された文書のコピーを送信する前に、送信当事者がそのコピーを保存することを要求する(詳細は、「否認不可」の節を参照のこと)。	false 有効値： {true, false}
timeToAcknowledgeReceipt	応答役割が、取引文書の受領を確認するのにかかる時間。このパラメータは、送信者側で指定される。 (詳細は、「コア取引トランザクション意味情報」の節を参照のこと)	デフォルト値なし。

x. Responding Role

XML 要素名： RespondingRole

DTD 宣言：

```
<!ELEMENT RespondingRole (Documentation*)>
```

```
<!ATTLIST RespondingRole
    name CDATA #REQUIRED
    nameID ID #IMPLIED>
```

定義：

Responding Role は、例えば販売者が発注書の受取りを送信することが許可されているなどの、最初の応答を送信することが認定されている役割である。この役割は、バイナリコラボレーション内の応答者である。

親：

BinaryCollaboration

属性：

属性名	定義	デフォルト値
Name	応答役割の名前を定義する。	必須入力
nameID	名前の XML ID バージョン。	オプション

y. Start

要素名：Start

DTD 宣言：

```
<!ELEMENT Start (Documentation*) >
<!ATTLIST Start
    toBusinessState CDATA #REQUIRED
    toBusinessStateIDRef IDREF #IMPLIED >
```

定義：

「バイナリコラボレーション」の開始状態。「バイナリコラボレーション」には少なくとも1つの開始アクティビティが存在する必要がある。まったく定義されない場合は、すべてのアクティビティが進入可能点として見なされる。

親：

- BinaryCollaboration

属性：

属性名	定義	デフォルト値
toBusinessState	BinaryCollaboration の開始可能点であるアクティビティの名前。	必須入力
toBusinessStateIDRef	toBusinessState の XML IDREF バージョン。	オプション

z. SubstitutionSet

要素名： SubstitutionSet

DTD 宣言：

```
<!ELEMENT SubstitutionSet (DocumentSubstitution |
AttributeSubstitution, Documentation)*>
<!ATTLIST SubstitutionSet
    name CDATA #IMPLIED
    nameID ID #IMPLIED
    applyToScope CDATA #IMPLIED
>
```

定義：

Substitution Set は、1 つ以上の AttributeSubstitution 要素または DocumentSubstitution 要素、あるいはその両方を格納するコンテナである。SubstitutionSet 全体で、既存のプロセス仕様の属性値および文書を置き換えることになる文書および属性値が指定される。

親：

- ProcessSpecification

属性：

属性名	定義	デフォルト値
name	置換セットの名前。	オプション入力
nameID	置換セットの ID。	オプション入力
applyToScope	置き換えられる属性または文書へのパスを指定する。	オプション入力

aa. Success

要素名： Success

DTD 宣言：

```
<!ELEMENT Success (ConditionExpression?, Documentation*)
>
<!ATTLIST Success
    fromBusinessState CDATA #REQUIRED
    conditionGuard (Success | BusinessFailure |
        TechnicalFailure | AnyFailure) #IMPLIED
```

定義：

アクティビティからの遷移として、バイナリコラボレーションの良好終了を定義する。

親：

- BinaryCollaboration

属性：

属性名	定義	デフォルト値
fromBusinessState	BusinessTransaction または BinaryCollaboration の良好終了への遷移を示すアクティビティの名前。	必須入力
conditionGuard	この遷移をガードする条件。	オプション 有効値： {Success, BusinessFailure, TechnicalFailure, AnyFailure}

aa. Transition

要素名： Transition

DTD 宣言：

```
<!ELEMENT Transition (ConditionExpression?, Documentation*)
>
<!ATTLIST Transition
  onInitiation (true | false) "false"
  fromBusinessState CDATA #IMPLIED
  fromBusinessStateIDRef IDREF #IMPLIED
  toBusinessState CDATA #IMPLIED
  toBusinessStateIDRef IDREF #IMPLIED
  conditionGuard (Success | BusinessFailure |
    TechnicalFailure | AnyFailure) #IMPLIED
```

定義：

遷移は、バイナリコラボレーションにおける 2 つのビジネス状態間の遷移である。
振り付けは、ビジネス状態間の遷移として表現される。

親：

- BinaryCollaboration
- BusinessPartnerRole

属性：

属性名	定義	デフォルト値
onInitiation	これが入り子の BusinessTransactionActivity であること、そして関連した要求の受領後に、該当トランザクションに戻る前に 2 つ目のアクティビティを実行することで、元の要求者に応答を送信するように指定することが指定される。	false 有効値： {true, false}
fromBusinessState	遷移前の状態の名前	デフォルト値なし。

fromBusinessStateIDRef	fromBusinessState の XML IDREF バージョン。	オプション
toBusinessState	遷移後の状態の名前。	デフォルト値なし。
toBusinessStateIDRef	toBusinessState の XML IDREF バージョン。	オプション
conditionGuard	前のトランザクションの状態への参照。Success、BusinessFailure、TechnicalFailure、および AnyFailure の固定値。	オプション 有効値： {Success, BusinessFailure, TechnicalFailure, AnyFailure}

8.2 XML から UML への相互参照

以下の表は、DTD における XML 要素名から、それに相対する、UML 仕様スキーマにおけるクラスを参照する表である。

XML 要素	UML クラス
Attachment	Attachment
AuthorizedRole	AuthorizedRole
Binary Collaboration	Binary Collaboration
BusinessPartner Role	BusinessPartner Role
Business Transaction Activity	Business Transaction Activity
Business Transaction	Business Transaction
Responding BusinessActivity	Responding BusinessActivity
Requesting BusinessActivity	Requesting BusinessActivity

Collaboration Activity	Collaboration Activity
DocumentEnvelope	DocumentEnvelope
Documentation	None (Should be added)
ebXml Process Specification	(From Package model: ebXml Process Specification)
Failure	Failure
Include	(From Package model: Include)
MultiParty Collaboration	MultiParty Collaboration
Package	(From Package model: Package)
Performs	Performs
Schema	Schema
Fork	Fork
Start	Start
Success	Success
Join	Join
Transition	Transition

以下に記す UML 仕様のクラスは抽象であり、DTD には同等の要素がない。DTD には、その具象サブタイプだけがある。

- BusinessState
- TerminalState
- BusinessActivity
- BusinessAction
- DocumentSecurity

8.3 有範囲名前参照

ebXML プロセス仕様は再利用を促進する構造になっている。
ProcessSpecification には、参照を通じて他の ProcessSpecification を編入できる。

さらに、ProcessSpecification の内容は再帰パッケージの構造に構成できる。
ProcessSpecification はパッケージの容器（コンテナ）だからパッケージを内包できる。パッケージそのものもパッケージコンテナだから、別のパッケージを内包できる。

以下に述べるように、パッケージはネームスペースとして機能する。

パッケージは最終的に、何がしかのレベルで PackageContent を持つ。
PackageContent のタイプは BusinessTransaction、BinaryCollaboration、MultiPartyCollaboration である。

PackageContent は常に、1 パッケージの中で一意な名前を持つ。親 PackageContent の中では、低位の要素が一意な名前を持つ。

各 PackageContent タイプは、この ebXMLProcessSpecification によって参照される「取引文書」定義のコア構成要素「論理モデル」のビルトインコンテキストプロバイダである。

ebXMLProcessSpecification の中での命名には以下の事柄があてはまる：

仕様要素は、属性を介して名前によって他の仕様要素を参照する。設計パターンは、要素が名前属性を持っており、名前付きの要素を参照するほかの要素が、参照された要素の lowerCamelCase バージョン(例：InitiatingRole は属性名を持つ一方で、InitiatingRole を参照する Performs は InitiatingRole という属性を持つ)として定義された属性を介して名前を参照する。2つのタイプの属性は命名と参照の目的で、それぞれ XML ID/IDREF ベース、そしてテキスト形式で表示される。名付けられた要素それぞれには、必須とされる名前属性およびオプションの nameID 属性が含まれる。参照用要素には、参照される要素のために lowerCamelCase 属性および lowerCamelCaseIDRef 属性が含まれている。XML ID/IDREF 機能では、すべての ID が文書内で一意であることと、そしてすべての IDREF が、定義された ID 値を指すことが要求される。テキスト形式の属性にはこのような機能はないため、名前が重複する場合もある。参照された要素を、参照属性内のテキスト形式の属性を使用して明確に識別するには、XPath 構文を利用することを強く推奨する。しかし、これは DTD またはスキーマにおいて実施されていない。

これらの2つの解決法を提示した目的は、プロセス仕様文書を XML で直接作成できるようにするためと、機械で読み取り可能な nameID そして参照を自動的に割り当てることができる開発ツールを将来サポートできるようにするためである。これらの2つの方法を同時に使用することも可能であるが、この場合 ID および IDREF バージョンが明確な参照を提供し、テキスト形式のバージョンが意味のある名前を提供する。名付けられた要素と参照の例を以下に示す。

```
<Package name="ebXMLOrdering">
```

```
<BinaryCollaboration name="OrderCollaboration" nameID="b112">
  <InitiatingRole name="buyer" nameID="r224"/>
  <RespondingRole name="seller" nameID="r225"/>
</BinaryCollaboration>
</Package>
```

<!-- the XPath approach -->

<Performs

```
initiatingRole=../Package[@name="ebXMLOrdering"]/BinaryCollaboration[@name="OrderCollaboration"]/InitiatingRole[@name="buyer"]/>
```

<!-- Combination approach -->

```
<Performs initiatingRole="buyer" initiatingRoleIDRef="r224"/>
```

上述のように、フルパス仕様を使用する必要はなく、1 つの参照を解決できるのであれば、他の XPath 表現を使用することも可能である。例えば、購買者が文書にとって一意である場合、XPath は以下のように記述されてもよい。

```
<Performs initiatingRole=../InitiatingRole[@name="buyer"]/>
```

Relative paths are also allowed for example:

```
<BusinessTransactionActivity fromAuthorizedRole='../InitiatingRole[@name="buyer"]' ... />
```

8.4 置換セット

一般的な『ebXML Business Process Specifications』は、特定の文書形式、構造、タイミングパラメータなどの技術あるいはビジネス詳細に密着に結びついていない。置換セットにより、一般的ビジネスプロセスを特定の用法に特殊化する機能がサポートされる。例えば、発注プロセスは非常に一般的であるが、このプロセスを特殊化するためには、一般用法を越えるための特定文書の機能が必要になる場合がある。

置換セットは、特定のプロセス仕様のコピー内に配置され、より明確な文書定義参照および属性値になるように置き換えを行う。

置換セットは、1 つ以上の AttributeSubstitution 要素または DocumentSubstitution 要素、あるいはその両方を格納するコンテナである。SubstitutionSet 全体は、既存のプロセス仕様内の文書および属性値を置き換える文書または属性値を指定する。

8.5 上記 DTD に照らし合わせたサンプル XML 文書

付録 A を参照のこと。

9 ビジネスシグナルの構造

ebXML メッセージ取扱サービス仕様シグナルの構造は、基本的な関連付け要件を満たすために使用される一意のメッセージ識別子およびダイジェストを含む、取引サービス状態の関連付け基盤の働きをする。ここで提供されるビジネスシグナルの搬送内容構造は、オプションかつ規範的であり、ビジネス意味情報および法的意味情報をビジネスシグナルに与えることになっている。シグナルは、取引トランザクションごとに変化しないため、1 回定義されるだけで済む。シグナルの定義は、『ビジネスプロセス仕様スキーマ』およびメメッセージ取扱サービス仕様と連動して定義される。以下で、receiptAcknowledgment(RosettaNet および Edifecs の協力で、RosettaNet のウェブサイトから抜粋された)、acceptanceAcknowledgement、および例外のビジネスシグナル搬送内容の DTD を示す。

9.1.1 ReceiptAcknowledgment の DTD

```
<!--  
  
    RosettaNet XML Message Schema.  
    http://www.rosettanet.org  
    RosettaNet XML Message Schema.  
    Receipt Acknowledgement  
    Version 1.1  
-->  
  
<!ENTITY % common-attributes "id CDATA #IMPLIED">  
  
<!ELEMENT ReceiptAcknowledgement (  
    fromRole ,  
    NonRepudiationInformation? ,  
    receivedDocumentDateTime ,  
    receivedDocumentIdentifier ,  
    thisMessageDateTime ,  
    thisMessageIdentifier ,  
    toRole ) >  
  
<!ELEMENT fromRole  
    ( PartnerRoleDescription ) >  
  
<!ELEMENT PartnerRoleDescription (  
    ContactInformation? ,  
    GlobalPartnerRoleClassificationCode ,  
    PartnerDescription ) >  
  
<!ELEMENT ContactInformation (  
    contactName ,
```

```

        EmailAddress ,
        telephoneNumber ) >

<!ELEMENT contactName
  ( FreeFormText ) >

<!ELEMENT FreeFormText
  ( #PCDATA ) >
<!ATTLIST FreeFormText
  xml:lang CDATA #IMPLIED >

<!ELEMENT EmailAddress
  ( #PCDATA ) >

<!ELEMENT telephoneNumber
  ( CommunicationsNumber ) >

<!ELEMENT CommunicationsNumber
  ( #PCDATA ) >

<!ELEMENT GlobalPartnerRoleClassificationCode
  ( #PCDATA ) >

<!ELEMENT PartnerDescription (
  BusinessDescription ,
  GlobalPartnerClassificationCode ) >

<!ELEMENT BusinessDescription (
  GlobalBusinessIdentifier ,
  GlobalSupplyChainCode ) >

<!ELEMENT GlobalBusinessIdentifier
  ( #PCDATA ) >

<!ELEMENT GlobalSupplyChainCode
  ( #PCDATA ) >

<!ELEMENT GlobalPartnerClassificationCode
  ( #PCDATA ) >

<!ELEMENT NonRepudiationInformation (
  GlobalDigestAlgorithmCode ,
  OriginalMessageDigest ) >

<!ELEMENT GlobalDigestAlgorithmCode
  ( #PCDATA ) >

<!ELEMENT OriginalMessageDigest
  ( #PCDATA ) >

<!ELEMENT receivedDocumentDateTime
  ( DateTimeStamp ) >

<!ELEMENT DateTimeStamp
  ( #PCDATA ) >

<!ELEMENT receivedDocumentIdentifier

```

```

        ( ProprietaryDocumentIdentifier ) >
<!ELEMENT ProprietaryDocumentIdentifier
  ( #PCDATA ) >
<!ELEMENT thisMessageDateTime
  ( DateTimeStamp ) >
<!ELEMENT thisMessageIdentifier
  ( ProprietaryMessageIdentifier ) >
<!ELEMENT ProprietaryMessageIdentifier
  ( #PCDATA ) >
<!ELEMENT toRole
  ( PartnerRoleDescription ) >

```

9.1.2 AcceptanceAcknowledgement Ⓞ DTD

```

<!--
  RosettaNet XML Message Schema.
  http://www.rosettanet.org
  RosettaNet XML Message Schema.
  Acceptance Acknowledgement Exception
  Version 1.1
-->
<!ENTITY % common-attributes "id CDATA #IMPLIED">
<!ELEMENT AcceptanceAcknowledgementException (
  fromRole ,
  reason ,
  theMessageDatetime ,
  theOffendingDocumentDateTime ,
  theOffendingDocumentIdentifier ,
  thisMessageIdentifier ,
  toRole ) >
<!ELEMENT fromRole
  ( PartnerRoleDescription ) >
<!ELEMENT PartnerRoleDescription (
  ContactInformation? ,
  GlobalPartnerRoleClassificationCode ,
  PartnerDescription ) >
<!ELEMENT ContactInformation (
  contactName ,
  EmailAddress ,
  telephoneNumber ) >
<!ELEMENT contactName
  ( FreeFormText ) >
<!ELEMENT FreeFormText
  ( #PCDATA ) >
<!ATTLIST FreeFormText

```

```

        xml:lang CDATA #IMPLIED >

<!ELEMENT EmailAddress
    ( #PCDATA ) >

<!ELEMENT telephoneNumber
    ( CommunicationsNumber ) >

<!ELEMENT CommunicationsNumber
    ( #PCDATA ) >

<!ELEMENT GlobalPartnerRoleClassificationCode
    ( #PCDATA ) >

<!ELEMENT PartnerDescription (
    BusinessDescription ,
    GlobalPartnerClassificationCode ) >

<!ELEMENT BusinessDescription (
    GlobalBusinessIdentifier ,
    GlobalSupplyChainCode ) >

<!ELEMENT GlobalBusinessIdentifier
    ( #PCDATA ) >

<!ELEMENT GlobalSupplyChainCode
    ( #PCDATA ) >

<!ELEMENT GlobalPartnerClassificationCode
    ( #PCDATA ) >

<!ELEMENT reason
    ( FreeFormText ) >

<!ELEMENT theMessageDatetime
    ( DateTimeStamp ) >

<!ELEMENT DateTimeStamp
    ( #PCDATA ) >

<!ELEMENT theOffendingDocumentDateTime
    ( DateTimeStamp ) >

<!ELEMENT theOffendingDocumentIdentifier
    ( ProprietaryDocumentIdentifier ) >

<!ELEMENT ProprietaryDocumentIdentifier
    ( #PCDATA ) >

<!ELEMENT thisMessageIdentifier
    ( ProprietaryMessageIdentifier ) >

<!ELEMENT ProprietaryMessageIdentifier
    ( #PCDATA ) >

<!ELEMENT toRole
    ( PartnerRoleDescription ) >

```

9.1.3 例外シグナルの DTD

```
<!--
  RosettaNet XML Message Schema.
  http://www.rosettnet.org
  RosettaNet XML Message Schema.
  Exception
  Version 1.1
-->

<!ENTITY % common-attributes "id CDATA #IMPLIED">

<!ELEMENT Exception (
  fromRole? ,
  reason ,
  theMessageDatetime ,
  theOffendingDocumentDateTime? ,
  theOffendingDocumentIdentifier? ,
  thisMessageIdentifier ,
  toRole? ) >

<!ELEMENT fromRole
  ( PartnerRoleDescription ) >

<!ELEMENT PartnerRoleDescription (
  ContactInformation? ,
  GlobalPartnerRoleClassificationCode? ,
  PartnerDescription? ) >

<!ELEMENT ContactInformation (
  contactName? ,
  EmailAddress? ,
  telephoneNumber? ) >

<!ELEMENT contactName
  ( FreeFormText ) >

<!ELEMENT FreeFormText
  ( #PCDATA ) >
<!ATTLIST FreeFormText
  xml:lang CDATA #IMPLIED >

<!ELEMENT EmailAddress
  ( #PCDATA ) >

<!ELEMENT telephoneNumber
  ( CommunicationsNumber ) >

<!ELEMENT CommunicationsNumber
  ( #PCDATA ) >

<!ELEMENT GlobalPartnerRoleClassificationCode
  ( #PCDATA ) >
```

```

<!ELEMENT PartnerDescription (
    BusinessDescription? ,
    GlobalPartnerClassificationCode? ) >

<!ELEMENT BusinessDescription (
    GlobalBusinessIdentifier? ,
    GlobalSupplyChainCode? ) >

<!ELEMENT GlobalBusinessIdentifier
    ( #PCDATA ) >

<!ELEMENT GlobalSupplyChainCode
    ( #PCDATA ) >

<!ELEMENT GlobalPartnerClassificationCode
    ( #PCDATA ) >

<!ELEMENT reason
    ( FreeFormText ) >

<!ELEMENT theMessageDatetime
    ( DateTimeStamp ) >

<!ELEMENT DateTimeStamp
    ( #PCDATA ) >

<!ELEMENT theOffendingDocumentDateTime
    ( DateTimeStamp ) >

<!ELEMENT theOffendingDocumentIdentifier
    ( ProprietaryDocumentIdentifier ) >

<!ELEMENT ProprietaryDocumentIdentifier
    ( #PCDATA ) >

<!ELEMENT thisMessageIdentifier
    ( ProprietaryMessageIdentifier ) >

<!ELEMENT ProprietaryMessageIdentifier
    ( #PCDATA ) >

<!ELEMENT toRole
    ( PartnerRoleDescription ) >

```

10 生成規則

本節では、『ビジネスプロセス仕様スキーマ』のUMLバージョンからXMLバージョンへのマッピングを定義している一連の生成規則を解説する。

これらの生成規則の主な目的は、『ビジネスプロセス仕様スキーマ』のUMLクラス図バージョンから1回行われる『ビジネスプロセス仕様スキーマ』のDTDバージョンの生成を統制することである。

『ビジネスプロセス仕様スキーマ』のクラス図バージョンは、『ebXML Business Process Specifications』を直接作成するためのものではない。しかしビジネスプロセス仕様がこのクラス図の1つのインスタンスとして実際に作成された場合、この生成規則はまた、このようなインスタンスをXML仕様文書をDTD用に変換するために必要な規定の定義を提供する。具体クラス、抽象クラス、統合関連、特殊化関連、非方向関連の生成規則が定義されている。

1. クラスはXML要素として表現される。
2. クラス属性は、XML属性として表現される。注：発生要件(必須、またはオプション)および属性のデフォルト値はモデル化されていない。
3. 特殊化クラス(別のクラスから継承されたクラス)は、ベースクラスからのすべての属性および統合関連と含むXML要素として表現される。反復される属性は、1つの発生として標準化される。
4. 抽象クラスはXML DTDで表現されない。抽象クラスは継承され、集合体として表される。抽象クラスを統合するクラスは、原則的に特殊化クラスの異なった種類それぞれ1つを統合する。
5. 統合関連は、統合クラスをXML子要素として、適切な濃度で表現する。
6. 非方向関連は、この関連が指しているクラスと同名の元のクラスにおいて属性を定義する。このタイプの属性は“参照属性”と呼ばれ、この属性が指すクラスの名前を含む。参照クラスには、“名前”属性が含まれなくてはならない。
7. 固定化した<<Enumeration>>を含んだクラスと同名のクラスを持ったクラス属性データ型は、XML属性の一覧として表現される。一覧クラスには、明示的な関連は含まれない。
8. 対応するクラス定義を持たないクラス属性のデータ型(例：Time、URI、Boolean)は、DTDの文字列として表現される。XMLスキーマバージョンでは、これらのデータ型は以下のようにマップされる。
Time - xsd:duration
URI - xsd:anyURI
Boolean - xsd:boolean
9. 各クラスにはオプションとして、特殊化インスタンスの注釈としての“Documentation*”要素が与えられている。これはモデル化されていない。

付録 A : XML Business Process Specification のサンプル

```
<!-- edited by Kurt Kanaskie (Lucent Technologies) -->
<!DOCTYPE ProcessSpecification SYSTEM "ebBPSS-v1.01.dtd">
<ProcessSpecification name="Simple" version="1.1" uuid="[1234-5678-901234]">
  <!-- Business Documents -->
    <BusinessDocument name="Catalog Request"/>
    <BusinessDocument name="Catalog"/>
    <BusinessDocument name="Purchase Order"/>
    <BusinessDocument name="PO Acknowledgement"/>
    <BusinessDocument name="Credit Request"/>
    <BusinessDocument name="Credit Confirm"/>
    <BusinessDocument name="ASN"/>
    <BusinessDocument name="CreditAdvice"/>
    <BusinessDocument name="DebitAdvice"/>
    <BusinessDocument name="Invoice"/>
    <BusinessDocument name="Payment"/>
    <BusinessDocument name="Inventory Report Request"/>
    <BusinessDocument name="Inventory Report"/>
    <BusinessDocument name="Inventory Report"/>
  <Package name="Ordering">
    <!-- First the overall MultiParty Collaboration -->
    <MultiPartyCollaboration name="DropShip">
      <BusinessPartnerRole name="Customer">
        <Performs initiatingRole="requestor"/>
        <Performs initiatingRole="buyer"/>
        <Transition fromBusinessState="Catalog Request"
toBusinessState="Create Order"/>
      </BusinessPartnerRole>
      <BusinessPartnerRole name="Retailer">
        <Performs respondingRole="provider"/>
        <Performs respondingRole="seller"/>
        <Performs initiatingRole="Creditor"/>
        <Performs initiatingRole="buyer"/>
        <Performs initiatingRole="Payee"/>
        <Performs respondingRole="Payor"/>
        <Performs initiatingRole="requestor"/>
        <Transition fromBusinessState="Create Order"
toBusinessState="Check Credit"/>
        <Transition fromBusinessState="Check Credit"
toBusinessState="Create Order"/>
      </BusinessPartnerRole>
      <BusinessPartnerRole name="DropShip Vendor">
        <Performs respondingRole="seller"/>
        <Performs initiatingRole="payee"/>
        <Performs respondingRole="provider"/>
      </BusinessPartnerRole>
      <BusinessPartnerRole name="Credit Authority">
        <Performs respondingRole="credit service"/>
        <Performs respondingRole="payor"/>
      </BusinessPartnerRole>
    </MultiPartyCollaboration>
    <!-- Now the Binary Collaborations -->
```

```

        <BinaryCollaboration name="Request Catalog">
            <InitiatingRole name="requestor"/>
            <RespondingRole name="provider"/>
            <BusinessTransactionActivity name="Catalog Request"
businessTransaction="Catalog Request" fromAuthorizedRole="requestor"
toAuthorizedRole="provider"/>
        </BinaryCollaboration>
        <BinaryCollaboration name="Firm Order" timeToPerform="P2D">
            <Documentation>timeToPerform = Period: 2 days from
start of transaction</Documentation>
            <InitiatingRole name="buyer"/>
            <RespondingRole name="seller"/>
            <BusinessTransactionActivity name="Create Order"
businessTransaction="Create Order" fromAuthorizedRole="buyer"
toAuthorizedRole="seller"/>
        </BinaryCollaboration>
        <BinaryCollaboration name="Product Fulfillment"
timeToPerform="P5D">
            <Documentation>timeToPerform = Period: 5 days from
start of transaction</Documentation>
            <InitiatingRole name="buyer"/>
            <RespondingRole name="seller"/>
            <BusinessTransactionActivity name="Create Order"
businessTransaction="Create Order" fromAuthorizedRole="buyer"
toAuthorizedRole="seller"/>
            <BusinessTransactionActivity name="Notify shipment"
businessTransaction="Notify of advance shipment"
fromAuthorizedRole="buyer" toAuthorizedRole="seller"/>
            <Start toBusinessState="Create Order"/>
            <Transition fromBusinessState="Create Order"
toBusinessState="Notify shipment"/>
            <Success fromBusinessState="Notify shipment"
conditionGuard="Success"/>
            <Failure fromBusinessState="Notify shipment"
conditionGuard="BusinessFailure"/>
        </BinaryCollaboration>
        <BinaryCollaboration name="Inventory Status">
            <InitiatingRole name="requestor"/>
            <RespondingRole name="provider"/>
            <BusinessTransactionActivity name="Inventory Report
Request" businessTransaction="Inventory Report Request"
fromAuthorizedRole="requestor" toAuthorizedRole="provider"/>
            <BusinessTransactionActivity name="Inventory Report"
businessTransaction="Inventory Report" fromAuthorizedRole="provider"
toAuthorizedRole="requestor"/>
        </BinaryCollaboration>
        <BinaryCollaboration name="Credit Inquiry">
            <InitiatingRole name="creditor"/>
            <RespondingRole name="credit service"/>
            <BusinessTransactionActivity name="Check Credit"
businessTransaction="Check Credit" fromAuthorizedRole="creditor"
toAuthorizedRole="credit service"/>
        </BinaryCollaboration>
        <BinaryCollaboration name="Credit Payment">
            <InitiatingRole name="payee"/>
            <RespondingRole name="payor"/>

```

```

        <BusinessTransactionActivity name="Process Credit
Payment" businessTransaction="Process Credit Payment"
fromAuthorizedRole="payee" toAuthorizedRole="payor"/>
    </BinaryCollaboration>
    <!-- A compound BinaryCollaboration for illustration
purposes-->
    <BinaryCollaboration name="Credit Charge">
        <InitiatingRole name="charger"/>
        <RespondingRole name="credit service"/>
        <CollaborationActivity name="Credit Inquiry"
binaryCollaboration="Credit Inquiry" fromAuthorizedRole="charger"
toAuthorizedRole="credit service"/>
        <CollaborationActivity name="Credit Payment"
binaryCollaboration="Credit Payment" fromAuthorizedRole="charger"
toAuthorizedRole="credit service"/>
        <Transition fromBusinessState="Credit Inquiry"
toBusinessState="Credit Payment"/>
    </BinaryCollaboration>
    <BinaryCollaboration name="Fulfillment Payment">
        <InitiatingRole name="payee"/>
        <RespondingRole name="payor"/>
        <BusinessTransactionActivity name="Process Payment"
businessTransaction="Process Payment" fromAuthorizedRole="payee"
toAuthorizedRole="payor"/>
    </BinaryCollaboration>
    <!-- Here are all the Business Transactions needed -->
    <BusinessTransaction name="Catalog Request">
        <RequestingBusinessActivity name="">
            <DocumentEnvelope isPositiveResponse="true"
businessDocument="Catalog Request"/>
        </RequestingBusinessActivity>
        <RespondingBusinessActivity name="">
            <DocumentEnvelope isPositiveResponse="true"
businessDocument="Catalog"/>
        </RespondingBusinessActivity>
    </BusinessTransaction>
    <BusinessTransaction name="Create Order">
        <RequestingBusinessActivity name=""
isNonRepudiationRequired="true" timeToAcknowledgeReceipt="P2D"
timeToAcknowledgeAcceptance="P3D">
            <DocumentEnvelope isPositiveResponse="true"
businessDocument="Purchase Order"/>
        </RequestingBusinessActivity>
        <RespondingBusinessActivity name=""
isNonRepudiationRequired="true" timeToAcknowledgeReceipt="P5D">
            <DocumentEnvelope isPositiveResponse="true"
businessDocument="PO Acknowledgement"/>
        </RespondingBusinessActivity>
    </BusinessTransaction>
    <BusinessTransaction name="Check Credit ">
        <RequestingBusinessActivity name="">
            <DocumentEnvelope isPositiveResponse="true"
businessDocument="Credit Request"/>
        </RequestingBusinessActivity>
        <RespondingBusinessActivity name="">
            <DocumentEnvelope isPositiveResponse="true"
businessDocument="Credit Confirm"/>

```

```

        </RespondingBusinessActivity>
    </BusinessTransaction>
    <BusinessTransaction name="Notify of advance shipment">
        <RequestingBusinessActivity name="">
            <DocumentEnvelope isPositiveResponse="true"
businessDocument="ASN"/>
        </RequestingBusinessActivity>
        <RespondingBusinessActivity name=""
timeToAcknowledgeReceipt="P2D"/>
    </BusinessTransaction>
    <BusinessTransaction name="Process Credit Payment">
        <RequestingBusinessActivity name="">
            <DocumentEnvelope isPositiveResponse="true"
businessDocument="CreditAdvice"/>
        </RequestingBusinessActivity>
        <RespondingBusinessActivity name="">
            <DocumentEnvelope isPositiveResponse="true"
businessDocument="DebitAdvice"/>
        </RespondingBusinessActivity>
    </BusinessTransaction>
    <BusinessTransaction name="Process Payment">
        <RequestingBusinessActivity name="">
            <DocumentEnvelope isPositiveResponse="true"
businessDocument="Invoice"/>
        </RequestingBusinessActivity>
        <RespondingBusinessActivity name="">
            <DocumentEnvelope isPositiveResponse="true"
businessDocument="Payment"/>
        </RespondingBusinessActivity>
    </BusinessTransaction>
    <BusinessTransaction name="Request Inventory Report">
        <RequestingBusinessActivity name="">
            <DocumentEnvelope isPositiveResponse="true"
businessDocument="Inventory Report Request"/>
        </RequestingBusinessActivity>
        <RespondingBusinessActivity name="">
            <DocumentEnvelope isPositiveResponse="true"
businessDocument="Inventory Report"/>
        </RespondingBusinessActivity>
    </BusinessTransaction>
    <BusinessTransaction name="Inventory Report">
        <RequestingBusinessActivity name="">
            <DocumentEnvelope isPositiveResponse="true"
businessDocument="Inventory Report"/>
        </RequestingBusinessActivity>
        <RespondingBusinessActivity name=""/>
    </BusinessTransaction>
</Package>
</ProcessSpecification>

```

付録 B : ビジネスプロセス仕様スキーマの DTD

```
<!-- ===== -->
<!-- Editor: Kurt Kanaskie (Lucent Technologies) -->
<!-- Version: Version 1.01 -->
<!-- Updated: 2001-05-24 -->
<!-- -->
<!-- Public Identifier: -->
<!-- "-//ebXML//DTD ビジネスプロセス仕様スキーマ v1.01//EN" -->
<!-- -->
<!-- Purpose: -->
<!-- The ebXML Specification DTD provides a standard -->
<!-- framework by which business systems may be -->
<!-- configured to support execution of business -->
<!-- transactions. It is based upon prior UN/CEFACT -->
<!-- work, specifically the metamodel behind the -->
<!-- UN/CEFACT Unified Modeling Methodology (UMM) defined -->
<!-- in the N090R9.1 specification. -->
-->
<!-- -->
<!-- The 仕様スキーマ supports the specification -->
<!-- of Business Transactions and the choreography of -->
<!-- Business Transactions into Business Collaborations. -->
<!-- -->
<!-- Notes: -->
<!-- time periods are represented using ISO 8601 format -->
<!-- (e.g. P2D for 2 Days, P2H30M for 2 Hours 30 Minutes -->
<!-- -->
<!-- Naming and reference is based on convention that an -->
<!-- Element with a name attribute (e.g. AuthorizedRole) -->
<!-- is refernced by an attribute in another element with -->
<!-- the name in lowerCamelCase (e.g. authorizedRole). -->
<!-- -->
<!-- fromBusinessState and toBusinessState refer to the -->
<!-- the names of a BusinessTransactionActivity, -->
<!-- CollaborationActivity, Fork, and Join, all are targets for -->
<!-- from/to in Transition. This deviates from the normal -->
<!-- convention of lowerCamelCase attribute name -->
<!-- BusinessState is used as a generic term for: -->
<!-- Fork, Join, Success, Failure -->
<!-- -->
<!-- Constraints: -->
<!-- - attributes specificationLocation, pattern, specification -->
<!-- uri, are of type xsd:anyURI -->
<!-- - attributes timeTo* are of type xsd:duration -->
<!-- -->
<!-- ===== -->
<!ELEMENT ProcessSpecification (Documentation*, SubstitutionSet*,
(Include | BusinessDocument | ProcessSpecification | Package |
BinaryCollaboration | BusinessTransaction | MultiPartyCollaboration)*)>
<!ATTLIST ProcessSpecification
name ID #REQUIRED
uuid CDATA #REQUIRED
version CDATA #REQUIRED
```

```

>
<!ELEMENT Documentation (#PCDATA)>
<!ATTLIST Documentation
    uri CDATA #IMPLIED
>
<!ELEMENT Include (Documentation*)>
<!ATTLIST Include
    name CDATA #REQUIRED
    uuid CDATA #REQUIRED
    uri CDATA #REQUIRED
    version CDATA #REQUIRED
>
<!ELEMENT BusinessDocument (ConditionExpression?, Documentation*)>
<!ATTLIST BusinessDocument
    name CDATA #REQUIRED
    nameID ID #IMPLIED
    specificationLocation CDATA #IMPLIED
    specificationElement CDATA #IMPLIED
>
<!ELEMENT ConditionExpression (Documentation*)>
<!ATTLIST ConditionExpression
    expressionLanguage CDATA #IMPLIED
    expression CDATA #IMPLIED
>
<!ELEMENT SubstitutionSet (DocumentSubstitution | AttributeSubstitution
| Documentation)*>
<!ATTLIST SubstitutionSet
    name CDATA #IMPLIED
    nameId IDREF #IMPLIED
    applyToScope CDATA #IMPLIED
>
<!ELEMENT DocumentSubstitution (Documentation*)>
<!ATTLIST DocumentSubstitution
    originalBusinessDocument CDATA #IMPLIED
    originalBusinessDocumentID IDREF #IMPLIED
    substituteBusinessDocument CDATA #IMPLIED
    substituteBusinessDocumentId IDREF #IMPLIED
>
<!ELEMENT AttributeSubstitution (Documentation*)>
<!ATTLIST AttributeSubstitution
    attributeName CDATA #IMPLIED
    value CDATA #IMPLIED
>
<!ELEMENT Package (Documentation*, (Package | BinaryCollaboration |
BusinessTransaction | MultiPartyCollaboration)*)>
<!ATTLIST Package
    name CDATA #REQUIRED
    nameID ID #IMPLIED
>
<!ELEMENT BinaryCollaboration (Documentation*, InitiatingRole,
RespondingRole, (Documentation | Start | Transition | Success | Failure
| BusinessTransactionActivity | CollaborationActivity | Fork | Join)*)>
<!ATTLIST BinaryCollaboration
    name CDATA #REQUIRED
    nameID ID #IMPLIED
    pattern CDATA #IMPLIED
    beginsWhen CDATA #IMPLIED

```

```

        endsWhen CDATA #IMPLIED
        preCondition CDATA #IMPLIED
        postCondition CDATA #IMPLIED
        timeToPerform CDATA #IMPLIED
    >
<!ELEMENT MultiPartyCollaboration (Documentation*,
BusinessPartnerRole*)>
<!ATTLIST MultiPartyCollaboration
    name CDATA #REQUIRED
    nameID ID #IMPLIED
>
<!ELEMENT InitiatingRole (Documentation*)>
<!ATTLIST InitiatingRole
    name CDATA #REQUIRED
    nameID ID #IMPLIED
>
<!ELEMENT RespondingRole (Documentation*)>
<!ATTLIST RespondingRole
    name CDATA #REQUIRED
    nameID ID #IMPLIED
>
<!-- A BusinessState is one of Start, Success, Failure, Fork, Join,
BusinessTransactionActivity or CollaborationActivity -->
<!-- fromBusinessState and toBusinessState are fully qualified using
XPath -->
<!ELEMENT Transition (ConditionExpression?, Documentation*)>
<!ATTLIST Transition
    onInitiation (true | false) "false"
    fromBusinessState CDATA #IMPLIED
    fromBusinessStateIDRef IDREF #IMPLIED
    toBusinessState CDATA #IMPLIED
    toBusinessStateIDRef IDREF #IMPLIED
    conditionGuard (Success | BusinessFailure | TechnicalFailure |
AnyFailure) #IMPLIED
>
<!-- Start is a special type of Transition in that it only has a
destination -->
<!ELEMENT Start (Documentation*)>
<!ATTLIST Start
    toBusinessState CDATA #REQUIRED
    toBusinessStateIDRef IDREF #IMPLIED
>
<!-- Success is a special type of Transition in that it only has a
origination -->
<!ELEMENT Success (ConditionExpression?, Documentation*)>
<!ATTLIST Success
    fromBusinessState CDATA #REQUIRED
    fromBusinessStateIDRef IDREF #IMPLIED
    conditionGuard (Success | BusinessFailure | TechnicalFailure |
AnyFailure) #IMPLIED
>
<!-- Failure is a special type of Transition in that it only has a
origination -->
<!ELEMENT Failure (ConditionExpression?, Documentation*)>
<!ATTLIST Failure
    fromBusinessState CDATA #REQUIRED
    fromBusinessStateIDRef IDREF #IMPLIED

```



```

        conditionGuard (Success | BusinessFailure | TechnicalFailure |
AnyFailure) #IMPLIED
    >
<!-- Fork is a special type of BusinessState that can be transitioned to
-->
<!ELEMENT Fork (Documentation*)>
<!ATTLIST Fork
    name CDATA #REQUIRED
    nameID ID #IMPLIED
>
<!-- Join is a special type of BusinessState that can be transitioned to
-->
<!ELEMENT Join (Documentation*)>
<!ATTLIST Join
    name CDATA #REQUIRED
    nameID ID #IMPLIED
    waitForAll (true | false) "true"
>
<!-- fromAuthorizedRole and toAuthorizedRole are fully qualified using
XPath -->
<!-- BusinessTransactionActivity is a BusinessState that can be
transitioned to -->
<!ELEMENT BusinessTransactionActivity (Documentation*)>
<!ATTLIST BusinessTransactionActivity
    name CDATA #REQUIRED
    nameID ID #IMPLIED
    businessTransaction CDATA #REQUIRED
    businessTransactionIDRef IDREF #IMPLIED
    fromAuthorizedRole CDATA #REQUIRED
    fromAuthorizedRoleIDRef IDREF #IMPLIED
    toAuthorizedRole CDATA #REQUIRED
    toAuthorizedRoleIDRef IDREF #IMPLIED
    isConcurrent (true | false) "true"
    isLegallyBinding (true | false) "true"
    timeToPerform CDATA #IMPLIED
>
<!-- fromAuthorizedRole and toAuthorizedRole are fully qualified using
XPath -->
<!-- CollaborationActivity is a BusinessState that can be transitioned
to -->
<!ELEMENT CollaborationActivity (Documentation*)>
<!ATTLIST CollaborationActivity
    name CDATA #REQUIRED
    nameID ID #IMPLIED
    fromAuthorizedRole CDATA #REQUIRED
    fromAuthorizedRoleIDRef IDREF #IMPLIED
    toAuthorizedRole CDATA #REQUIRED
    toAuthorizedRoleIDRef IDREF #IMPLIED
    binaryCollaboration CDATA #REQUIRED
    binaryCollaborationIDRef IDREF #IMPLIED
>
<!ELEMENT BusinessTransaction (Documentation*,
RequestingBusinessActivity, RespondingBusinessActivity)>
<!ATTLIST BusinessTransaction
    name CDATA #REQUIRED
    nameID ID #IMPLIED
    pattern CDATA #IMPLIED

```

```

    beginsWhen CDATA #IMPLIED
    endsWhen CDATA #IMPLIED
    isGuaranteedDeliveryRequired (true | false) "false"
    preCondition CDATA #IMPLIED
    postCondition CDATA #IMPLIED
  >
<!ELEMENT RequestingBusinessActivity (Documentation*, DocumentEnvelope)>
<!ATTLIST RequestingBusinessActivity
  name CDATA #IMPLIED
  nameID ID #IMPLIED
  isAuthorizationRequired (true | false) "false"
  isIntelligibleCheckRequired (true | false) "false"
  isNonRepudiationReceiptRequired (true | false) "false"
  isNonRepudiationRequired (true | false) "false"
  timeToAcknowledgeAcceptance CDATA #IMPLIED
  timeToAcknowledgeReceipt CDATA #IMPLIED
>
<!ELEMENT RespondingBusinessActivity (Documentation*,
DocumentEnvelope*)>
<!ATTLIST RespondingBusinessActivity
  name CDATA #IMPLIED
  nameID ID #IMPLIED
  isAuthorizationRequired (true | false) "false"
  isIntelligibleCheckRequired (true | false) "false"
  isNonRepudiationReceiptRequired (true | false) "false"
  isNonRepudiationRequired (true | false) "false"
  timeToAcknowledgeReceipt CDATA #IMPLIED
>
<!ELEMENT DocumentEnvelope (Documentation*, Attachment*)>
<!ATTLIST DocumentEnvelope
  businessDocument CDATA #REQUIRED
  businessDocumentIDRef IDREF #IMPLIED
  isPositiveResponse (true | false) "false"
  isAuthenticated (true | false) "false"
  isConfidential (true | false) "false"
  isTamperProof (true | false) "false"
>
<!ELEMENT Attachment (Documentation*)>
<!ATTLIST Attachment
  name CDATA #REQUIRED
  nameID ID #IMPLIED
  businessDocument CDATA #IMPLIED
  businessDocumentIDRef IDREF #IMPLIED
  mimeType CDATA #REQUIRED
  specification CDATA #IMPLIED
  version CDATA #IMPLIED
  isAuthenticated (true | false) "false"
  isConfidential (true | false) "false"
  isTamperProof (true | false) "false"
>
<!ELEMENT BusinessPartnerRole (Documentation*, Performs*, Transition*)>
<!ATTLIST BusinessPartnerRole
  name CDATA #REQUIRED
  nameID ID #IMPLIED
>
<!-- initiatingRole/respondingRole is fully qualified using XPath -->
<!ELEMENT Performs (Documentation*)>

```

```
<!ATTLIST Performs
  initiatingRole CDATA #IMPLIED
  inititiatingRoleIDRef IDREF #IMPLIED
  respondingRole CDATA #IMPLIED
  respondingRoleIDRef IDREF #IMPLIED
>
```

付録 C : ビジネスプロセス仕様スキーマの XML

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- edited by Kurt Kanaskie (Lucent Technologies) -->
<!-- Updated 2001-05-24
    Differences from DTD version:
    <xsd:attribute name="pattern" type="xsd:anyURI"/>
    <xsd:attribute name="uri" type="xsd:anyURI" use="required"/>
    <xsd:attribute name="location" type="xsd:anyURI"/>
    <xsd:attribute name="logicalModel" type="xsd:anyURI"/>
    <xsd:attribute name="specification" type="xsd:anyURI"/>
    <xsd:attribute name="timeToPerform" type="xsd:duration"/>
    <xsd:attribute name="timeToPerform" type="xsd:duration"/>
    <xsd:attribute name="timeToAcknowledgeAcceptance"
type="xsd:duration"/>
    <xsd:attribute name="timeToAcknowledgeReceipt"
type="xsd:duration"/>
    <xsd:attribute name="timeToAcknowledgeAcceptance"
type="xsd:duration"/>
    <xsd:attribute name="timeToAcknowledgeReceipt"
type="xsd:duration"/>
    <xsd:attribute name="isAuthenticated" type="xsd:boolean"
value="false"/>
    <xsd:attribute name="isConfidential" type="xsd:boolean"
value="false"/>
    <xsd:attribute name="isTamperProof" type="xsd:boolean"
value="false"/>
    <xsd:attribute name="isGuaranteedDeliveryRequired"
type="xsd:boolean" value="false"/>
    <xsd:attribute name="isConcurrent" type="xsd:boolean"
value="true"/>
    <xsd:attribute name="isLegallyBinding" type="xsd:boolean"
value="true"/>
    <xsd:attribute name="isAuthenticated" type="xsd:boolean"
value="false"/>
    <xsd:attribute name="isConfidential" type="xsd:boolean"
value="false"/>
    <xsd:attribute name="isTamperProof" type="xsd:boolean"
value="false"/>
    <xsd:attribute name="waitForAll" type="xsd:boolean" value="true"/>
    <xsd:attribute name="isAuthorizationRequired" type="xsd:boolean"
value="false"/>
    <xsd:attribute name="isIntelligibleCheckRequired"
type="xsd:boolean" value="false"/>
    <xsd:attribute name="isNonRepudiationReceiptRequired"
type="xsd:boolean" value="false"/>
    <xsd:attribute name="isNonRepudiationRequired" type="xsd:boolean"
value="false"/>
    <xsd:attribute name="isAuthorizationRequired" type="xsd:boolean"
value="false"/>
    <xsd:attribute name="isIntelligibleCheckRequired"
type="xsd:boolean" value="false"/>
    <xsd:attribute name="isNonRepudiationReceiptRequired"
type="xsd:boolean" value="false"/>
```

```

        <xsd:attribute name="isNonRepudiationRequired" type="xsd:boolean"
value="false"/>
        <xsd:attribute name="onInitiation" type="xsd:boolean"
value="false"/>
        <xsd:attribute name="isPositiveResponse" type="xsd:boolean"/>
-->
<xsd:schema targetNamespace="http://www.ebxml.org/BusinessProcess"
xmlns:xsd="http://www.w3.org/2000/10/XMLSchema"
xmlns="http://www.ebxml.org/BusinessProcess"
elementFormDefault="qualified">
    <xsd:element name="Attachment">
        <xsd:complexType>
            <xsd:sequence>
                <xsd:element ref="Documentation" minOccurs="0"
maxOccurs="unbounded"/>
            </xsd:sequence>
            <xsd:attribute name="name" type="xsd:string"
use="required"/>
            <xsd:attribute name="nameID" type="xsd:ID"/>
            <xsd:attribute name="businessDocument"
type="xsd:string"/>
            <xsd:attribute name="businessDocumentIDRef"
type="xsd:IDREF"/>
            <xsd:attribute name="specification"
type="xsd:anyURI"/>
            <xsd:attribute name="mimeType" type="xsd:string"
use="required"/>
            <xsd:attribute name="version" type="xsd:string"/>
            <xsd:attribute name="isAuthenticated"
type="xsd:boolean" value="false"/>
            <xsd:attribute name="isConfidential"
type="xsd:boolean" value="false"/>
            <xsd:attribute name="isTamperProof" type="xsd:boolean"
value="false"/>
        </xsd:complexType>
    </xsd:element>
    <xsd:element name="InitiatingRole">
        <xsd:complexType>
            <xsd:sequence>
                <xsd:element ref="Documentation" minOccurs="0"
maxOccurs="unbounded"/>
            </xsd:sequence>
            <xsd:attribute name="name" type="xsd:string"
use="required"/>
            <xsd:attribute name="nameID" type="xsd:ID"/>
        </xsd:complexType>
    </xsd:element>
    <xsd:element name="RespondingRole">
        <xsd:complexType>
            <xsd:sequence>
                <xsd:element ref="Documentation" minOccurs="0"
maxOccurs="unbounded"/>
            </xsd:sequence>
            <xsd:attribute name="name" type="xsd:string"
use="required"/>
            <xsd:attribute name="nameID" type="xsd:ID"/>
        </xsd:complexType>

```

```

</xsd:element>
<xsd:element name="BinaryCollaboration">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="Documentation" minOccurs="0"
maxOccurs="unbounded"/>
      <xsd:element ref="InitiatingRole"/>
      <xsd:element ref="RespondingRole"/>
      <xsd:choice minOccurs="0" maxOccurs="unbounded">
        <xsd:element ref="Documentation"/>
        <xsd:element ref="Start"/>
        <xsd:element ref="Transition"/>
        <xsd:element ref="Success"/>
        <xsd:element ref="Failure"/>
        <xsd:element
ref="BusinessTransactionActivity"/>
        <xsd:element ref="CollaborationActivity"/>
        <xsd:element ref="Fork"/>
        <xsd:element ref="Join"/>
      </xsd:choice>
    </xsd:sequence>
    <xsd:attribute name="name" type="xsd:string"
use="required"/>
    <xsd:attribute name="nameID" type="xsd:ID"/>
    <xsd:attribute name="pattern" type="xsd:anyURI"/>
    <xsd:attribute name="beginsWhen" type="xsd:string"/>
    <xsd:attribute name="endsWhen" type="xsd:string"/>
    <xsd:attribute name="preCondition" type="xsd:string"/>
    <xsd:attribute name="postCondition"
type="xsd:string"/>
    <xsd:attribute name="timeToPerform"
type="xsd:duration"/>
  </xsd:complexType>
</xsd:element>
<xsd:element name="BusinessDocument">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="Documentation" minOccurs="0"
maxOccurs="unbounded"/>
      <xsd:element ref="ConditionExpression"
minOccurs="0" maxOccurs="1"/>
    </xsd:sequence>
    <xsd:attribute name="name" type="xsd:string"
use="required"/>
    <xsd:attribute name="nameID" type="xsd:ID"/>
    <xsd:attribute name="specificationLocation"
type="xsd:string"/>
    <xsd:attribute name="specificationElement"
type="xsd:string"/>
  </xsd:complexType>
</xsd:element>
<xsd:element name="SubstitutionSet">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="DocumentSubstitution"
minOccurs="0" maxOccurs="unbounded"/>

```

```

        <xsd:element ref="AttributeSubstitution"
minOccurs="0" maxOccurs="unbounded"/>
        <xsd:element ref="Documentation" minOccurs="0"
maxOccurs="unbounded"/>
    </xsd:sequence>
    <xsd:attribute name="name" type="xsd:string"/>
    <xsd:attribute name="nameId" type="xsd:ID"/>
    <xsd:attribute name="applyToScope" type="xsd:string"/>
</xsd:complexType>
</xsd:element>
<xsd:element name="DocumentSubstitution">
    <xsd:complexType>
        <xsd:sequence>
            <xsd:element ref="Documentation" minOccurs="0"
maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:attribute name="originalBusinessDocument"
type="xsd:string"/>
        <xsd:attribute name="originalBusinessDocumentID"
type="xsd:ID"/>
        <xsd:attribute name="substituteBusinessDocument"
type="xsd:string"/>
        <xsd:attribute name="substituteBusinessDocumentId"
type="xsd:ID"/>
    </xsd:complexType>
</xsd:element>
<xsd:element name="AttributeSubstitution">
    <xsd:complexType>
        <xsd:sequence>
            <xsd:element ref="Documentation" minOccurs="0"
maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:attribute name="attributeName"
type="xsd:string"/>
        <xsd:attribute name="value" type="xsd:string"/>
    </xsd:complexType>
</xsd:element>
<xsd:element name="ConditionExpression">
    <xsd:complexType>
        <xsd:sequence>
            <xsd:element ref="Documentation" minOccurs="0"
maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:attribute name="expressionLanguage"
type="xsd:string"/>
        <xsd:attribute name="expression" type="xsd:string"/>
    </xsd:complexType>
</xsd:element>
<xsd:element name="BusinessPartnerRole">
    <xsd:complexType>
        <xsd:sequence>
            <xsd:element ref="Documentation" minOccurs="0"
maxOccurs="unbounded"/>
            <xsd:element ref="Performs" minOccurs="0"
maxOccurs="unbounded"/>
            <xsd:element ref="Transition" minOccurs="0"
maxOccurs="unbounded"/>

```

```

        </xsd:sequence>
        <xsd:attribute name="name" type="xsd:string"
use="required"/>
        <xsd:attribute name="nameID" type="xsd:ID"/>
    </xsd:complexType>
</xsd:element>
<xsd:element name="BusinessTransaction">
    <xsd:complexType>
        <xsd:sequence>
            <xsd:element ref="Documentation" minOccurs="0"
maxOccurs="unbounded"/>
            <xsd:element ref="RequestingBusinessActivity"/>
            <xsd:element ref="RespondingBusinessActivity"/>
        </xsd:sequence>
        <xsd:attribute name="name" type="xsd:string"
use="required"/>
        <xsd:attribute name="nameID" type="xsd:ID"/>
        <xsd:attribute name="pattern" type="xsd:anyURI"/>
        <xsd:attribute name="beginsWhen" type="xsd:string"/>
        <xsd:attribute name="endsWhen" type="xsd:string"/>
        <xsd:attribute name="isGuaranteedDeliveryRequired"
type="xsd:boolean" value="false"/>
        <xsd:attribute name="preCondition" type="xsd:string"/>
        <xsd:attribute name="postCondition"
type="xsd:string"/>
    </xsd:complexType>
</xsd:element>
<xsd:element name="BusinessTransactionActivity">
    <xsd:complexType>
        <xsd:sequence>
            <xsd:element ref="Documentation" minOccurs="0"
maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:attribute name="name" type="xsd:string"
use="required"/>
        <xsd:attribute name="nameID" type="xsd:ID"/>
        <xsd:attribute name="businessTransaction"
type="xsd:string" use="required"/>
        <xsd:attribute name="businessTransactionIDRef"
type="xsd:IDREF"/>
        <xsd:attribute name="fromAuthorizedRole"
type="xsd:string" use="required"/>
        <xsd:attribute name="fromAuthorizedRoleIDRef"
type="xsd:IDREF"/>
        <xsd:attribute name="toAuthorizedRole"
type="xsd:string" use="required"/>
        <xsd:attribute name="toAuthorizedRoleIDRef"
type="xsd:IDREF"/>
        <xsd:attribute name="isConcurrent" type="xsd:boolean"
value="true"/>
        <xsd:attribute name="isLegallyBinding"
type="xsd:boolean" value="true"/>
        <xsd:attribute name="timeToPerform"
type="xsd:duration"/>
    </xsd:complexType>
</xsd:element>
<xsd:element name="CollaborationActivity">

```



```

        <xsd:complexType>
            <xsd:sequence>
                <xsd:element ref="Documentation" minOccurs="0"
maxOccurs="unbounded" />
            </xsd:sequence>
            <xsd:attribute name="nameID" type="xsd:ID" />
            <xsd:attribute name="name" type="xsd:string"
use="required" />
            <xsd:attribute name="fromAuthorizedRole"
type="xsd:string" use="required" />
            <xsd:attribute name="fromAuthorizedRoleIDRef"
type="xsd:IDREF" />
            <xsd:attribute name="toAuthorizedRole"
type="xsd:string" use="required" />
            <xsd:attribute name="toAuthorizedRoleIDRef"
type="xsd:IDREF" />
            <xsd:attribute name="binaryCollaboration"
type="xsd:string" use="required" />
            <xsd:attribute name="binaryCollaborationIDRef"
type="xsd:IDREF" />
        </xsd:complexType>
    </xsd:element>
    <xsd:element name="DocumentEnvelope">
        <xsd:complexType>
            <xsd:sequence>
                <xsd:element ref="Documentation" minOccurs="0"
maxOccurs="unbounded" />
                <xsd:element ref="Attachment" minOccurs="0"
maxOccurs="unbounded" />
            </xsd:sequence>
            <xsd:attribute name="businessDocument"
type="xsd:string" use="required" />
            <xsd:attribute name="businessDocumentIDRef"
type="xsd:IDREF" />
            <xsd:attribute name="isPositiveResponse"
type="xsd:boolean" />
            <xsd:attribute name="isAuthenticated"
type="xsd:boolean" value="false" />
            <xsd:attribute name="isConfidential"
type="xsd:boolean" value="false" />
            <xsd:attribute name="isTamperProof" type="xsd:boolean"
value="false" />
        </xsd:complexType>
    </xsd:element>
    <xsd:element name="Documentation">
        <xsd:complexType>
            <xsd:simpleContent>
                <xsd:restriction base="xsd:string">
                    <xsd:attribute name="uri"
type="xsd:anyURI" />
                </xsd:restriction>
            </xsd:simpleContent>
        </xsd:complexType>
    </xsd:element>
    <xsd:element name="Failure">
        <xsd:complexType>
            <xsd:sequence>

```

```

        <xsd:element ref="Documentation" minOccurs="0"
maxOccurs="unbounded" />
        <xsd:element ref="ConditionExpression"
minOccurs="0" maxOccurs="1" />
    </xsd:sequence>
    <xsd:attribute name="fromBusinessState"
type="xsd:string" use="required" />
    <xsd:attribute name="fromBusinessStateIDRef"
type="xsd:IDREF" />
    <xsd:attribute name="conditionGuard">
        <xsd:simpleType>
            <xsd:restriction base="xsd:NMTOKEN">
                <xsd:enumeration value="Success" />
                <xsd:enumeration
value="BusinessFailure" />
                <xsd:enumeration
value="TechnicalFailure" />
                <xsd:enumeration
value="AnyFailure" />
            </xsd:restriction>
        </xsd:simpleType>
    </xsd:attribute>
</xsd:complexType>
</xsd:element>
<xsd:element name="Fork">
    <xsd:complexType>
        <xsd:sequence>
            <xsd:element ref="Documentation" minOccurs="0"
maxOccurs="unbounded" />
        </xsd:sequence>
    </xsd:complexType>
    <xsd:attribute name="name" type="xsd:string"
use="required" />
    <xsd:attribute name="nameID" type="xsd:ID" />
</xsd:complexType>
</xsd:element>
<xsd:element name="Include">
    <xsd:complexType>
        <xsd:sequence>
            <xsd:element ref="Documentation" minOccurs="0"
maxOccurs="unbounded" />
        </xsd:sequence>
        <xsd:attribute name="name" type="xsd:string"
use="required" />
        <xsd:attribute name="uuid" type="xsd:string"
use="required" />
        <xsd:attribute name="uri" type="xsd:anyURI"
use="required" />
        <xsd:attribute name="version" type="xsd:string"
use="required" />
    </xsd:complexType>
</xsd:element>
<xsd:element name="Join">
    <xsd:complexType>
        <xsd:sequence>
            <xsd:element ref="Documentation" minOccurs="0"
maxOccurs="unbounded" />
        </xsd:sequence>
    </xsd:complexType>
</xsd:element>

```

```

        <xsd:attribute name="name" type="xsd:string"
use="required"/>
        <xsd:attribute name="nameID" type="xsd:ID"/>
        <xsd:attribute name="waitForAll" type="xsd:boolean"
value="true"/>
        </xsd:complexType>
    </xsd:element>
    <xsd:element name="MultiPartyCollaboration">
        <xsd:complexType>
            <xsd:sequence>
                <xsd:element ref="Documentation" minOccurs="0"
maxOccurs="unbounded"/>
                <xsd:element ref="BusinessPartnerRole"
minOccurs="0" maxOccurs="unbounded"/>
            </xsd:sequence>
            <xsd:attribute name="name" type="xsd:string"
use="required"/>
            <xsd:attribute name="nameID" type="xsd:ID"/>
        </xsd:complexType>
    </xsd:element>
    <xsd:element name="Package">
        <xsd:complexType>
            <xsd:sequence>
                <xsd:element ref="Documentation" minOccurs="0"
maxOccurs="unbounded"/>
                <xsd:choice minOccurs="0" maxOccurs="unbounded">
                    <xsd:element ref="Package"/>
                    <xsd:element ref="BinaryCollaboration"/>
                    <xsd:element ref="BusinessTransaction"/>
                </xsd:choice>
            </xsd:sequence>
            <xsd:attribute name="name" type="xsd:string"
use="required"/>
            <xsd:attribute name="nameID" type="xsd:ID"/>
        </xsd:complexType>
    </xsd:element>
    <xsd:element name="Performs">
        <xsd:complexType>
            <xsd:sequence>
                <xsd:element ref="Documentation" minOccurs="0"
maxOccurs="unbounded"/>
            </xsd:sequence>
            <xsd:attribute name="initiatingRole" type="xsd:string"
use="optional"/>
            <xsd:attribute name="initiatingRoleIDRef"
type="xsd:IDREF"/>
            <xsd:attribute name="respondingRole" type="xsd:string"
use="optional"/>
            <xsd:attribute name="respondingRoleIDRef"
type="xsd:IDREF"/>
        </xsd:complexType>
    </xsd:element>
    <xsd:element name="ProcessSpecification">
        <xsd:complexType>
            <xsd:sequence>

```

```

        <xsd:element ref="Documentation" minOccurs="0"
maxOccurs="unbounded" />
        <xsd:element ref="SubstitutionSet" minOccurs="0"
maxOccurs="unbounded" />
        <xsd:choice minOccurs="0" maxOccurs="unbounded">
            <xsd:element ref="Include" />
            <xsd:element ref="BusinessDocument" />
            <xsd:element ref="ProcessSpecification" />
            <xsd:element ref="Package" />
            <xsd:element ref="BinaryCollaboration" />
            <xsd:element ref="BusinessTransaction" />
            <xsd:element
ref="MultiPartyCollaboration" />
        </xsd:choice>
    </xsd:sequence>
    <xsd:attribute name="name" type="xsd:ID"
use="required" />
    <xsd:attribute name="uuid" type="xsd:string"
use="required" />
    <xsd:attribute name="version" type="xsd:string"
use="required" />
</xsd:complexType>
</xsd:element>
<xsd:element name="RequestingBusinessActivity">
    <xsd:complexType>
        <xsd:sequence>
            <xsd:element ref="Documentation" minOccurs="0"
maxOccurs="unbounded" />
            <xsd:element ref="DocumentEnvelope" />
        </xsd:sequence>
        <xsd:attribute name="name" type="xsd:string"
use="required" />
        <xsd:attribute name="nameID" type="xsd:ID" />
        <xsd:attribute name="isAuthorizationRequired"
type="xsd:boolean" value="false" />
        <xsd:attribute name="isIntelligibleCheckRequired"
type="xsd:boolean" value="false" />
        <xsd:attribute name="isNonRepudiationReceiptRequired"
type="xsd:boolean" value="false" />
        <xsd:attribute name="isNonRepudiationRequired"
type="xsd:boolean" value="false" />
        <xsd:attribute name="timeToAcknowledgeAcceptance"
type="xsd:duration" />
        <xsd:attribute name="timeToAcknowledgeReceipt"
type="xsd:duration" />
    </xsd:complexType>
</xsd:element>
<xsd:element name="RespondingBusinessActivity">
    <xsd:complexType>
        <xsd:sequence>
            <xsd:element ref="Documentation" minOccurs="0"
maxOccurs="unbounded" />
            <xsd:element ref="DocumentEnvelope"
minOccurs="0" maxOccurs="unbounded" />
        </xsd:sequence>
        <xsd:attribute name="name" type="xsd:string"
use="required" />

```

```

        <xsd:attribute name="nameID" type="xsd:ID"/>
        <xsd:attribute name="isAuthorizationRequired"
type="xsd:boolean" value="false"/>
        <xsd:attribute name="isIntelligibleCheckRequired"
type="xsd:boolean" value="false"/>
        <xsd:attribute name="isNonRepudiationReceiptRequired"
type="xsd:boolean" value="false"/>
        <xsd:attribute name="isNonRepudiationRequired"
type="xsd:boolean" value="false"/>
        <xsd:attribute name="timeToAcknowledgeReceipt"
type="xsd:duration"/>
    </xsd:complexType>
</xsd:element>
<xsd:element name="Start">
    <xsd:complexType>
        <xsd:sequence>
            <xsd:element ref="Documentation" minOccurs="0"
maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:attribute name="toBusinessState"
type="xsd:string" use="required"/>
        <xsd:attribute name="toBusinessStateIDRef"
type="xsd:IDREF"/>
    </xsd:complexType>
</xsd:element>
<xsd:element name="Success">
    <xsd:complexType>
        <xsd:sequence>
            <xsd:element ref="Documentation" minOccurs="0"
maxOccurs="unbounded"/>
            <xsd:element ref="ConditionExpression"
minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="fromBusinessState"
type="xsd:string" use="required"/>
        <xsd:attribute name="fromBusinessStateIDRef"
type="xsd:IDREF"/>
        <xsd:attribute name="conditionGuard">
            <xsd:simpleType>
                <xsd:restriction base="xsd:NMTOKEN">
                    <xsd:enumeration value="Success"/>
                    <xsd:enumeration
value="BusinessFailure"/>
                    <xsd:enumeration
value="TechnicalFailure"/>
                    <xsd:enumeration
value="AnyFailure"/>
                </xsd:restriction>
            </xsd:simpleType>
        </xsd:attribute>
    </xsd:complexType>
</xsd:element>
<xsd:element name="Transition">
    <xsd:complexType>
        <xsd:sequence>
            <xsd:element ref="Documentation" minOccurs="0"
maxOccurs="unbounded"/>

```

```

        <xsd:element ref="ConditionExpression"
minOccurs="0" maxOccurs="1"/>
    </xsd:sequence>
    <xsd:attribute name="onInitiation" type="xsd:boolean"
value="false"/>
    <xsd:attribute name="fromBusinessState"
type="xsd:string"/>
    <xsd:attribute name="fromBusinessStateIDRef"
type="xsd:IDREF"/>
    <xsd:attribute name="toBusinessState"
type="xsd:string"/>
    <xsd:attribute name="toBusinessStateIDRef"
type="xsd:IDREF"/>
    <xsd:attribute name="conditionGuard">
        <xsd:simpleType>
            <xsd:restriction base="xsd:NMTOKEN">
                <xsd:enumeration value="Success"/>
                <xsd:enumeration
value="BusinessFailure"/>
                <xsd:enumeration
value="TechnicalFailure"/>
                <xsd:enumeration
value="AnyFailure"/>
            </xsd:restriction>
        </xsd:simpleType>
    </xsd:attribute>
</xsd:complexType>
</xsd:element>
</xsd:schema>

```

11 References

1. UN/CEFACT Modelling Methodology (CEFACT/TMWG/N090R8E)
2. RosettaNet Implementation Framework: Core Specification, Version: Release 2.00.00, 3 January 2001

12 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。

13 連絡先

チームリーダー（B P チーム）：

Paul Levine
Telcordia Technologies, Inc.
45 Knightsbridge Road
Piscataway, N.J. 08854
US

電話：732-699-3042
電子メール：plevine@telcordia.com

副チームリーダー（Contect/Metamodel グループ）

Karsten Riemer
Sun Microsystems
1 Network Drive
Burlington, MA 01803
USA

電話：781-442-2679
電子メール：karsten.riemer@sun.com

編集（本書）

Karsten Riemer
Sun Microsystems
1 Network Drive
Burlington, MA 01803
USA

電話：781-442-2679
電子メール：karsten.riemer@sun.com

著作権について

Copyright © UN/CEFACT and OASIS, 2001. All Rights Reserved

本書および本書の翻訳版は、上記の著作権通知およびこの段落を含めることを要件とし、自由にその一部または全部をコピーして配布したり、その解説や実施を支援する説明の作成、コピー、刊行、配布などを行ったりしてよい。ただし、英語以外の言語に翻訳する際に必要な場合を除き、著作権通知や ebXML、UN/CEFACT、OASIS などへの参照を取り除くなど、本書自体を変更することは一切してはならない。

上述の制約付き許可は永続的なものであり、ebXML やその継承者や譲受者によって破棄されることはない。

本書および本書に含まれる情報は「無保証」で提供されており、ebXML は、明示、暗示の別を問わず、いかなる保証もしない。これには、本書の情報の使用が他の権利を侵害しないこと、暗示される商品性の保証、特定の目的の適合性などが含まれるが、これらに限定されない。

Copyright Statement

Copyright © ebXML 2001. All Rights Reserved.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this paragraph are included on all such copies and derivative works. However, this document itself may not be modified in any way, such as by removing the copyright notice or references to ebXML, UN/CEFACT, or OASIS, except as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by ebXML or its successors or assigns. This document and the information contained herein is provided on an "AS IS" basis and ebXML DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.