

B2B アプリケーション用メタフレームワークの検討

武田 智義[†] 内藤広志^{††}

[†]大阪工業大学大学院 情報科学研究科 情報科学専攻

^{††}大阪工業大学 情報科学部 情報システム学科

企業間商取引(B2B)を推進する多くの標準化団体によって様々な XML ベースの B2B フレームワークが提案されている。しかし、これらは相互に互換性がないことが問題である。そのため、複数の B2B フレームワークに容易に対応可能な B2B アプリケーション開発が重要である。本研究では、それを実現するための B2B アプリケーション用メタフレームワークを検討した。代表的な B2B フレームワークの RosettaNet および ebXML のビジネスメッセージ構造とその受信処理フローを調査し、両者に共通な受信処理機能を抽出し、メタフレームワークのメッセージ構造と受信処理フローを提案した。

Conceptual Model of Meta Framework for B2B Application

Tomoyoshi TAKEDA[†] Hiroshi NAITO^{††}

[†]Graduate School of Information Science and Technology, Osaka Institute of Technology

^{††}Faculty of Information Science and Technology, Osaka Institute of Technology

The various XML-based Business-to-Business(B2B) frameworks is proposed by the many standardization organizations that promote e-commerce. However, the problem with these frameworks is that they are mutually incompatible. Therefore, B2B application development that can be easily applicable to several B2B frameworks is important. In order to realize it, we examine meta framework for B2B application. we try to compare business message structures and its receiving process flows of the two popular frameworks ,ebXML and RosettaNet, extract those common receiving process functions and propose message structure and its receiving process flow of meta framework.

1. はじめに

インターネットを用いて企業間商取引(B2B)を実現する取り組みが本格化している。B2Bを推進するため、多くの標準化団体が、企業間での取引手段を共通化する枠組みである XML ベースの B2B フレームワークを提案している。代表的な B2B フレームワークとして、RosettaNet[1]と ebXML[2]がある。RosettaNet は、情報機器業界、電子部品業界、半導体製造業界におけるサプライチェーン効率化を目的としている。規定されている仕様には、(1)企業間で行われるビジネスプロセス、交換されるビジネス文書の構造および、その交換手順を規定する Partner Interface Processe(PIP)、(2)PIP を実装するために必要な技術要件を規定する RosettaNet Implementation Framework(RN IF)、(3)ビジネス文書で用いる語彙を規定した辞書に分類できる。一方、ebXML は、特定の業界に限らず、世界的規模でオープンな B2B

を可能にすることを目的としている。規定されている仕様には、取引パートナーと遂行可能なビジネスプロセスの公開/発見するためのメカニズムやメッセージングサービスなどがある。

このように個々の B2B フレームワーク策定は活発に進められており、多くの企業によって B2B フレームワークに基づいた B2B アプリケーションが実装されつつある。しかし、現状では全世界かつ全業界において B2B フレームワークは統一されていない。そのため、取引企業間で B2B フレームワークが異なる場合、相互の B2B アプリケーションの拡張・変更が余儀なくされる。そこで、開発効率を向上させるためにも再利用技術を用いて、複数の B2B フレームワークに容易に対応可能な B2B アプリケーション開発が重要である。本研究では、設計・実装の再利用技術の一つであるフレームワークに着目し、B2B アプリケーション用メタフレームワークを検討した。本報告では、前述し

た RosettaNet および ebXML のビジネスメッセージ構造とその受信処理フローを調査し、比較を行う。また、両者に共通な受信処理機能を抽出し、メタフレームワークのビジネスメッセージ受信処理フローを説明する。

2. ビジネスメッセージの比較

ビジネスメッセージに関しては、RosettaNet は RosettaNet Implementation Framework 2.0(RNIF 2.0)[3]で、ebXML Message Service Specification 1.0(ebMS 1.0)[4]で定義されている。この規格書に基づいて両者のビジネスメッセージ構造を比較する。

2.1 RosettaNet ビジネスメッセージの構造

RosettaNet ビジネスメッセージは、各ヘッダ、サービスコンテンツおよび、必要であれば添付文書を MIME multipart/related コンテントタイプを用いてパッケージ化される。一般的な RosettaNet ビジネスメッセージの構造を図 1 に示す。また、デジタル署名や内容の暗号化を行う場合には S/MIME も用いられる。

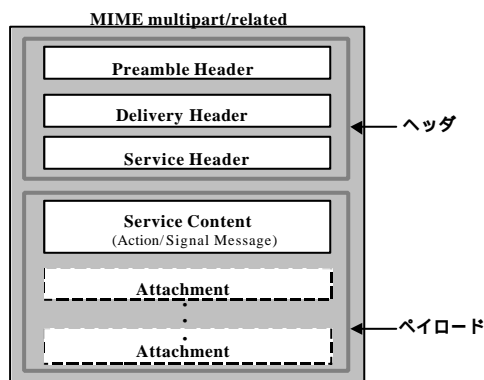


図 1. RosettaNet ビジネスメッセージの構造

ビジネスメッセージを構成する各コンポーネントの概要を次に示す。

(1) Preamble

Preamble は、メッセージ構造が準拠している B2B フレームワーク名・版の情報を含む。

(2) Delivery Header

Delivery Header は、経路とメッセージイン

スタンス情報を含む。送受信者の企業識別番号と所在地や、メッセージ追跡 ID とメッセージ作成日時などの情報が記述される。

(3) Service Header

Service Header は、ペイロードの内容を管理するための情報を含む。送受信者の役割、実行されている PIP 識別情報、ビジネスアクティビティ名、メッセージのタイプ (Action または、Signal) 添付文書などの情報が記述される。

(4) Service Content

Service Content は、各 PIP で規定されているビジネス文書または、受信確認や例外メッセージが記述される。

2.2 ebXML メッセージの構造

ebXML メッセージは、RosettaNet と同様に MIME multipart を用いてパッケージ化され、SOAP Messages with Attachments[5]に従って構造化されている。一般的な ebXML メッセージの構造を図 2 に示す。

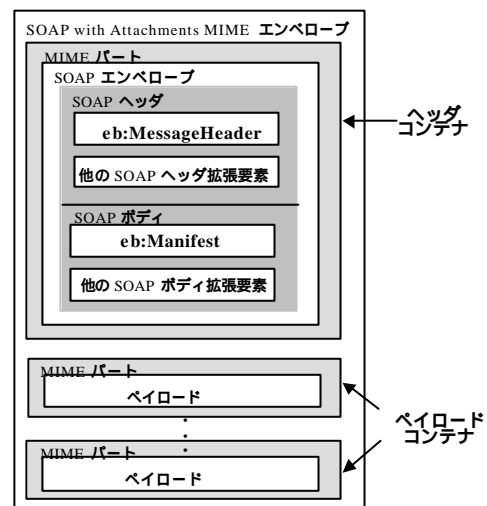


図 2. ebXML メッセージの構造

ヘッダコンテナは、SOAP1.1 準拠メッセージを含む。SOAP ヘッダおよびボディに ebXML 独自の拡張要素が記述されており、その概要を次に示す。

(1) 主な ebXML SOAP ヘッダ拡張要素

必須要素である MessageHeader 要素は、送受信者の識別情報、サービス名、アクション名、

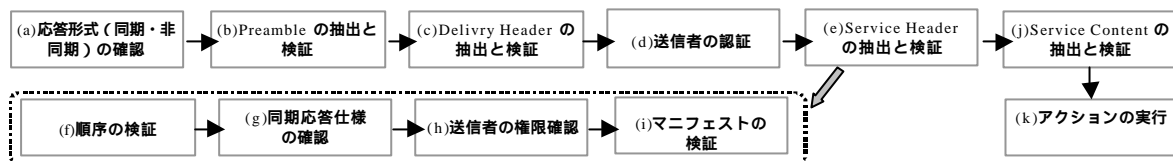


図 3. RosettaNet ビジネスメッセージの受信処理フロー

メッセージ識別子などのメッセージ情報、パートナー間での対話を識別する情報などが記述される。

(2) 主な ebXML SOAP ボディ拡張要素

Manifest 要素は、URL によってペイロードを参照するための情報を含む。

3. ビジネスメッセージ受信処理フローの比較

受信処理フローに関しては、RosettaNet は RNIF2.0 に基づき、ebXML はメッセージサービスと CPA 文書の間を記述した技術書 [6] に基づいて説明する。

3.1 RosettaNet 受信処理フロー

RosettaNet ビジネスメッセージの各コンポーネントの抽出および、検証を行う受信処理フローを図 3 に示す。

(a) 応答形式 (同期・非同期) の確認

HTTP POST を通してメッセージを受信した際、送信者は同じ HTTP コネクションで同期応答を要求する場合がある。RosettaNet では、HTTP ヘッダの x-RN-Response-Type ヘッダの値により同期・非同期を区別する。

(b) Preamble の抽出と検証

Preamble を抽出し、対応する DTD およびメッセージガイドラインに基づいた検証を行う。

(c) Delivery Header の抽出と検証

Delivery Header を抽出し、対応する DTD およびメッセージガイドラインに基づいた検証を行う。

(d) 送信者の認証

(c) で抽出した送信者情報および、メッセージのデジタル署名をもとに送信者認証を行う。なお、送信者が不明の場合、認証は必要ない。

(e) Service Header の抽出と検証

Service Header を抽出し、対応する DTD および、メッセージガイドラインに基づいた検証を行う。暗号化されている場合は、復号化処理を行う。

(f) 順序の検証

受信メッセージが、実行されている PIP インスタンスに関連付けられるか否か、また PIP の最初のメッセージであれば新しい PIP インスタンスが開始されるか否かを検証する。パートナー間で設定された妥当な PIP か否か、インスタンス ID または PIP/Activity/Action コードが妥当な順序に一致しているか否かを検証する。

(g) 同期応答仕様の確認

送信者が同じ HTTP コネクションで同期応答を必要としている場合、対応する PIP が、この受信メッセージに対して同期応答を許可しているか否かを検証する。

(h) 送信者の権限確認

(e) で取得した PIP 情報をもとに、送信者がこの PIP に参加するための権限を持っているか否かを確認する。

(i) マニフェストの検証

受信メッセージが Action メッセージの場合、ペイロードに添付文書が含まれる場合がある。指定された添付文書の数、Content-ID が存在するか否かなどの検証を行う。

(j) Service Content の抽出と検証

Service Content を抽出し、各 PIP の DTD および、メッセージガイドラインに基づいた検証を行う。また、PIP ではビジネス辞書、技術辞書の語彙を用いているため、各辞書に基づいたスキーマ検証も必要である。

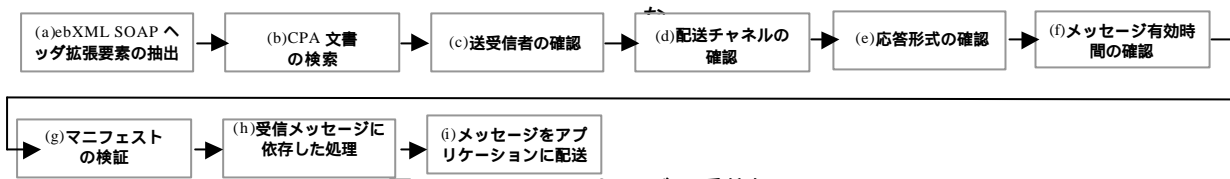


図 4. ebXML メッセージの受信処理フロー

3.2 ebXML 受信処理フロー

ebXML メッセージのヘッダ情報を抽出し、検証を行う受信処理フローを図 4 に示す。

(a) ebXML SOAP ヘッダ拡張要素の抽出

受信処理を行う際、必要となる情報をヘッダから抽出する。送受信者情報、コラボレーションプロトコル合意書(CPA 文書)[7]の識別情報、対話識別情報、メッセージ参照識別子、サービス情報、アクション情報などを抽出する。

(b) CPA 文書の検索

CPA 文書は、取引パートナー間でお互いに合意された XML 文書である。遂行可能なビジネスプロセスに関する情報や、メッセージ交換を行う上での技術的な詳細情報が記述される。受信者は、CPA 識別情報をもとに CPA 文書を検索する必要がある。

(c) 送受信者の確認

指定された CPA 文書が、送受信者間のものであるか否かを確認する。

(d) 配送チャネルの確認

サービス情報から CPA 文書で定義されている配送チャネルを決定する。その配送チャネルと関連する通信プロトコルおよびエンドポイントを特定し、受信メッセージが期待されるチャネルで到着したか否かを確認する。

(e) 応答形式の確認

同期応答パラメータにより同期・非同期応答を区別する。その情報と CPA 文書の同期応答モードと一致するか否かを確認する。また、受信確認要求パラメータにより受信確認が必要か否かの確認も行う。

(f) メッセージ有効時間の確認

受信メッセージが有効時間の値を越えてい

いか否かを確認する。

(g) マニフェストの検証

マニフェストで指定されたすべての MIME 部がペイロードコンテナに存在するか否かを確認する。

(h) 受信メッセージに依存した処理

受信メッセージの配送セマンティクスパラメータおよび信頼性メッセージング手法パラメータの値に応じて受信処理は異なる。ebXML 高信頼性メッセージングを用いる場合、受信者は、重複メッセージ処理、永続的記憶装置へ保存処理、受信確認メッセージの返信処理が求められる。

(i) メッセージをアプリケーションに配送

対話識別情報、サービス情報、アクション情報を用いて、受信メッセージをアプリケーションに配送する。

4. B2B 用アプリケーションメタフレームワークの提案

4.1 メッセージを記述するクラス

RosettaNet および ebXML のビジネスメッセージに共通な情報を格納するため、図 5 に示すクラス図の設計を行なった。BusinessTransactionActivity クラスはビジネストランザクションの実行をあらわす。Party クラスは、当事者をあらわし、送信者/受信者の役割を持つ。BusinessTransaction クラスは、送受信者間のトランザクション情報を持つ。BusinessMessage クラスは、ビジネスメッセージをあらわし、要求/応答メッセージの役割を持つ。Document クラスは、文書をあらわす抽象クラスである。XMLDocument は、XML 文書をあらわし、ビ

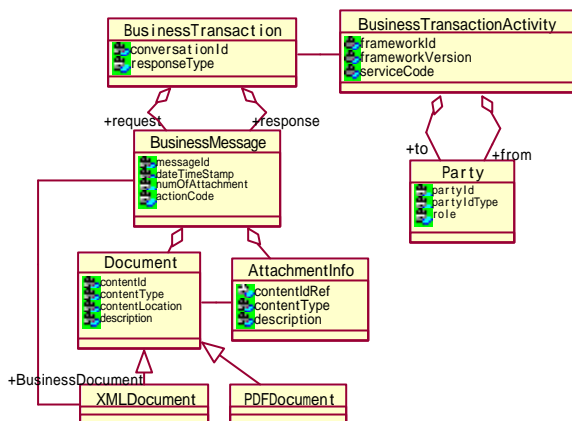


図 5. メッセージ情報を格納するクラス

ビジネス文書の役割を持つ。AttachmentInfo クラスは、ペイロードを参照するためのマニフェスト情報を持つ。PDFDocument クラスは、添付文書として PDF 文書をペイロードに格納して送受信を行う場合に用いられる。

4.2 ビジネスメッセージ受信処理フロー

RosettaNet および ebXML の受信処理フローを比較し、共通な処理機能を抽出する。メタフレームワークの受信処理フローを図 6 に示す。また、RosettaNet および ebXML の受信処理フローとの対応関係を図 7 に示す。

| Meta Framework | RosettaNet | ebXML |
|----------------|------------|-------|
| (1) | * | * |
| (2) | * | * |
| (3) | (a)、(g) | (e) |
| (4) | (c) | (c) |
| (5) | (d) | * |
| (6) | (i) | (g) |
| (7) | (e) | (a) |
| (8) | (f) | (f) |
| (9) | (h) | * |
| (10) | (k) | (i) |

図 7. 受信処理フローの対応関係

なお、図 7 中の“*”は、RNIF2.0 および [6] に記述されていない処理であることを示す。

(1)メッセージの開梱

受信ビジネスメッセージの各 MIME 部を抽

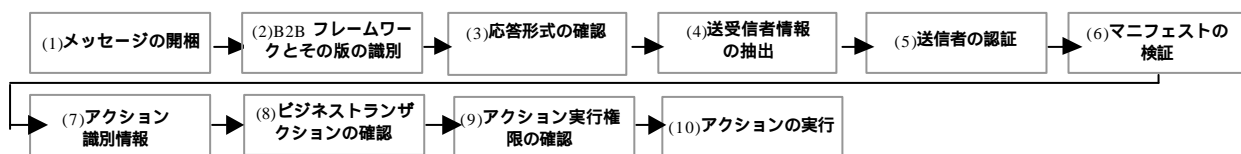


図 6. B2B 用アプリケーションメタフレームワークの受信処理フロー

出する。この結果、それぞれの Content-Type ヘッダを参照し、データのタイプに対応した Document クラスのサブクラスに MIME 部が格納される。また、Content-Transfer-Encoding ヘッダを参照して、Base-64 や Quoted-Printable でエンコード化されていた場合、デコード処理も行う。

(2) B2B フレームワークとその版の識別

受信メッセージが準拠する B2B フレームワークとその版の識別を行う。RosettaNet では、x-RN-Version ヘッダおよび Preamble 情報で識別する。ebXML では、SOAPAction ヘッダおよび、ebXML SOAP エンベロープ拡張に対する名前空間宣言によって識別する。この結果、BusinessTransactionActivity クラスの frameworkId および frameworkVersion 属性に B2B フレームワーク識別情報が格納される。

(3) 応答形式の確認

応答メッセージを同期・非同期で行うか否かを確認する。RosettaNet では処理(a)および(g)に対応し、ebXML では、処理(e)に対応する。この結果、BusinessTransaction クラスの responseType 属性に応答形式情報が格納される。

(4) 送受信者情報の抽出

送受信者を識別するための情報を抽出する。RosettaNet では処理(c)に対応し、ebXML では処理(c)に対応する。この結果、from/to の役割を担う Party クラスの属性に送受信者情報が格納される。

(5) 送信者の認証

(4)で抽出した送信者情報およびデジタル署名があればそれをもとに送信者認証を行う。RosettaNet では処理(d)に対応する。ebXML に

において、対応する処理は、[6]では記述されていない。

(6) マニフェストの検証

マニフェストで指定された添付文書に対して検証を行う。指定された Content-ID をもつ Document オブジェクトがあるかを確認する。RosettaNet では処理(i)に対応し、ebXML では処理(g)に対応する。この結果、AttachmentInfo クラスの属性にマニフェスト情報が格納される。また、BusinessMessage クラスの numOfAttachment 属性に添付文書の数格納される。

(7) アクション識別情報の抽出

受信メッセージからアクションを識別するために必要な情報を抽出する。RosettaNet では処理(e)に対応し、ebXML では(a)に対応する。この結果、BusinessTransaction クラスの actionCode 属性と BusinessTransactionActivity クラスの serviceCode 属性および conversationId 属性にアクション識別情報が格納される。各属性と、RosettaNet および ebXML の要素との関係を図 8 に示す。

| 属性 | RosettaNet | ebXML |
|----------------|--------------------------|----------------|
| serviceCode | pipCode | Service |
| actionCode | GlobalBusinessActionCode | Action |
| conversationId | PipInstanceId | ConversationId |

図 8. クラス属性とメッセージ要素の対応関係

(8) ビジネストラザクシヨンの確認

(7)で抽出したアクション情報をもとに送受信者間での対話の順序を確認する。RosettaNet および ebXML 両者共に処理(f)に対応する。

(9) アクション実行権限の確認

送信者がアクションを実行できる権限を有しているか否かを確認する。RosettaNet では処理(h)に対応する。ebXML において、対応する処理は、[6]では記述されていない。

(10) アクションの実行

BusinessTransaction クラスの conversationId、ServiceCode および actionCode 属性の情

報をもとにアクションを実行する。

5. おわりに

RosettaNet と ebXML のメッセージ構造とメッセージ受信処理フローを比較し、共通する部分を抽出し、B2B アプリケーション用のメタフレームワークとして受信処理フローとそれに必要なクラスを提案した。RosettaNet では、必要な情報がメッセージ内に記述されているが、ebXML では CPA 文書に記述された情報を参照しなければならない点異なる。しかし、両者の受信処理フローは統一できることがわかった。今後は、メタフレームワークの受信処理フローをどのように実装するかを検討したい。

参考文献

- [1] RosettaNet, <http://www.rosettanet.org/>
- [2] ebXML, <http://www.ebxml.org/>
- [3] RosettaNet, “RosettaNet Implementation Framework v.2.0”, <http://www.rosettanet.org/>
- [4] ebXML, “Message Service Specification v.1.01”, <http://www.ebxml.org/specs/ebMS.pdf>
- [5] W3C, “SOAP Messages with Attachments”, <http://www.w3.org/TR/2000/NOTE-SOAP-attachments-20001211>
- [6] Chan, Arvala, “Preliminary draft for appendix on how to use the Messaging Service with a CPA”, <http://lists.oasis-open.org/archives/ebxml-cppa/200112/msg00013.html>
- [7] ebXML, “Collaboration-Protocol Profile and Agreement Specification v.1.0”, <http://www.ebxml.org/specs/ebCCP.pdf>