

プラットフォーム通信標準仕様

V3.1



一般財団法人全国地域情報化推進協会

目次

本書の位置づけ	1
1. プラットフォーム通信標準仕様とは	3
2. プラットフォーム通信仕様	4
2. 1 通信プロトコルの階層	4
2. 2 通信プロトコルと通信セキュリティ	9
2. 3 SOAP 通信と通信モデル	9
2. 4 高信頼性通信機能	15
3. プラットフォーム通信標準のメッセージ定義仕様	16
3. 1 XML 定義の策定方針	16
3. 2 XML 定義仕様	18
3. 3 WSDL の XML 定義記述要件	27
4. プラットフォーム通信標準のビジネスプロセス制御定義仕様	29
4. 1 定義	29
4. 2 プロセス制御手段に関する利用規定	30
4. 3 ビジネスプロセス (BP) のフロー制御に関する規定	31
4. 4 WS-BPEL の適用範囲、制約に関する規定	33
4. 5 メッセージ交換パターンに対する WS-BPEL 実装規定	35
4. 6 追加規定事項	40
5. プラットフォーム通信標準における認証・認可・セキュリティ機能	43
5. 1 対象領域	43
5. 2 異なるサイト間におけるセキュリティ上の課題と対策技術	43
5. 3 異なるサイト間における認証・認可・セキュリティ仕様	44
6. プラットフォーム通信機能における MEP と異常系処理	55
6. 1 概念、用語の定義	55
6. 2 プラットフォーム通信機能における推奨メッセージ交換パターンの定義	57
6. 3 プラットフォーム通信機能の通信障害とその検知方法定義	59
6. 4 プラットフォーム通信機能における異常系処理	64
6. 5 データ交換システムパターンと、メッセージ交換パターンの関係	65
6. 6 その他の規定事項	70
7. プラットフォーム通信仕様におけるメッセージ共通ヘッダ仕様	74
7. 1 共通ヘッダで規定するデータ項目	74
7. 2 共通ヘッダの各項目の設定方法	79
付録 1. ビジネス電文の例	85

本書の位置づけ

地域情報プラットフォーム標準仕様（以降、地域情報 PF 仕様と表記）は、業務モデル標準、サービス協調技術標準の仕様、およびガイドライン形式である。

本書「プラットフォーム通信標準仕様」は、サービス協調技術標準の仕様に位置づけられ、地域情報プラットフォーム（以降、地域情報 PF と表記）の通信手順・方式について規定するものである。

	業務モデル標準	サービス協調技術標準
地域情報プラットフォーム標準仕様書	自治体及び民間が提供する地域情報サービスの連携に必要な業務アプリケーションユニットのインターフェース仕様 ◆自治体業務アプリケーションユニット標準仕様 ◆防災業務アプリケーションユニット標準仕様 ◆教育情報アプリケーションユニット標準仕様 ◆健康情報業務アプリケーションユニット標準仕様※	サービス連携を支える基盤アプリの諸要件・プロトコル等を取り決めた仕様 ◆アーキテクチャ標準仕様 ◆プラットフォーム通信標準仕様
	GISを活用した業務ユニット、アプリケーションを構築するための共通仕様 ◆GIS共通サービス標準仕様	
	各種システム製品等の地域情報プラットフォーム準拠及び相互接続を確認する仕様 ◆地域情報プラットフォーム準拠確認及び相互接続確認仕様	
	仕様に準拠したサービス基盤および業務アプリケーションを導入する関連者向けに必要な事項をとりまとめたもの(指針) ◆地域情報プラットフォームガイドライン	
その他	◆地域情報プラットフォーム基本説明書 ◆地域情報プラットフォーム標準仕様運用規則	
資料参考	◆地域情報プラットフォームガイドライン 技術解説 要約 ◆地域情報プラットフォームにおけるGIS共通サービス基本提案書	

※ 今後、標準仕様体系に取り込み予定

図 0. 1 地域情報 PF 仕様の体系

標準仕様書における表記の説明

- ・ サイト内とサイト間の仕様差異の表現方法

本仕様書においてサイト内とサイト間で仕様に差異がある場合の表記方法を下記に示す。

- 【サイト内】の定義 : 自治体内システム間を連携する場合の仕様を規定する箇所を識別する
- 【サイト間】の定義 : 自治体間や自治体と民間のそれぞれのシステムを連携する場合の仕様を規定する箇所を識別する
- 【共通】の定義 : サイト内とサイト間で同一の仕様を規定する箇所を識別する

・ PF 準拠確認チェックルール表記の説明

PF 準拠確認チェックルールとは、地域情報 PF のサービス基盤の機能を本仕様書で定めた規定事項にそって機能単位で列挙したものである。これはベンダ等が提供(または提供を予定)する各種システム製品が、地域情報 PF 仕様に準拠しているかどうかの確認に利用することができる (PF 準拠についての詳細は、「地域情報プラットフォーム準拠確認および相互接続確認仕様」を参照)。

PF 準拠確認チェックルール表記方法を下記に示す。

- ◆ 準拠ルール識別番号 : 準拠ルールを識別するための番号
- ◆ 準拠ルール本文 : 準拠ルールの内容を表す文
- ◆ 準拠ルール本文の語尾 : 準拠ルールが (必須)、(オプション)、(推奨) のいずれかを表す
- ◆ 章節項番号 : 準拠ルールの元となった仕様書の本文の記載箇所を特定するための章、節、項の番号

記載例 :

<<準拠ルール>>

【共通】

(CS-R020001) : TCP/ IP による通信が可能である。(必須) [2.2.1]

(CS-R020002) : HTTP は、HTTP 1.1 を使用する。(必須) [2.2.1]

・
・

【サイト内】

(CS-R020008) : データ交換システムパターンのサポートはオプションである。(オプション)
[2.3.2 (2)]

・

【サイト間】

(CS-R020010) : データ交換システムパターンのサポートはオプションである。(オプション)
[2.3.2 (2)]

・

1. プラットフォーム通信標準仕様とは

プラットフォーム通信標準仕様では、サービスを提供するコンピュータと、サービスを使用するコンピュータ間の通信の取り決めを定める。規定する項目は下記の通りである。

プラットフォーム通信仕様：

- 1) 通信プロトコル
- 2) 通信セキュリティ
- 3) サービス通信
- 4) 高信頼性通信機能
- 5) 電子封筒形式（メッセージ形式）
- 6) ビジネス電文の分割、圧縮、アーカイブ
- 7) ビジネス電文のセキュリティ
- 8) 認証・認可・セキュリティ
- 9) MEP と異常系処理
- 10) 共通ヘッダ仕様

プラットフォーム通信標準のメッセージ定義仕様：

- 1) XML (Extensible Markup Language) 定義の策定方針
- 2) Web サービス I/F の定義仕様
- 3) メッセージの XML 定義仕様

プラットフォーム通信標準のビジネスプロセス制御定義仕様：

- 1) ビジネスプロセス表記の仕様
- 2) BPEL (Business Process Execution Language) /WSDL (Web Service Description Language) の XML 定義記述要件
- 3) BPEL 定義の記述要件

オプションの定義は以下の通りとする。

ユースケースによって、要不要の判断が必要となるもの。

なお、平成 25 年 5 月 24 日に成立した「行政手続における特定個人を識別するための番号の利用等に関する法律（以降、番号法と表記）」と「行政手続における特定の個人を識別するための番号の利用等に関する法律の施行に伴う関係法律の整備等に関する法律（以降、整備法と表記）」を踏まえ、「社会保障・税に関わる番号制度（以降、番号制度と表記）」が導入されることとなった。

番号制度の導入で推進されている地方公共団体等の情報システムの仕組みにおいては、ある自治体から他の団体に特定個人情報等の情報項目をやり取りする際に、自治体内の様々な業務システムは中間サーバを経由して他の団体への情報提供を行うこととなる。したがって、自治体内の業務システムと中間サーバの間での明確な通信規約が存在し守られていることは重要事項のひとつである。

この通信規約について『「地方公共団体における情報連携プラットフォームに係る中間サーバー・ソフトウェアの設計・開発作業の請負」システム方式設計書（以降、中間サーバ・システム方式設計書と表記）』では、本仕様で前述した通信の規定項目の多くを通信仕様として活用している。よって、こうした規定項目に準ずることで中間サーバとの通信が可能となる。ただし、中間サーバ・システム方式設計書として特段の指定要件がある場合には、そちらの要件を優先することを本仕様として推奨する。

2. プラットフォーム通信仕様

2. 1 通信プロトコルの階層

図2. 1に、通信プロトコルの階層を示す。

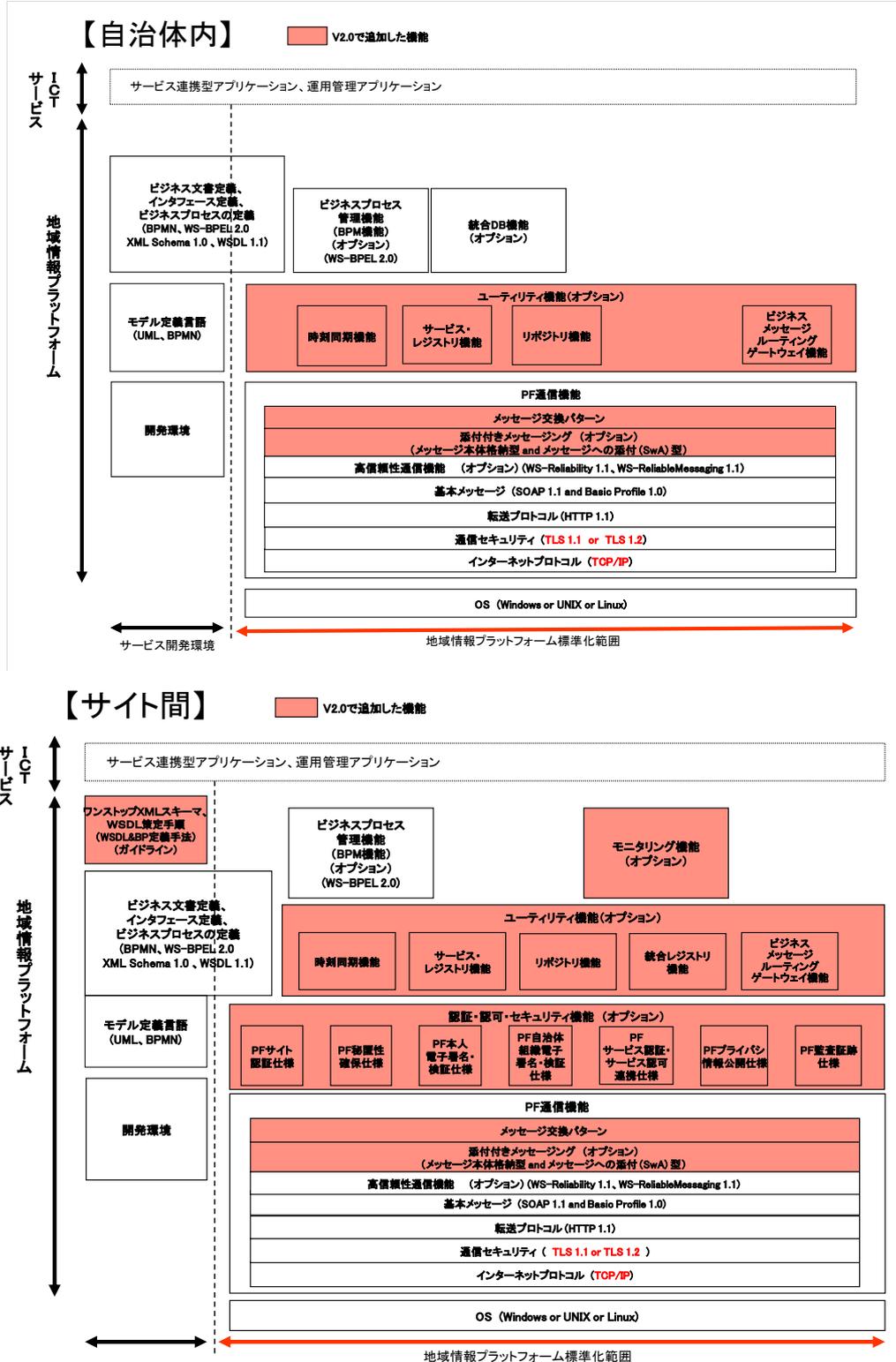


図2. 1 技術マップ

技術マップ上の国際標準について、その概説を行う。

「オプション」「等」が記されているものは、その代表的なもののみを記載する。

なお、仕様概要の記述にあたり、以下の資料を参考とした。「平成 15 年度 Web サービス相互運用技術の調査研究報告書」(財) 情報処理相互運用技術協会編集・発行 (平成 16 年 3 月発行)

「仕様の確定状況」欄に記載されている用語の意味は以下の通りである。

標準策定団体で標準として制定された状態

OASIS (Organization for the Advancement of Structured Information Standards) における OASIS Standard、W3C (World Wide Web Consortium) における W3C Recommendation など

ノート標準化団体に提出された仕様

W3C における W3C Note (現在は W3C Member Submission)

表 2. 1 国際標準の概説

項番	規格名 (略称)	策定団体	仕様の確定状況	仕様の概要	仕様の URL
1	XML 1.0	W3C	標準	文書やデータの意味や構造を記述するための拡張可能なマークアップ言語。利用者が自由にタグを定義でき、文書中の文字列に意味付けができる言語構造を持ち、文書処理から電子商取引にいたるまでネットワーク上のデータ処理のあらゆる場面で広く活用されている。	Extensible Markup Language (XML) 1.0 (Fourth Edition) W3C Recommendation 16 August 2006, edited in place 29 September 2006 http://www.w3.org/TR/2006/REC-xml-20060816
2	XML Schema 1.0	W3C	標準	XML のスキーマ言語の 1 つである。W3C が、XML でのすべてのニーズに対応する唯一のスキーマ言語として策定した仕様である。XML Schema の構文は XML と同じで、データ型の定義や名前空間が扱えるなど、DTD (Document Type Definition) にはない特徴を持っているが、仕様が複雑で理解を困難にしているという問題もある。	XML Schema Part 1: Structures Second Edition W3C Recommendation 28 October 2004 http://www.w3.org/TR/2004/REC-xmlschema-1-20041028/ XML Schema Part 2: Datatypes Second Edition W3C Recommendation 28 October 2004 http://www.w3.org/TR/2004/REC-xmlschema-2-20041028/
3	WS-BPEL 2.0	OASIS	標準	Web サービスの連携を実現するためのビジネスプロセスの挙動を定義する仕様である。WS-BPEL は WSDL により定義された Web サービスのインタフェースを連携させ、複雑な Web サービスのプロセスを	Web Services Business Process Execution Language Version 2.0 OASIS Standard 11 April 2007 http://docs.oasis-open.org/

項番	規格名 (略称)	策定 団体	仕様の 確定 状況	仕様の概要	仕様の URL
				定義するものである。プロセスの振る舞いの記述とメッセージの振る舞いを記述する仕様から構成される。	wsbpel/2.0/OS/wsbpel-v2.0-0S.pdf
4	WS-Reliability 1.1	OASIS	標準	Web サービス利用時のメッセージに信頼を持たせるための仕様であり、SOAP に信頼性向上の仕様を付加するものである。本仕様では、アプリケーションレベルでの非同期メッセージと信頼性のための3つの機能（送達保証、重複削除、順序保証）を SOAP に追加している。	Web Services Reliable Messaging TC WS-Reliability 1.1 OASIS Standard, 15 November 2004 http://docs.oasis-open.org/wsrn/ws-reliability/v1.1/wsrn-ws_reliability-1.1-spec-os.pdf
5	WS-Reliable Messaging 1.1	OASIS	標準	同上	Web Services Reliable Messaging (WS-Reliable Messaging) Version 1.1 OASIS Standard 14 June 2007 http://docs.oasis-open.org/ws-rx/wsrn/200702/wsrn-1.1-spec-os-01.pdf
6	SOAP 1.1	W3C	ノート	SOAP はネットワーク経由でオブジェクト間の通信を行う軽量の通信プロトコルである。通信内容の記述に XML を用いているため、言語やプラットフォームに依存しない通信プロトコルである。また、SOAP はデータ構造のみが規定されており、転送プロトコルとして HTTP (Hypertext Transfer Protocol) など既存の任意の通信プロトコルを使用する。	Simple Object Access Protocol (SOAP) 1.1 W3C Note 08 May 2000 http://www.w3.org/TR/2000/NOTE-SOAP-20000508
7	WSDL 1.1	W3C	ノート	Web サービスが提供する機能を記述するための、XML ベースの言語仕様の1つ。各 Web サービスがどのような機能を提供するのか、その機能を利用するために、どのようなパラメータを渡す必要があるかなどを標準的なデータ仕様で記述できるようにし、Web サービス同士のコミュニケーションを可能にする。	Web Services Description Language (WSDL) 1.1 W3C Note 15 March 2001 http://www.w3.org/TR/2001/NOTE-wsdl-20010315

項番	規格名 (略称)	策定 団体	仕様の 確定 状況	仕様の概要	仕様の URL
8	WS-Security 1.1	OASIS	標準	Web サービスでのセキュリティ確保のための仕様である。WS-Security は SOAP メッセージに対する暗号、署名の方法やセキュリティ・トークン（ユーザ/パスワードや X.509 証明書などのセキュリティ情報）を SOAP メッセージに格納する方法を規定している。これにより、メッセージの完全性、秘匿性確保、送信者認証を実現する。	Web Services Security: SOAP Message Security 1.1 (WS-Security 2004) OASIS Standard Specification, 1 February 2006 http://www.oasis-open.org/committees/download.php/16790/wss-v1.1-spec-os-SOAPMessageSecurity.pdf
9	SOAP 1.2	W3C	標準	SOAP 1.2 は、SOAP 1.1 を基にして W3C において策定された標準仕様であり、ネットワーク経由でオブジェクト間の通信を行う軽量の通信プロトコルである。SOAP 1.1 とは、仕様の詳細が異なり、互換性は無い。	SOAP Version 1.2 Part 1: Messaging Framework (Second Edition) W3C Recommendation 27 April 2007 http://www.w3.org/TR/2007/REC-soap12-part1-20070427/ SOAP Version 1.2 Part 2: Adjuncts (Second Edition) W3C Recommendation 27 April 2007 http://www.w3.org/TR/2007/REC-soap12-part2-20070427/
10	UDDI 2.0, 3.0	OASIS	標準	Web サービスのレジストリ作成、レジストリへのアクセス制御方法、記録をほかのレジストリに転送する仕組みについて通信プロトコル仕様を定めている。3.0 では、複数の UDDI レジストリでツリー構造を構築する仕組みの導入、検索機能や多言語対応の強化などの改善が図られている。	UDDI Version 2 Specifications http://www.oasis-open.org/committees/uddi-spec/doc/tcpspecs.htm#uddiv2 UDDI Version 3 Specification http://www.oasis-open.org/committees/uddi-spec/doc/tcpspecs.htm#uddiv3
11	ebXML RIM 3.0, RS 3.0	OASIS	標準	いずれも ebXML (Electronic Business using eXtensible Markup Language) のレジストリサービス仕様である。具体的にはレジストリ情報モデル (Registry Information Model : RIM) とレジストリサービス仕様 (Registry Services Specification : RS) とからなり、RIM はレジストリで管理される登録データ	ebXML Registry Information Model Version 3.0 OASIS Standard, 2 May, 2005 ebXML Registry Services and Protocols Version 3.0 OASIS Standard, 2 May, 2005 http://docs.oasis-open.org/regrep/v3.0/regrep-3.0-os.z

項番	規格名 (略称)	策定 団体	仕様の 確定 状況	仕様の概要	仕様の URL
				のメタデータを定義し、RS は RIM で定義されたデータにアクセスするためのインタフェースを記述する。	ip
12	UML	OMG (Object Managem ent Group)	標準	Unified Modeling Language。 ユースケース図、クラス図、オブジェクト図、シーケンス図、コラボレーション図、状態チャート図、実装モデル、コンポーネント図などを含むオブジェクト指向による分析／設計の表現言語。	http://www.uml.org/
13	BPMN	OMG BPMI. or g (Busine ss Process Managem ent Initiat ive)	標準	Business Process Modeling Notation。 ビジネスプロセスモデリング表記。	http://www.bpmn.org/
14	Basic Profile 1.0	WS-I (Web Service s Interop erabili ty Organiz ation)	標準	Web サービス仕様群から構成される基本プロファイルで、相互運用性を向上させる仕様の明確化や説明を含んでいる。	Basic Profile Version 1.0 Final Material Date: 2004/04/16 19:06:16 http://www.ws-i.org/Profiles/BasicProfile-1.0-2004-04-16.html

2. 2 通信プロトコルと通信セキュリティ

2. 2. 1 通信プロトコル

- TCP/IP による通信が可能なこと。
- TCP/IP の上位レイヤとして、HTTP は HTTP 1.1 を採用する。
候補仕様 (HTTP 1.0、HTTP 1.1、SMTP (Simple Mail Transfer Protocol) (メール)) のうち、リアルタイム性と普及実績から HTTP 1.1 を採用する。

2. 2. 2 通信セキュリティ

セキュリティに関する詳細については「5. プラットフォーム通信標準における認証・認可・セキュリティ機能」を参照のこと。

2. 3 SOAP 通信と通信モデル

2. 3. 1 SOAP 通信

- サービス間の通信仕様として、SOAP 1.1 を採用する。
なお、相互接続性のため、WS-I で策定された Basic Profile 1.0 に準拠するものとする。
サービス連携実現には、インターネットやファイアウォールと相性が良いことが必要であり、HTTP へのバインディングが規定されている SOAP を採用する。また、標準化の状況、製品やオープンソースでの実装状況を考慮して SOAP 1.1 を採用する。

2. 3. 2 通信モデル

(1) 電子封筒形式 (メッセージ形式)

PF 通信で交換されるメッセージに関する用語を定義する。

(a) 交換される情報の種類

- ・ ビジネスメッセージ： 業務を実施するために使われるメッセージ。
(例、申請メッセージ、業務処理結果メッセージ、等)
- ・ システムメッセージ： 業務以外にシステム運用上使われるメッセージ。
(例、モニタリング機能用メッセージ、等)

(b) 交換される情報構成 (1つのメッセージを構成する要素)

- ・ 共通ヘッダ： サービス通信の情報や BPM 機能の制御のための共通ヘッダ情報。共通ヘッダについては、「7. プラットフォーム通信仕様におけるメッセージ共通ヘッダ仕様」で詳しく説明する。
- ・ 本体： 業務項目 (本文) や共通ヘッダ項目から構成されるメッセージの本体情報。

(c) 交換される情報の種類

- ・ 本体： 共通ヘッダと本文から構成される XML メッセージ
- ・ 添付ファイル： 任意のファイル形式

PF 通信では、ビジネスメッセージやシステムメッセージを、電子封筒と呼ばれる SOAP 電子封筒形式に格納し、通信を行う。PF 通信で採用する 2 種類の電子封筒形式を図 2.2 に示す。PF 通信機能（プラットフォーム通信機能）はこの 2 種類の両方の電子封筒形式をサポートしなければならない。

PF通信で使用する電子封筒形式は、下記の2つ。

- (A)ビジネスメッセージ用: ビジネス電文(申請や提供依頼、ワンストップ処理の情報交換等)、及び、業務ユニット間のデータ交換で使用するメッセージ用の電子封筒
- (B)システムメッセージ用: モニタリング機能や認証、メッセージ受領等、システム管理のためのメッセージ用の電子封筒

電子封筒形式 (※)添付ファイルがある場合は、別途規定

(A)ビジネスメッセージ用

(B)システムメッセージ用

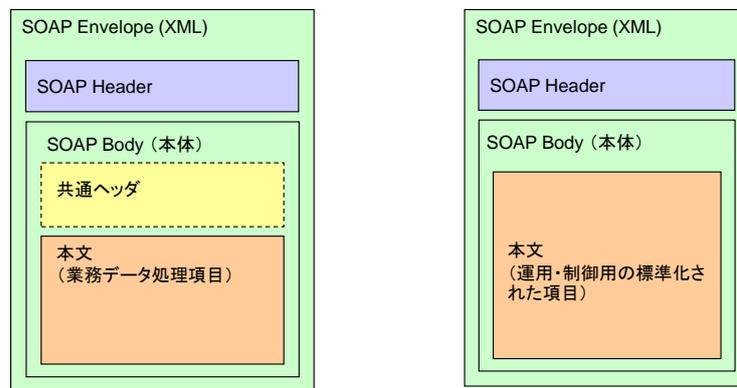


図 2. 2 PF 通信で採用する電子封筒形式

(2) 添付ファイルがある場合の電子封筒形式

PF 通信機能において添付ファイルの取り扱いはオプションであるが、添付ファイルを扱う場合には、次の図 2.3、図 2.4 に示す 2 つの方式「メッセージ本体格納型」と「メッセージへの添付 (SwA: SOAP Messages with Attachments) 型」のいずれも実施できる機能を持つ必要がある。

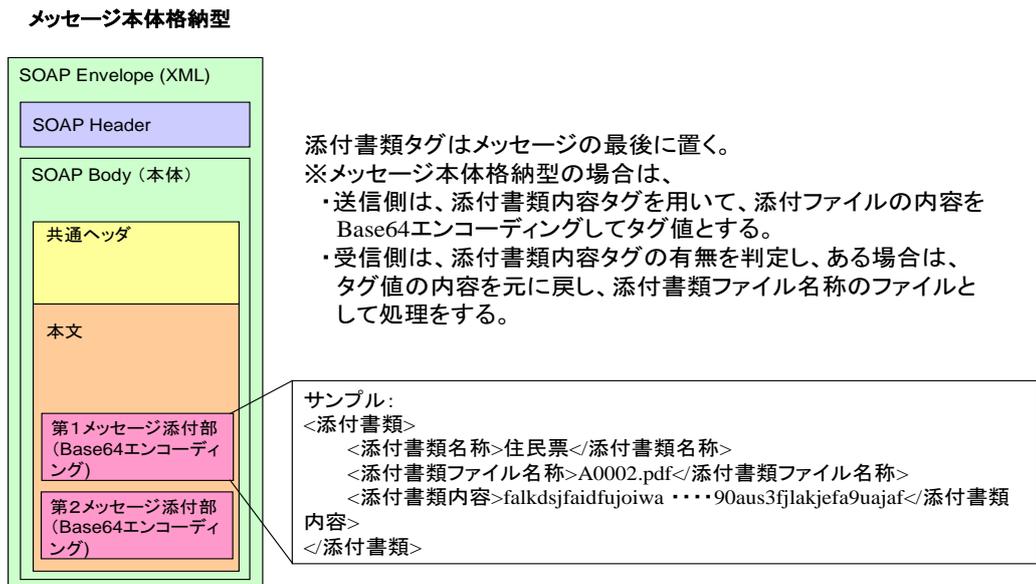


図 2. 3 メッセージ本体格納型で添付ファイルを扱うための電子封筒形式

メッセージへの添付 (SwA: SOAP Messages with Attachments) 型

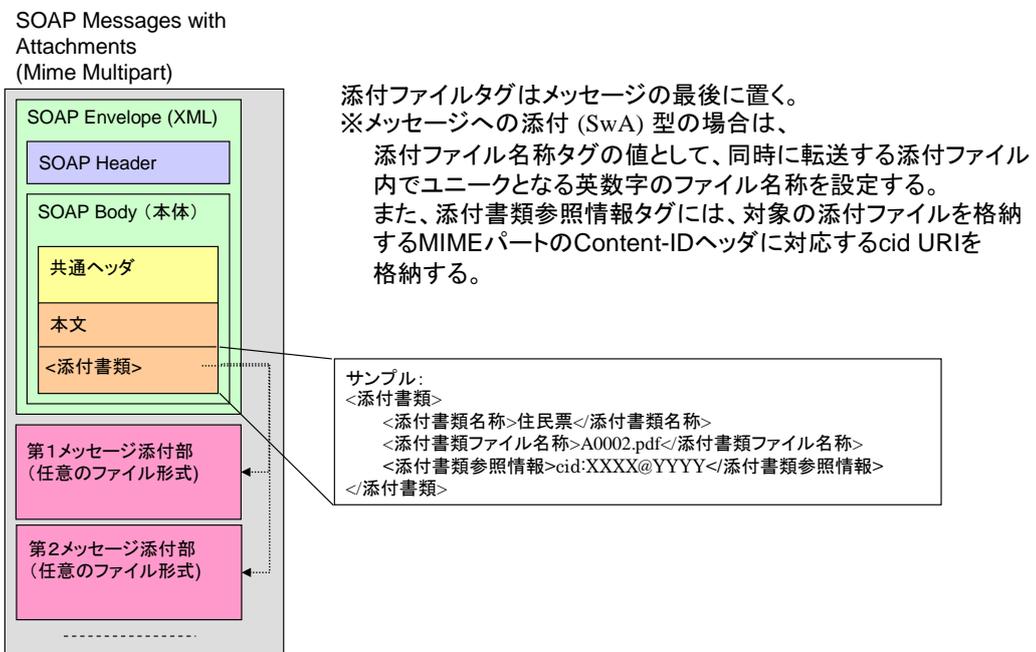


図 2. 4 メッセージへの添付 (SwA: SOAP Messages with Attachments) 型で添付ファイルを扱うための電子封筒形式

次に、添付ファイルを扱う場合のデータ項目名とXMLスキーマ作成仕様を表2. 2に示す。

表2. 2 添付ファイルを扱う場合のデータ項目名とXMLスキーマ作成仕様

No.	データ項目	データ型 (XML 定義用)	最小 文字数	最大 文字数	最小 出現回数	最大 出現回数	説明
1	添付書類	添付書類情報					・メッセージ内の出現回数 を讀込むスキーマで定義
2	添付書類名称	string	(※1)	(※1)	1	1	
3	添付書類ファイル名	string	(※1)	(※1)	1	1	
4	添付書類内容	string	(※1)	(※1)	1	1	・本表の[No. 4 添付書類内 容の項目]か、[No. 5 添付書 類参照情報の項目]のどち らか一方を必ず使う ・添付ファイルがある場合 の電子封筒形式でメッ セージ本体格納型を選択 した場合は、これを使う ・Base64 エンコーディング する
5	添付書類参照情報	ref:swaRef	(※1)	(※1)	1	1	・[4 添付書類内容の項目] か、[5 添付書類参照情報の 項目]のどちらか一方を必 ず使う ・添付ファイルがある場合 の電子封筒形式で SwA 型を 選択した場合は、これを使 う

(※1)最小文字数・最大文字数に関しては、本仕様では規定していない。

表2. 2の「添付書類参照情報」で使用している ref:swaRef というデータ型は、添付ファイルを参照するために WS-I Attachments Profile 1.0 (WS-I AP) の中で定義されているもので、XML Schema の xs:anyURI データ型をベースにしている。ref の示す名前空間は <http://ws-i.org/profiles/basic/1.1/xsd> である。

また、添付ファイルを扱う方式として図2. 3、図2. 4に示した「メッセージ本体格納型」か「メッセージへの添付 (SwA: SOAP Messages with Attachments) 型」を選択するにあわせ、XML 定義においても「添付書類内容」と「添付書類参照情報」のいずれか一方が必ず出現する。

なお、「メッセージへの添付 (SwA: SOAP Messages with Attachments) 型」の電子封筒形式を採用する場合は、添付ファイルの相互接続性に関わる WS-I AP の規定について、実状を考慮して採用を検討することが望ましい。

(3) データ交換システムパターン

【共通】

本節にて、PF 通信機能が想定する、データ交換システムのパターンを規定する。データ交換システムパターンのサポートはオプションである。

添付ファイルや統合 DB 機能、BPM 機能を考慮した場合、下記の 5 種類のデータ交換システムパターンのタイプがある。

- [Type1] 本文と添付（本文内に埋め込み）型
- [Type2] 本文と添付（MIME で添付）型（SwA 型）
- [Type3] 本文とリファレンス（統合 DB 機能の検索キーを交換）型
- [Type4] BPM 機能と連携した本文と添付（本文内に埋め込み）型
- [Type5] BPM 機能と連携した本文とリファレンス（統合 DB 機能の検索キーを交換）型

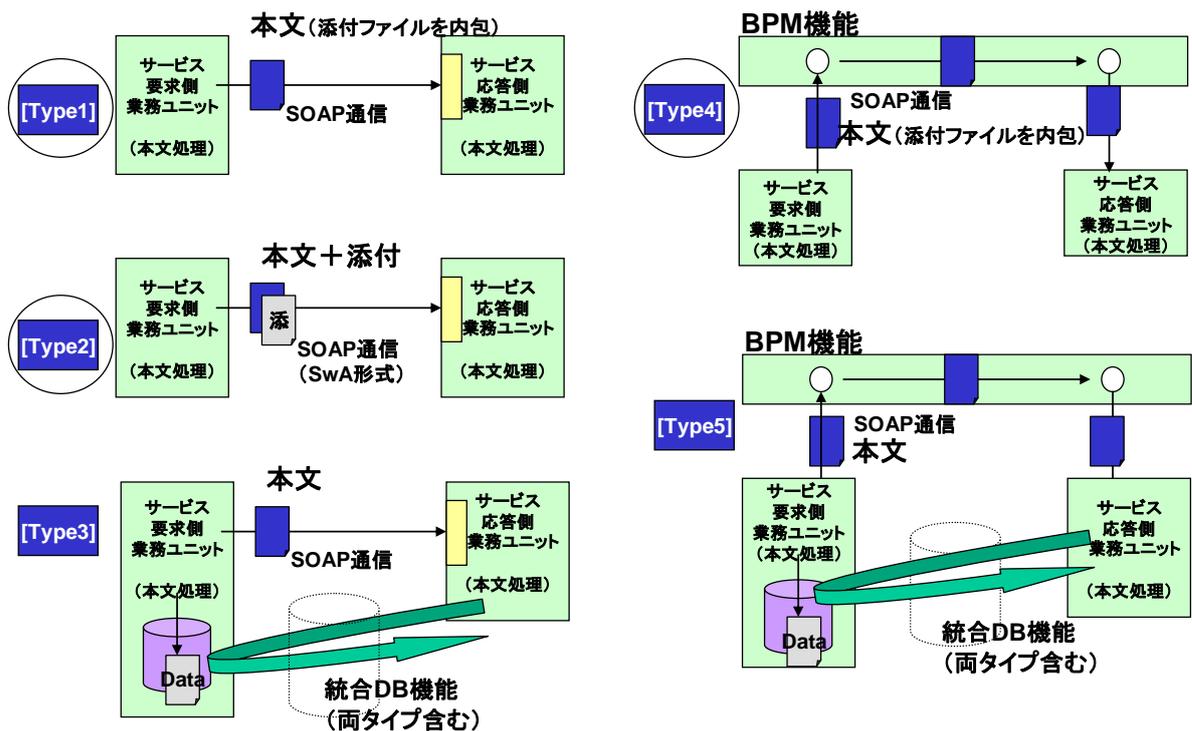


図 2. 5 データ交換システムパターン

【サイト内】

サイト内は、5 つのデータ交換システムパターンのいずれかを採用すること。

【サイト間】

サイト間のデータ交換システムパターンでは、下記の 3 種類を採用する。データ交換システムパターンをサポートする場合、3 種類のどれでも使用できる必要がある。

- [Type1] 本文と添付（本文内に埋め込み）型
- [Type2] 本文と添付（MIME で添付）型（SwA 型）
- [Type4] BPM 機能と連携した本文と添付（本文内に埋め込み）型

(4) データ交換システムパターン別の電子封筒形式

データ交換システムパターン別の電子封筒形式について、図 2.6 に規定する。

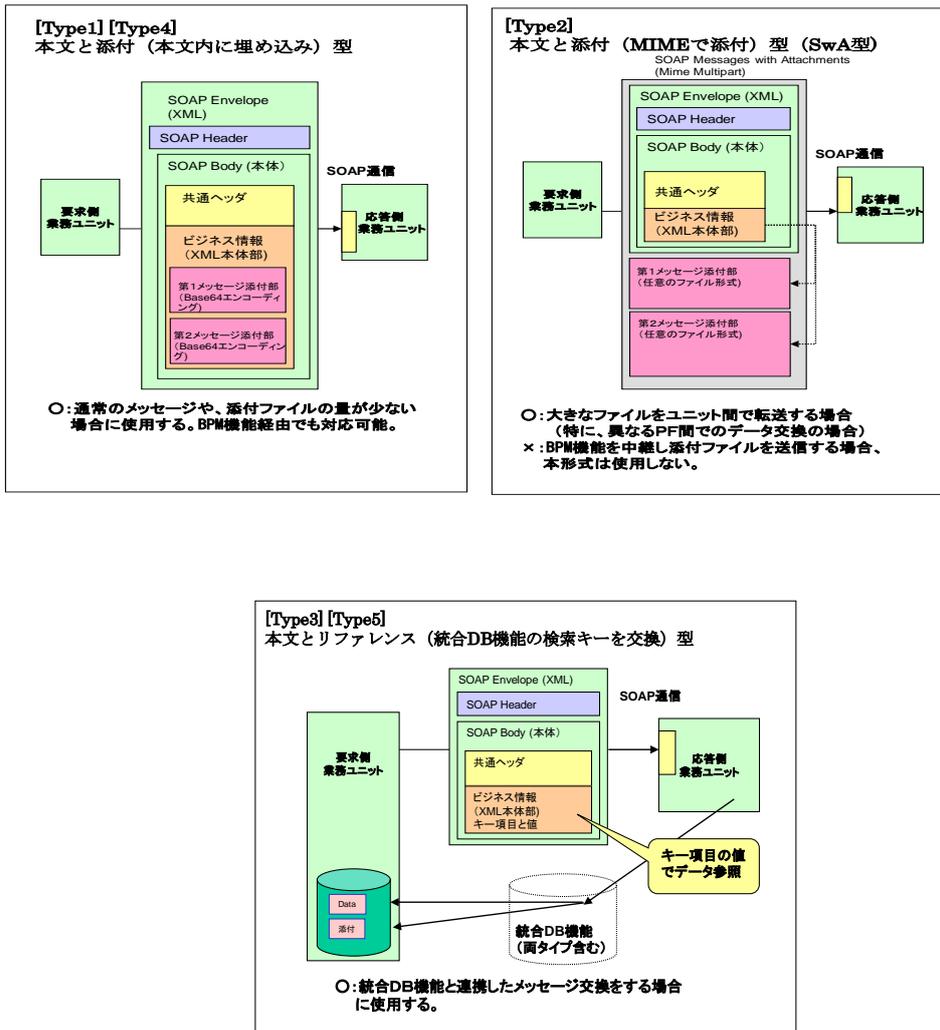


図 2. 6 データ交換システムパターン別の電子封筒形式とその特徴

(5) ビジネス電文の分割、圧縮、アーカイブ
特に規定しない。

(6) ビジネス電文のセキュリティ

詳細は、「5. 3 異なるサイト間における認証・認可・セキュリティ仕様」を参照。

2. 4 高信頼性通信機能

高信頼性通信機能は SOAP 通信処理の信頼性を向上させるものであり、メッセージの送達保障（失敗時再送）、重複排除、順序保障の機能を持つ。高信頼性通信機能は、オプションである。

SOAP を高信頼にする仕様として、「WS-Reliability 1.1」と「WS-ReliableMessaging 1.1」がある。両仕様は送達保証、重複排除、順序保証の機能としては同等であるが、異なる仕様であり、相互接続性が確保できないこと、いずれの仕様とも十分普及する状況に至っていないことから、継続して普及動向を監視し、仕様化を検討する。

高信頼性通信機能を使用する場合は、その内部で処理される障害（再送可能な失敗、重複受信など）はその都度上位アプリケーション（BPM 機能など）に通知するのではなく、最終的な障害（再送がすべて失敗した場合など）のみを通知することが望ましい。

（業務的な受領の確認（受領 Ack）が必要な場合）

BPM 機能と業務サービス間では、BPM 機能からサービスが呼び出され、業務サービス側で職員が処理を行うことが自治体要件として発生する。その際、メッセージ交換の高信頼性を確保することが必要な場合に使用する。具体的には、受信したサービス側で、ビジネス電文の検定と業務システムへの取りこみが正常に行えたという結果を、受領 Ack として送信元のクライアントへ通知する。詳細については「6. プラットフォーム通信機能における MEP と異常系処理」を参照のこと。

<<準拠ルール>>

【共通】

- (CS-R020001) : TCP/IP による通信が可能である。(必須) [2.2.1]
- (CS-R020002) : HTTP は、HTTP 1.1 を使用する。(必須) [2.2.1]
- (CS-R020003) : SOAP は、SOAP 1.1 を使用する。(必須) [2.3.1]
- (CS-R020004) : SOAP 通信は Basic Profile 1.0 に準拠すること。(必須) [2.3.1]
- (CS-R020005) : ビジネスメッセージ用とシステムメッセージ用の両方の電子封筒形式を使用できること。(必須) [2.3.2 (1)]
- (CS-R020006) : 添付ファイルのサポートはオプションである。(オプション) [2.3.2 (4)]
- (CS-R020007) : 添付ファイルをサポートする場合 (CS-R020006)、メッセージ本体格納型とメッセージへの添付 (SwA) 型の両方の方式を使用できること。(必須) [2.3.2 (4)]

【サイト内】

- (CS-R020008) : データ交換システムパターンのサポートはオプションである。(オプション) [2.3.2 (2)]
- (CS-R020009) : データ交換システムパターンをサポートする場合 (CS-R020008)、[Type1]、[Type2]、[Type3]、[Type4]、[Type5]のどれかをサポートしなければならない。(必須) [2.3.2 (2)]

【サイト間】

- (CS-R020010) : データ交換システムパターンのサポートはオプションである。(オプション) [2.3.2 (2)]
- (CS-R020011) : データ交換システムパターンをサポートする場合 (CS-R020010)、[Type1]、[Type2]、[Type4]のすべてを使用できること。(必須) [2.3.2 (2)]

3. プラットフォーム通信標準のメッセージ定義仕様

3. 1 XML 定義の策定方針

本節では、XML 定義の策定方針を定める。規定する項目と標準仕様との関係は下記。

- 1) メッセージの XML 定義仕様 (XML Schema)
- 2) Web サービス I/F の XML 定義仕様 (WSDL)
- 3) ビジネスプロセスの XML 定義仕様 (WS-BPEL)

Web サービスの実行アーキテクチャと XML 定義の関係を図 3. 1 に示す。

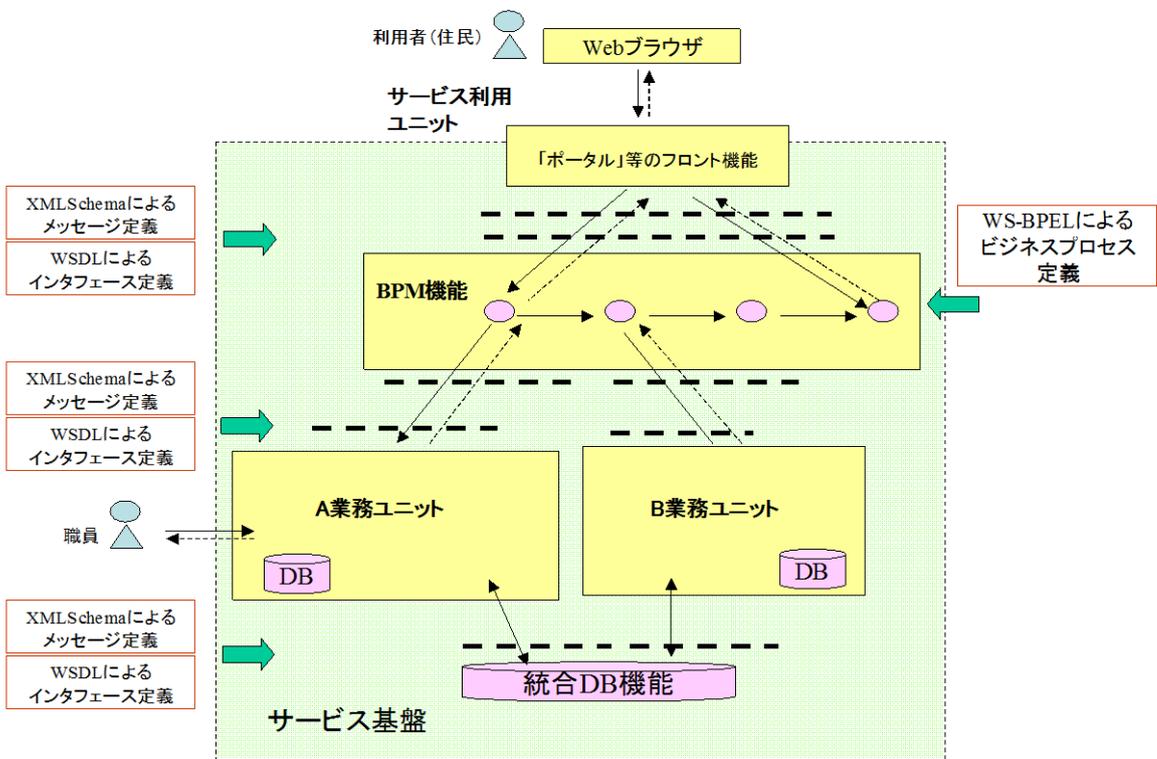


図 3. 1 地域情報 PF の実行アーキテクチャと XML 定義

それぞれのデータ定義の関係図を図3.2に示す。

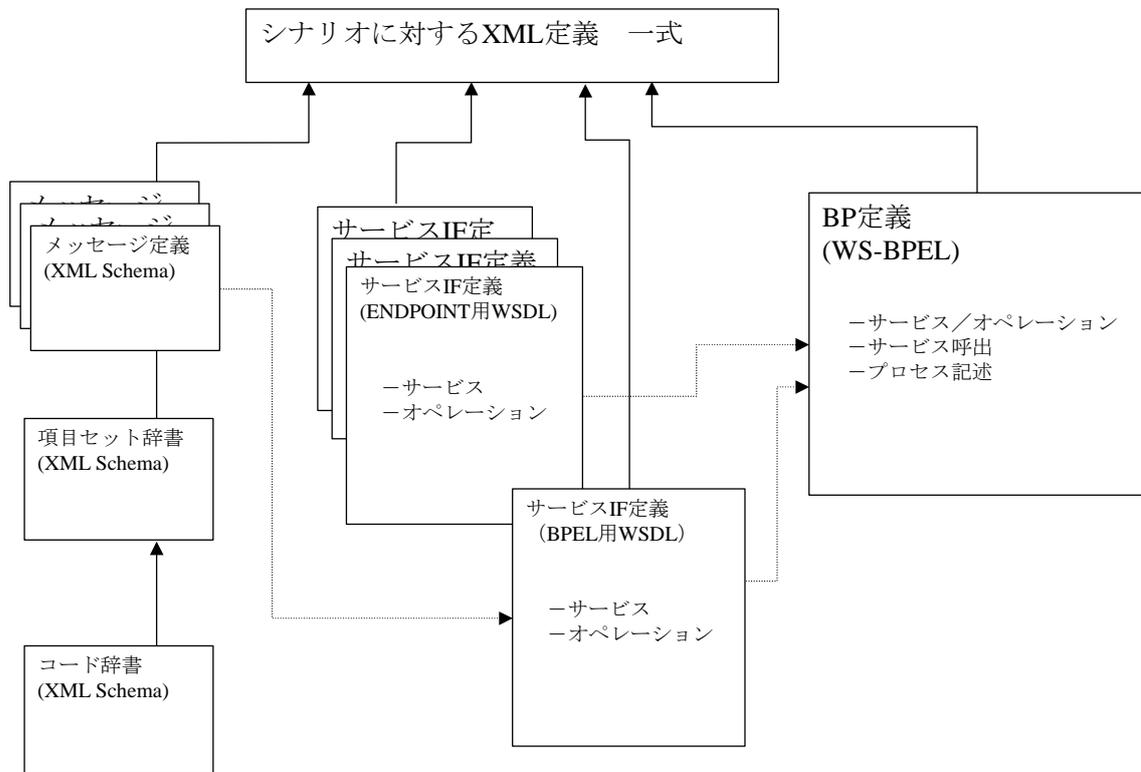


図3.2 データ定義の関連図

3. 2 XML 定義仕様

【共通】

本節では、ビジネス電文（メッセージ他）の XML スキーマのルールを規定する。

ルールの策定にあたり、「電子自治体のシステム構築のあり方に関する検討会」で報告されたルール（※1）を尊重し、国際標準の Web サービス系標準や国連の UN/CEFACT で採用されている方法（※2）も考慮するものとした。これらの参考資料をもとに、製品実装上デファクトになっている機能、互換性がなくても範囲を限定したり、制約を設けたりすることによって互換性が維持できる機能など、有効性の観点から地域情報 PF として用いるべきと判断したルールを規定している。

関連サイト：※1) http://www.soumu.go.jp/denshijiti/denshi_kentoukai.html

※2) <http://xml.coverpages.org/ni2005-01-31-a.html>

地域情報 PF 標準仕様の XML ルールは、他の外部団体が決めた標準仕様への強制力はない。他の外部団体が決めた標準仕様を取り込む際には、取り込む側の仕様において留意点を記載する。

（1）要素・属性などの命名規約

① 国外との整合性を必要としない場合

「(3)文字コード・外字に関する規則」を守る範囲で、以下で規定される、ひらがな・カタカナ・漢字を積極的に用いる。

- ・ ひらがな：JIS X0208:1997 第4区とする。（JIS X0213:2004 は、対応できなくなる実装が存在する可能性が出るため、含めない。）
- ・ かたかな：JIS X0208:1997 第5区のもののみとし、JIS X 0201 のカナ文字 (0xA0-0xD0、半角カナ) は含めない。（JIS X0213:2004 は、対応できなくなる実装が存在する可能性が出るため、含めない。）
- ・ 漢字：JIS X0208:1997 第16区から第84区のもののみとする。（JIS X0213:2004 は、対応できなくなる実装が存在する可能性が出るため、含めない。）

今後の電子自治体を推進するには、業務に関する名称をすべて英語にすることは現実的ではない（XML 日本語プロファイル（*1）参照）。ただし、システムを構築する環境によっては、上記のひらがな・カタカナ・漢字での構築が不可能な場合があるので、その場合は、英語を使用する。

② 国外との整合性を必要とする場合

W3C 勧告の Extensible Markup Language (XML) 1.0 (Third Edition) で規定する命名規則を満足する以下の文字セットで記載する。

[#x0041-#x005A] | [#x0061-#x007A] | [#x0030-#x0039] | #x002E(.) | #x002D(-) | #x005F(_) | #x003A(:)。ただし、先頭文字は、[#x0041-#x005A] | [#x0061-#x007A] | #x005F(_) | #x003A(:) に限る。

以下で規定される、記号は用いてはならない。

- ・ JIS0201 の #x0020-#x002C、#x002F、#x003B-#x0040、#x005B-#x005E、#x0060、#x007B-#x007E

(*1) XML 日本語プロファイル

http://www.y-adagio.com/public/standards/tr_xml_jpf2/toc.html

JIS X 4159 [拡張可能なマーク付け言語 (XML) 1.0] を補足する標準情報。
 文字符号化スキームとして、UTF-16 または UTF-8 を推奨している。

③ 親タグと子タグとの関係

タグ名は、その内部の構造やデータの型を表現する名称であるため、次に示すルールに基づき、タグ名の重複は禁止する。

- 同じ構造を持つ項目セットには、同じ名前を付与しても良い。異なる構造を持つ項目セットには、異なる名前を付与する。

間違った例：(構造の異なる2つの「届出人」が定義されている)

項目セット名	項目名	CD	データ型	桁数	出現回数	サンプル値
届出人	氏名		N	100	1	総務 太郎
	フリガナ		N	100	1	ソウム タロウ
届出人	氏名		N	100	1	総務 太郎
	フリガナ		N	100	1	ソウム タロウ
	英字		X	100	1	Soumu Tarou

- 同一構造が子孫に現れる場合のみ、子孫に祖先のタグ名がでて来ても良い。異なる構造に同じ名前を付与してはならない。

正しい例：(「届出人氏名」と「氏名」として異なる名前を使い分けている)

項目セット名	CD	データ型	桁数	出現回数	サンプル値
届出人情報					
届出人区分		N	20	1	本人
届出人氏名		氏名情報		1	
氏名		N	100	1	総務 太郎
フリガナ		N	100	1	ソウム タロウ
届出人電話番号		電話番号情報		1	
電話番号		X	20	1	000-123-456

間違った例：(親と子に「氏名」が現れるが、それぞれ異なる構造である)

項目セット名	CD	データ型	桁数	出現回数	サンプル値
届出人情報					
届出人区分		N	20	1	本人
氏名		氏名情報		1	
氏名		N	100	1	総務 太郎
フリガナ		N	100	1	ソウム タロウ
届出人電話番号		電話番号情報		1	
電話番号		X	20	1	000-123-456

(2) 名前空間 URI について

① 名前空間 URI 命名で使用する文字

国外との整合性を必要とする際に定義される英数字部分のみ。 平仮名・片仮名・漢字は使用しない。具体的に使える文字を下記に示す。

":" / "/" / "?" / "#" / "[" / "]" / "@" / "!" / "\$" / "&" / "'" / "(" / ")" /
"/ "*" / "+" / "," / ";" / "=" / A-Z / a-z / "-" / "." / "_" / "~" / 0-9

② 名前空間 URI 命名時の URN と URL の選択

URN を利用する。

LGWAN 他とインターネットの DNS の違い（本運用環境とテスト環境とに差が発生する恐れ有り）を考慮し、DNS の影響がない URN を採用する。ただしスキーマロケーション等、URL 表記で製品実装がされているものについては、URL を使用してもよい。

③ 名前空間 URI 命名規約

URN スキーム名としては、applic を利用する。

PF 標準で定めた URN を含む名前空間 URI の構文は次のものである。

urn:applic:xmlns:pf: {地域情報 PF 仕様種別} : {実装ファイル種別} : {バージョン文字列}

構文の中で { } に示した領域に設定する文字列のルールについて、以降の a. ~ c. に説明する。

a. 地域情報 PF 仕様種別

「地域情報 PF 仕様種別」とは、本仕様「図 0.1 地域情報 PF 仕様の体系」に含まれる仕様のうち、XML 定義がどの仕様の管理下であるかを識別するための文字列を設定する領域である。

表 3. 2. 1 に「地域情報 PF 仕様種別」を示す文字列を整理する。

表 3. 2. 1 地域情報 PF 仕様種別の文字列一覧

地域情報 PF 仕様の名前	地域情報 PF 仕様書種別の文字列
プラットフォーム通信標準仕様	common
自治体業務アプリケーションユニット標準仕様	lgxml
G I S 共通サービス標準仕様	gis
防災業務アプリケーションユニット標準仕様	dis
教育情報アプリケーションユニット標準仕様	edu
健康情報業務アプリケーションユニット標準仕様	med
準拠確認及び相互接続確認仕様(※1) (※1) 標準仕様として公開されていないが、相互接続確認イベント用ツールキットの XML 定義のために右記の文字列を確保	interop

b. 実装ファイル種別

実装ファイル種別は、[schema | wsd | bpe]の中から選択して設定する。

c. バージョン文字列

バージョン文字列は、[西暦 4 桁-修正回数 2 桁]を設定する。

バージョン文字列の更新方法については、「3. 2. (7) XML 定義のバージョンについて」にて説明する方針を踏まえる。文字列指定の基本的なルールは、西暦に作成した XML を公開する年の 4 桁を示し、修正回数にその公開される年の 1 月から数えて何回目の修正かを数字 2 桁で示す。なお、修正回数の初期値は「01」とする。

上記 a. ~c. を踏まえた名前空間 URI の指定事例を示す。

例)「プラットフォーム通信標準仕様」の一部として提供する「共通ヘッダ」・「受領 Ack」・「添付書類」の XML スキーマにかかわる名前空間 URI の場合
urn:applic:xmlns:pf:common:schema:2010-01

(3) 文字コード・外字に関する規則

本項で述べる文字セットについては、JIS X 0201 のラテン文字および JIS X0213:2004 の範囲を前提としている。

本仕様書で JIS X 0213:2004 の文字セットとは、JIS X 0213:2004 本則の名前の文字と附属書 5 表 2「数字・ラテン文字・特殊文字の代替名称」における代替名称の文字のいずれをも使用可能とする。

なお、文字に関しては、例えば住民票にする氏名は戸籍に記載の氏名とする必要があることなどによる制約や複雑な事項があるため、今後、有識者の意見を参考とし、適宜修正する。

①XML 文書の符号化

文字セット： JIS X 0213:2004 の範囲の文字を対象とする。

符号化規則： UCS (UTF-8 または UTF-16) により符号化する。

UTF-8 の Unicode signature は有り無しを問わない。

UTF-16 は BOM を必須とする。big endian も little endian も両方可とする。

②「内字」と「外字」

文字は「内字」と「外字」に分けられる。

「内字」は、単一自治体内の PF 範疇にあるすべての業務ユニットで等しく認識され、また、確実な利用（連携）が保障される文字である。特別の定義がない場合は JIS X0213:2004 の文字を「内字」と解釈する。

また、当面の経過措置として自治体内の合意に基づき、「内字」を JIS X 0208 の文字に限定することや、JIS X 0213:2004 中の第一、第二水準文字に限定することができるものとする。

「外字」は内字以外の文字である。

外字はさらに 2 つのカテゴリに分けられる。「標準外字」と「個別外字」である。

「標準外字」は「外字」であり、直接の連携は保障されないが、すべての業務ユニットでその字形（文字イメージ）について事前の合意が形成されている文字である。

つまり、直接データとして連携ができなくとも、正しい文字（字形）について相互に確認しあうことが可能である。具体的には、内字に相当する文字コードは与えられていないが、文字の一覧として PF 範疇のすべてのユニットに合意されたものがあり、その一覧上の文字を示すことで、双方で字形に対する意思疎通が図れるものをいう。

文字の一覧としては公的に整備されたものが望ましく、戸籍統一文字や法務省統一文字などを候

補として検討している。

「個別外字」は上記に含まれない文字を個別、独自に定義したもので、外部の組織との意思疎通が困難なものをさす。

③PF で利用可能な文字

【共通】

「内字」の利用（流通）を必須とする。

【サイト内】

「外字」については、PUA を用いて表現し、自治体内で十分な管理がなされる範囲で流通を許す。すべての「外字」と戸籍統一文字との対応表（マップ）を作成、管理することを推奨する。また、発展性の観点から「外字」の利用は、「標準外字」にとどめることを推奨する。

PUA : Private Use Area 外字私用領域

【サイト間】

「外字」を自治体外へ発信してはならない。

ただし、今後「外字」の利用では、「標準外字」が整備された場合、業務効率の向上等を鑑み、自治体外へ発信を容認する可能性も考慮しておく必要がある。

(4) スキーマ言語プロファイル

①スキーマ記述言語

W3C XML Schema 1.0 を採用する。

採用理由：地域情報 PF では、ビジネスプロセス (WS-BPEL)、サービスインタフェース (WSDL) の仕様を採用する。WSDL や BPEL 自身の仕様書の記述例やスキーマ定義に、W3C XML Schema が使われており、スキーマ定義として、統一のスキーマ定義言語を使用した方がよいと考える。また、製品対応の実績も考慮して採用した。

②スキーマ定義における制約条件型の表現

スキーマで表現するために必要なコストを上回るメリットが有ると確信できる場合に限る。
下記の制約を採用する。

文字列の長さ制約、数値の桁数、コードを列挙型

例、長さ 256 の文字列。文字列型を、0 以上 256 文字以下で制約する型を定義する。

③デフォルト値など XML 文書の内容を変える機能

使用しない。

デフォルト値に関しては、XML パーサの実装に依存する場合があるため。

④コード表

大規模なコード表（例：市町村コード）はスキーマ以外の手段でチェックする。

⑤ユーザ定義型

複数の項目の集まりがある意味を表し、その単位で再利用されることが多い場合、原則として名

前を付けて意図を表現する。

例、年月日型 (年、月、日、の項目の集合)

⑥属性

安易に使用せず、なるべく要素として定義する。

例、重量をあらわすタグに単位の属性を持たせる等。

⑦データ構造

内容データ構造については、mixed content と xsd:any タグを使用してはならない。

(5) XML に関する取り決め

①ルート要素の後にコメントや処理命令を書いてはならない。

②名前空間 URI として、相対 URI を使用してはならない。

③Unicode との変換のときコード位置が実装により異なる文字を直接使用してはならない。

④C1 制御文字¹は使用してはならない。また文字参照による参照も使用してはならない。

⑤名前空間を持たない要素を使用してはならない。

⑥名前空間はルート要素で必ず宣言する。ルート以外の要素で宣言してはならない。

(※1 本規定はビジネス電文自体には適用しない。ビジネス電文における名前空間の宣言については、付録 1 にいくつかの例を示す。)

⑦同一文書内で同一名前空間 URI に対応する複数の名前空間プレフィックスを利用してはならない。

⑧XML バージョン 1.0 を利用する。

XML 名前空間バージョン 1.0 を利用する。

⑨XML 宣言は指定を必須とする。

例、`<?xml version="1.0" ?>`

`<?xml version="1.0" encoding="utf-8" ?>`

⑩名前空間 URI は、URN とする。

⑪添付ファイル (本仕様 表 2. 2)、受領 Ack (本仕様 表 6. 1)、共通ヘッダ (本仕様 表 7. 1. 3) のいずれか、もしくは全てを扱う場合には、プラットフォーム通信標準仕様として提供している XML スキーマを適宜使用することとする。

⑫XML スキーマの `<xsd:documentation>` の領域に、そのスキーマを提供している地域情報 PF 仕様との関連情報やスキーマファイル自体の情報を示すこととする。この規約は、XML スキーマと標準仕様

¹ ISO 222 (JIS X 0202) の C1 領域 (#x80~#x9F) に割り当てられる制御用の文字で、ISO 6429 (JIS X 0211) に C1 集合として規定されている。Next Line (#x85)、Single Shift Two (#x8E)、Control Sequence Introducer (#x9B) 等が定義されている。

の対応関係の確認を容易にすること、および XML スキーマ自体のバージョンを目視で判別可能にすること、を目的とする。

含むべき情報については、次のとおりとする。

- ・地域情報 P F 仕様との関連情報として「PF 標準仕様」にバージョン情報を含めた標準仕様名を示す。
- ・スキーマファイル自体の情報として「管理主体」・「著作権」・「ファイル名」・「(ファイルの) 作成日」を示す。

<xsd:documentation>の領域に含める情報の書式を次の枠組の中に示すとおりとする。

```

<xsd:documentation>
  本文書は●●●に関する XML スキーマである。
  管理主体：APPLIC
  著作権：APPLIC
  ファイル名：●●●.xsd
  PF 仕様：●●●標準仕様 V●●●
  作成日：yyyy/mm/dd
</xsd:documentation>
```

上記の様な表記となるように、[改行]を入れること。

(6) データ項目の設定と XML 表現について

① データ項目の値の必須とオプションについて

ビジネスメッセージ部分のデータ項目の設定について、まず業務的な観点からそれぞれのデータ項目の値が必須となるかオプションとなるかを分析し、業務的な解釈を明確化する必要がある。業務的な解釈に基づいたデータ項目の値の必須やオプションといった属性は、XML 定義用にデータ項目の出現回数として整理される (※1)。

(※1 例えば、自治体業務アプリケーションユニット標準仕様の「業務 1-8 データ一覧」や「業務 1-9 インタフェース一覧」に示されている「出現回数」がこれに該当する)

② データ項目の値の省略について

業務的な観点での分析結果として、項目の値の設定がオプションとされたデータ項目については、XML メッセージにおいても項目の値が省略されることがありえる。

XML メッセージにおいて、項目の値が省略された場合の表現手段について、表 3. 2. 2 に整理する。それぞれに表現できる内容や利用可能な要素属性 (文字列型か数値型等) が異なる。

地域情報 PF 仕様においては、XML 定義を提供する個別の仕様 (※1) 側で用途を勘案し、表 3. 2. 2 を踏まえ適切な XML 表現を選択する。また、選択した XML 表現とその表現の解釈は個別の仕様 (※1) に記載もしくは定義する。

(※1) 個別の仕様とは、次を含む。

- ・ プラットフォーム通信標準仕様
- ・ 自治体業務アプリケーションユニット標準仕様
- ・ GIS 共通サービス標準仕様
- ・ 防災業務アプリケーションユニット標準仕様
- ・ 教育情報アプリケーションユニット標準仕様
- ・ 健康情報業務アプリケーションユニット標準仕様

なお、プラットフォーム通信標準仕様（本仕様）の「共通ヘッダ」と「受領Ack」のXMLスキーマにおいて指定するXML表現は、表3. 2. 2の「No.1 タグを省略する」である。詳細については、本仕様の「6. 6. 1 受領Ackのメッセージ形式」と「7. 1. 2 共通ヘッダの項目のXMLスキーマ作成仕様」を参照のこと。

表3. 2. 2 データ項目の値が省略された場合のXML表現方法

No.	XMLメッセージの表現方法	スキーマでの定義例	意味
1	タグを省略する	<code><xsd:element name=“要素名” minOccurs=“0” /></code>	<ul style="list-style-type: none"> ・要素の存在を含めて、データの受信側（処理側）に何も明示的に情報を伝えない。 ・左記のスキーマ定義は全てのデータ型（type）で使用可能である。
2	NULL 値表現 <code><要素名 nil=“true” ></code> <code></要素名></code>	<code><xsd:element name=“要素名” nillable=“true” /></code>	<ul style="list-style-type: none"> ・要素の値がないという状態（= NULL 値）を積極的に、データの受信側に伝えることができる。 ・左記のスキーマ定義は全てのデータ型（type）で使用可能である。 ・ただし、XML タグが必須で数値型の要素の値が省略される条件のもとでは、本表現を記述しないとエラーになる。
3	空タグ <code><要素名></要素名></code>	<code><xsd:element name=“要素名” type=“xsd:string” /></code>	<ul style="list-style-type: none"> ・要素が文字列型の場合に、長さ0の文字列であることを伝える。 ・XML タグが必須で要素の値を省略する場合、文字列型にのみ使用できる（数値型、日時型等では記述できない）。 ・下記の4と意味は同じであるが、こちらの表記が正規形であるため推奨する。
4	空タグ <code><要素名/></code>	<code><xsd:element name=“要素名” type=“xsd:string” /></code>	<ul style="list-style-type: none"> ・要素が文字列型の場合に、長さ0の文字列であることを伝える。 ・XML タグが必須で要素の値を省略する場合、文字列型にのみ使用できる（数値型、日時型等では記述できない）。 ・上記の3と意味は同じである。

(7) XML 定義のバージョンについて

①XML 定義のバージョン管理の対象

- ・本項で規定するXML 定義のバージョン管理の対象は、下記のものを含む。
名前空間 URI の構文で指定しているバージョン文字列

WSDLファイル名のバージョン文字列
XSDファイル名のバージョン文字列
BPELファイル名のバージョン文字列

・バージョン更新の際は、「名前空間 URI の構文で指定しているバージョン文字列（※1）」と各XML定義ファイルの「ファイル名のバージョン文字列」を合わせることとする。

（※1）「名前空間 URI の構文で指定しているバージョン文字列」については、本仕様の「3. 2（2）③名前空間 URI 命名規約」を参照のこと。

②XML定義のバージョン管理の方針

・バージョン更新の運用ルール（タイミング）は、名前空間 URI の「地域情報 PF 仕様種別（※2）」の単位で設定する。

（※2）「地域情報 PF 仕様種別」の詳細については、3. 2.（2）③を参照のこと。

・なお、プラットフォーム通信標準仕様の共通XML定義についてのバージョン管理の対象は、添付ファイル（本仕様2章）、受領Ack（本仕様6章）と共通ヘッダ（本仕様7章）とする。これらは単一のXMLスキーマファイル（XSDファイル）で提供されているため、それらのどの部分に仕様変更があっても修正を行い、XSDファイル名のバージョン文字列と名前空間URIのバージョン文字列と同期をとって更新する。ただし、<xsd:documentation>の領域に含まれる情報だけが変わる場合は、バージョン文字列は変更しない。

③XMLのバージョン管理対象となるバージョン文字列

本項①に示した管理対象のバージョン文字列について表現形式を規定する。

・バージョン文字列は、[西暦4桁-修正回数2桁]を設定する。

例) 2010-01

・西暦4桁は、作成したXMLを公開する年の4桁を設定する。

・修正回数2桁は、XMLを公開する年の1月から数えて何回目かを数字2桁で示す。なお、修正回数の初期値は「01」とする。

④ファイル命名規約

地域情報PF仕様として公開するWSDL、XSD、BPELファイルの命名規約を、本項で規定する。

・ファイル命名形式

{地域情報PF仕様種別の文字列} + {定義識別子} + {s} + {-バージョン文字列}. {拡張子}

・各項目を連結して表す。またバージョン文字列の前には、ハイフン(-)を付ける。

例) 共通ヘッダの場合 common-2010-01. xsd
自治体業務の場合 lgxml01s-2010-01. wsdl

・表3. 2. 3に、ファイル命名形式の構成要素の説明を記す。

表 3. 2. 3 ファイル命名形式の要素説明

No.	ファイル名構成要素	説明	
1	地域情報PF仕様種別の文字列	3. 2. (2) ③を参照	
2	定義識別子	共通定義の場合	なし
		業務標準	<p>業務標準仕様系は、ユニットドメイン識別子が決まっているものはそれを接頭辞 2 桁に含めて、業務ユニット番号を設定する。</p> <ul style="list-style-type: none"> 自治体業務アプリケーションユニット標準仕様は、業務ユニット番号のみ。 例：01 G I S 共通サービス標準仕様は、ユニット識別子(※1)+サービス番号 例：GS01 防災業務アプリケーションユニット標準仕様は、ユニット識別子(※1)+業務ユニット番号 例：AB01 教育情報アプリケーションユニット標準仕様は、ユニット識別子(※1)+業務ユニット番号 例：AK01 健康情報業務アプリケーションユニット標準仕様は、ユニット識別子(※1)+業務ユニット番号 例：AI01 <p>(※1) ユニットドメイン識別子が決まっているものはその接頭辞 2 桁を設定する。</p>
3	s “小文字”	これが追加されるのは、メッセージ定義ファイルと W S D L ファイルのみ。半角小文字。	
4	バージョン文字列	3. 2. (7) ①参照	
5	拡張子	ファイルの拡張子 (xsd, wsdl, bpel) から適切なものを指定	

3. 3 WSDL の XML 定義記述要件

(1) 仕様バージョン

- WSDL V1.1 を採用する。
- XML スキーマ定義言語：W3C XML Schema V1.0 を採用する。

(2) WSDL 記法規定

- 名称, location
 - ・ ファイル名の命名規則：(定義識別情報+バージョン).wsdl とする。
- types
 - ・ import の使用を認める。
- operation スタイル, use 属性
 - ・ document/literal とする。
- documentation

- ・ 自由に使用してよい。
- Endpoint 要件
 - ・ Port Type は、input および output を必須とする。
 - ・ input および output の名称 (name) は、PortType 内で一意でなければならない。
- WSDL:required 属性
 - ・ WSDL:required 属性は使用しない。
- binding
 - ・ wsdlsoap:operation の soapAction 属性の値は URI 形式である。
 - ・ wsdlsoap:operation の soapAction 属性の値と、HTTP リクエストメッセージの SOAPAction ヘッダフィールドの値は同じでなければならない。
 - ・ SOAPAction ヘッダフィールドの値は引用符で括った値でなければならない。
 - ・ wsdlsoap:operation の soapAction 属性の値を未設定にすることも可能だが、その場合、HTTP リクエストメッセージの SOAPAction ヘッダフィールドは、SOAPAction: "" となっていなければならない。

例：WSDL の binding 部が

```

<binding name="xxxx" type="yyyy">
  <wsdlsoap:binding style="document"
                    transport="http://schemas.xmlsoap.org/soap/http"/>
  <operation name="zzzz">
    <wsdlsoap:operation style="document" soapAction="aaaa"/>
    . . . . .
  
```

の場合、SOAPAction: "aaaa" となる。

- WSDL 定義は、WS-I Basic Profile 1.0 に準拠すること。

(3) その他

- 日本語の使用を認める。
- その他の規定は、3. 2 節に準ずる。

<<準拠ルール>>

【共通】

- (GS-R032001)： PF 通信機能で交換されるメッセージである XML インスタンスは、プラットフォーム通信標準のメッセージ定義仕様における XML 定義仕様を満たすこと。(必須) [3. 2]
- (GS-R032002)： 地域情報 PF 標準として定義される XML スキーマは、プラットフォーム通信標準のメッセージ定義仕様における XML 定義仕様を満たすこと。(必須) [3. 2]
- (GS-R032003)： PF 通信機能向けのサービスインタフェース定義の WSDL 定義は、プラットフォーム通信標準のメッセージ定義仕様における WSDL の XML 定義要件を満たすこと。(必須) [3. 3]

4. プラットフォーム通信標準のビジネスプロセス制御定義仕様

4. 1 定義

本節では、地域情報 PF 仕様におけるビジネスプロセスに関連した定義を行う。地域情報 PF 仕様では、各用語を以下の様に定義する。

- ・ **ビジネスプロセス管理機能** :
経営目標を実現する業務、ビジネスプロセスを IT の支援を受けて短期間・適正に設計し、その改善サイクルを継続的に進める経営手法のこと。
- ・ **ビジネスプロセス (BP:Business Process)** :
業務のこと。ただし地域情報 PF 仕様では、より明確に「1 つ以上のサービス実行を含み、一連の動作として定義される処理」と定義する。概ねの場合、サービスの高付加価値化を目的として定義される。
- ・ **ビジネスプロセスインスタンス (BP インスタンス)** :
具体的に実行中のビジネスプロセスのこと。
- ・ **ビジネスプロセス管理機能 (BPM 機能:Business Process Management)** :
BP として定義された処理内容を、要求に応じて実行可能なインスタンスとして生成し、生成された複数のプロセスインスタンスを並列に実行管理するソフトウェア基盤機能である。ただし、こうした機能を実現するプログラムの総称の意として「BPM」、もしくは「BPM 機能」という単語を用いることもある。
- ・ **WS-BPEL (Web Services Business Process Execution Language)** :
BP に関する定義、実行を行うための XML (Extensible Markup Language) によるスクリプト言語で、OASIS (Organization for the Advancement of Structured Information Standards) により国際標準として勧告されている。地域情報プラットフォーム標準仕様では、WS-BPEL を標準として採用する。
- ・ **WS-BPEL プロセス** :
WS-BPEL で記載された記述が起動し、実行状態のプロセスとして定義されるもの。

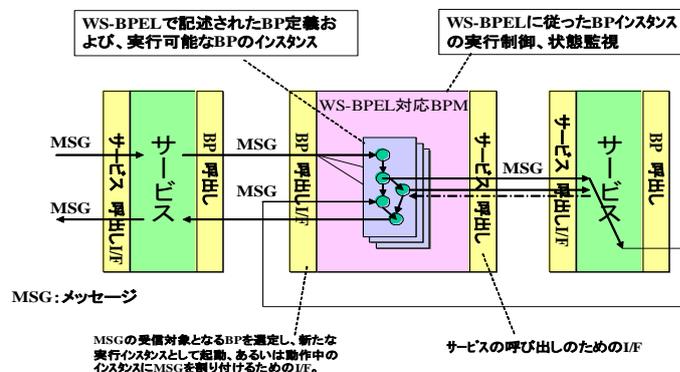


図 4. 1 ビジネスプロセス管理機能、BP、WS-BPEL の関係

4. 2 プロセス制御手段に関する利用規定

4. 2. 1 概要

地域情報 PF 仕様では、ビジネスプロセス (BP) は、WS-BPEL の一部を用いて定義される。このため WS-BPEL に依存して、適用されるビジネスプロセスの制御手段が存在する。以下の制御手段は、自治体の内外を問わず、地域情報 PF 仕様では利用できることを必須とする。

- ・ ①ビジネスプロセス呼び出し口の識別 :
- ・ ②コリレーションセットによる BP インスタンスの識別 :
- ・ ③BP のフロー制御のための構造化アクティビティ :

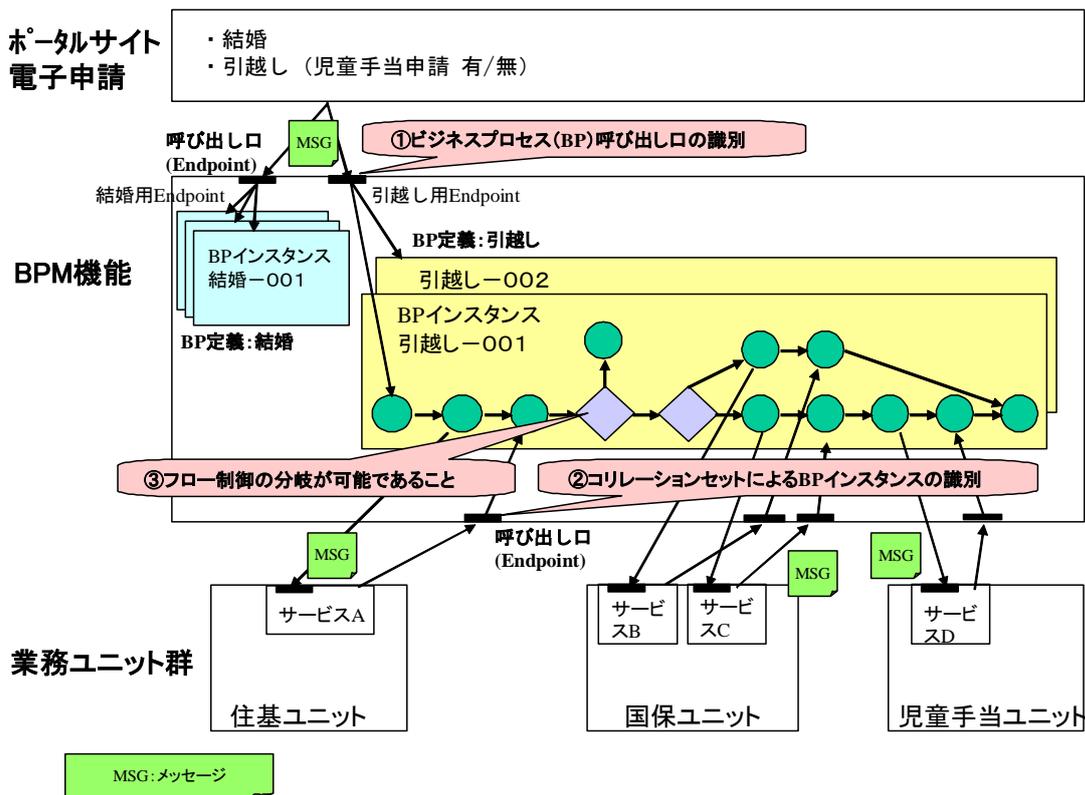


図 4.2 利用を前提とするプロセス制御手段

4. 2. 2 ビジネスプロセス (BP) 呼び出し口の識別に関する規定

この要件は、ビジネスプロセスの開始やビジネスプロセスに対する応答のための BPM 呼び出し口であるエンドポイント (Endpoint) を、サービスの呼び出し元から識別することにある。その仕様を定義すると次のようになる。

- ・ ビジネスプロセスの開始や、ビジネスプロセスへの応答を行うための BPM 呼び出し口として、BP 定義とアクティビティ (receive 等メッセージ受け口) の組単位に、個別の Endpoint を用意しなければならない。
- ・ Endpoint 呼出しの際、呼出し元も、どの BP 定義のどの呼び出し口に処理結果を戻すかを、意識して明示的に指定しなければならない。

4. 2. 3 コリレーションセットによる BP インスタンスの識別に関する規定

【共通】

この要件は、BPMにおいて受信メッセージを、BPM機能上の適切なBPのインスタンスへ割り付けることである。このため受信メッセージに対し、割り付けるべきBPインスタンスを特定するためのデータ項目を、コリレーションセットとして定義する必要がある。

コリレーションセットを定義することで、BPインスタンスやBPから呼び出される業務ユニットが送受信するメッセージ上でそのデータ項目と値を引き継ぐことにより、一連のメッセージを同一のBPインスタンスで処理できるようにする。ただし、コリレーションセットを有効に機能させるためには、以下の2事項が必須の条件となる。

- ・BP定義作成時にコリレーションセットとなるデータ項目を決定する。
- ・呼び出し側のポータルサイトや電子申請が、実行時にBP開始のための送信メッセージ上のコリレーションセットとなる項目にユニークな値を設定する。

これを受けて、コリレーションセットに関する仕様として、以下を定義する。

- ・7. で定義されるメッセージの共通ヘッダ部分の「共通コリレーションセット」項目を、すべての処理で共有し、使用することを推奨する。「共通コリレーションセット」項目を使用する場合には、サービス呼出側で、実行時にユニークな値を設定しなければならない。BPM機能から他のビジネスプロセスを呼び出す場合にも遵守すべき事項である。
- ・メッセージの共通ヘッダ部分の「共通コリレーションセット」項目の代わりに、業務的にユニークとなる業務項目を設定し、それをビジネスプロセス定義上のコリレーションセットとすることは否定しない。ただしその場合は、WS-BPELの中で定義する必要がある。

4. 2. 4 BP フロー制御のための構造化アクティビティに関する規定

【共通】

この要件は、ビジネスプロセスから呼び出した業務ユニット（サービス）の処理結果に応じて、ビジネスプロセスの実行フローを制御できるように、BPM機能内で分岐のための構造化アクティビティが利用できることである。4.3で定義される様に、業務ユニットの処理結果に応じて共通ヘッダの一部に関連情報が指定される。ビジネスプロセスの定義では、これに応じて、条件分岐などのフロー制御が必要となる。具体的な仕様は、4.4節にて規定する。

4. 3 ビジネスプロセス (BP) のフロー制御に関する規定

4. 3. 1 概要

地域情報PF仕様では、ビジネスプロセスに関するフロー制御に関して独自の仕様事項が存在する。ビジネスプロセスは、WS-BPELで記述されるが、実際の自治体業務では条件に応じた複数の処理フローのバリエーションが必要となる。

ビジネスプロセスを記述するWS-BPELは、確定的にビジネスプロセスを定義するため、通常は標準的な処理フローしか定義・実施できない。この結果、動的なフロー制御を考慮しないWS-BPELの定義だけでは今後のフローに関するバリエーションすべてを満足できるだけの、木目の細かい所望の処理フローの実現ができない。

そこでビジネスプロセスは、すべての考えられ得るフローに関するバリエーションを網羅的に取り込

んだ共通なフローを定義・設計し、実際にビジネスプロセスのインスタンスが起動する BPM 機能呼び出す際に、仔細、かつ利用する範囲のフローの内容を指定することで、所望のフローを実現する仕組みを新たに定義する。

ここで定義する仕様は、ポータルサイトと自治体、民間企業との間や自治体間等の複数組織体にわたるビジネスプロセスにおける必要性が大きいことから、複数の自治体に跨るビジネスプロセスでは、この仕組みの適用を必須事項とする。なお、単一自治体内においても同様のことが想定されるとともに、他の自治体との連携を行う際に必要となることから自治体内での適用も推奨する。

ここで定義する仕様は次の 2 つである。

- ・ ①ビジネスプロセス実施の依頼元による BP のフロー制御 :
- ・ ②サービス処理結果による BP のフロー制御 :

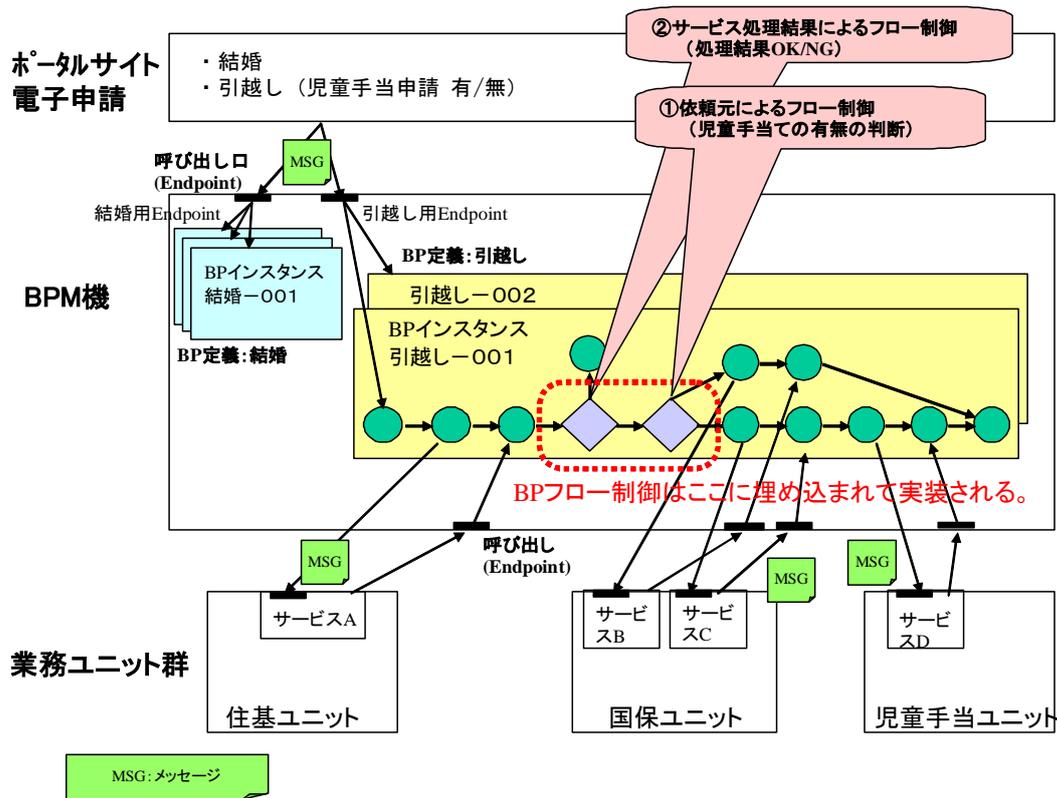


図 4.3 アーキテクチャ上のビジネスプロセスフロー制御の位置づけ

4. 3. 2 ビジネスプロセス (BP) 実施の依頼元による BP のフロー制御の仕様事項

【共通】

この要件は、BPM 機能においてビジネスプロセス実施の依頼元であるポータルサイト、電子申請等からの受信メッセージ、申請データにより、ビジネスプロセスの実行フローを制御できることである。具体的な仕様事項としては、以下が定義される。

- ・ 申請内容に応じて処理を制御・変更するために、7. で定義されるメッセージの共通ヘッダ部分の「ビジネスプロセス制御情報」が定義されている。依頼元は制御の内容を指定するため、適切な値をセットする。一方、ビジネスプロセスを定義する WS-BPEL 上では、その値に応じて条件分岐などのフロー制御するように記述しなければならない。なお、このためには、4.2 の BP フロー制御のための構造化アクティビティに関する仕様要件を満足しなければならない。

- ・ 上記以外の方法としては、本文などの共通ヘッダ以外の業務メッセージの値で、直接、フロー制御条件を読み取り制御する方法も存在するが、複雑化を招くため推奨しない。

4. 3. 3 サービス処理結果によるビジネスプロセス (BP) のフロー制御の仕様事項

【共通】

この要件は、BPM 機能から呼び出した業務ユニット (サービス) の処理結果に応じてビジネスプロセスの実行フローを制御できることである。具体的な仕様事項としては、以下が定義される。

- ・ 業務ユニットの処理結果に応じて処理を制御・変更するために、7. で定義されるメッセージの共通ヘッダ部分に「業務サービス結果情報」が定義される。業務ユニットは、「システムエラー報告」とは別に、制御のために適切な値をセットする。このため、ビジネスプロセスを定義する WS-BPEL 上では、その値に応じて条件分岐などのフロー制御するように記述しなければならない。なお、この為には、4.2 の BP フロー制御のための構造化アクティビティに関する規定を満足してなければならない。

4. 4 WS-BPEL の適用範囲、制約に関する規定

4. 4. 1 概要

【共通】

現在、WS-BPEL 仕様では、異なる BPM 製品間において、そのポータビリティまでは保証してはいない。このため、任意ビジネスプロセスの実行に関する相互運用性を確保するためには、WS-BPEL 上で利用できるアクティビティセットについて制約を設ける必要がある。この制約の選定基準は、任意ビジネスプロセスの定義に必須、かつ十分と考えられ、さらに BPM 製品への実装依存性が限定的と推定されるものに限られる。本節では、相互運用性の観点から定義される WS-BPEL の制約について定義する。なお、規定は6カテゴリに分類される。

- ・ WS-BPEL の仕様対応バージョン
- ・ サービス呼出し I/F に関する規定 (基本アクティビティ)
- ・ 実行制御に関する規定 (構造化アクティビティ)
- ・ 変数処理に関する規定
- ・ ハンドラに関する規定
- ・ WSDL と WS-BPEL 間の挙動、並びに制御との関係

4. 4. 2 WS-BPELWS-BPEL 仕様の対応バージョン

【共通】

- ・ OASIS WS-BPEL V2.0 を採用する。
- ・ XML Schema による WS-BPEL 言語の定義は OASIS から公開されているものを用いる。

4. 4. 3 サービス呼出し I/F に関する規定（基本アクティビティ）

【共通】

以下の基本アクティビティのみ利用可とする。

- ・ receive
- ・ reply
- ・ invoke

4. 4. 4 実行制御に関する規定（構造化アクティビティ）

【共通】

以下の構造化アクティビティのみ利用可とする。

- ・ sequence
- ・ flow
- ・ if
- ・ while
- ・ throw

【制約条件】 flow について link なしを制約条件とする。

4. 4. 5 変数処理に関する規定

【共通】

- ・ assign を利用する。

【制約条件】

- ・ XML のすべてのデータを割り当てる。
- ・ XML の部分データを変数に割り当てる場合は Xpath 1.0 を利用する。
バリデーションを使用しても良い。

4. 4. 6 ハンドラに関する規定

【共通】

- ・ fault を使用してエラーハンドリングを行う。

4. 4. 7 例外規定

【共通】

地域情報 PF 仕様の準拠性、相互接続性を必要としないシステムにおいては、本節で規定する仕様以外の仕様に基づいてシステム構築を行うことを阻むものではない。これについては、ガイドラインに留意点を記すので、調達者の責任で判断する。

4. 5 メッセージ交換パターンに対する WS-BPEL 実装規定

4. 5. 1 地域情報 PF 仕様における推奨メッセージ交換パターン

6. で定義されるように地域情報 PF 仕様では、下記 3 つのメッセージ交換パターンを採用する。一般にメッセージ交換パターンは、メッセージ送受信に対する業務要件に依存するため、業務設計時に指定される。ここでは、指定されたメッセージ交換パターンを WS-BPEL にて実装する際の規定事項を定義する。

- ・ リクエスト型受領 Ack あり
- ・ リクエスト・レスポンス型同期型レスポンス
- ・ リクエスト・レスポンス型同期型受領 Ack+非同期型レスポンス

4. 5. 2 定義、一般事項

本節での用語は以下の様に定義される。この定義は、6. のものと同じである。

4. 5. 2. 1 基本用語定義

- ・ メッセージ交換パターン (MEP : Message Exchange Patterns) : 後述するメッセージ群の交換の類型を定義するものである。
- ・ MEP 処理系 : BPM 処理系、WS-BPEL 処理系等の狭義の処理系のこと。
- ・ MEP 実行系 : MEP を処理するソフトウェアの総称のこと。前述 MEP 処理系が中心となるが、MEP 処理系を利用する業務ユニットレベルのものも含め、広義に定義する。
- ・ エンティティ : OSI (Open Systems Interconnection) の定義に基づき、メッセージを扱うものをエンティティと呼ぶ。具体的には、サービス、プロセスを意味する。

4. 5. 2. 2 メッセージの種類に関する定義

- ・ 要求メッセージ、リクエスト (req) : 業務依頼を意味する要求メッセージであり、地域情報プラットフォーム標準仕様における MEP 開始の送信メッセージ。
- ・ 応答メッセージ、レスポンス (res) : リクエストに対する業務処理結果を返す際のメッセージ。
- ・ 受領 Ack (Acknowledgement) (ack) : 業務プロセス、もしくは業務サービスでリクエスト、レスポンスを受信したことを送信元に通知するためのメッセージ。この受領 Ack は、デフォルトとして「受信直後の業務ユニットレベルの処理前」に送付されるものとする。これは、TCP/IP、高信頼性通信などの受領確認とは異なり、通常の業務メッセージとして表現される受領確認である。

4. 5. 2. 3 メッセージを扱うエンティティの定義

エンティティは役割に応じて以下の様に定義する。

- ・ 開始側 (initiator) : 要求メッセージを送信するサービス、あるいは BPM プロセスのことで、この仕様では、「発呼側」と呼ぶこともある。
- ・ 応答側 (responder) : 要求メッセージを受信し、必要に応じてレスポンスを送信するサービス、BPM 機能のプロセス、あるいは、統合 DB 機能のことで、この仕様では「着呼側」と呼ぶこともある。
- ・ メッセージ送信側 : 特に SOAP 通信レベル (高信頼性通信の場合は、そのレベル) から下位レベルでメッセージを送信する側のことで、上記「開始側」とは異なる。(「応答側」も「メッセージ送信側」)

に成り得る。) なお、高信頼性通信機能では、単に「送信側」と呼ぶこともある。

- ・ メッセージ受信側：特に SOAP 通信レベル（高信頼性通信の場合は、そのレベル）から下位のレベルでメッセージを受信する側のことで、上記「応答側」とは異なる。（「開始側」も「メッセージ受信側」に成り得る。） なお、高信頼性通信機能では、単に「受信側」と呼ぶこともある。

4. 5. 2. 4 表記法説明

各メッセージ交換パターンを概説する上で、表記法としては、一般的に利用される UML (Unified Modeling Language) のシーケンス図を用いる。ただし、表記に当たっては下記制約、特記事項を設ける。

- ・ シーケンス図の長方形は、通常の意味では、通信における 1 つのセッション境界を示す。ただし、同じ形式でトランザクション境界や、オブジェクトの生存期間を意味する場合もある。
- ・ 上記を受けて、2 つの長方形を続けて記す場合は、非同期を意味する。
- ・ シーケンス図中の開始側、応答側は、いずれか一方、もしくは両方が WS-BPEL のプロセスとして実現される。一方のみが WS-BPEL のプロセスである場合、他方はポータルサイト、業務ユニットなど何らかのサービス実装となる。
- ・ シーケンス図中の注釈は、WS-BPEL アクティビティを意味し、WS-BPEL プロセスとして実現された場合の各アクティビティを意味する。

4. 5. 3 WS-BPEL における同期型・非同期型

メッセージ交換パターンにおいては同期型・非同期型が定義される。WS-BPEL での実装においては、以下の様に対応付けられる。

(1) 同期型

一般には、受信内容に対する返信（受領 Ack、レスポンス）が、受信時と同一の HTTP (Hypertext Transfer Protocol) セッションで返される通信パターンを意味する。WS-BPEL を用いた同期型メッセージ交換は、送信側が `invoke`、受信側が `receive(pick)`、`reply` のアクティビティによりで実現される。

(2) 非同期型

一般には、受信内容に対する返信（受領 Ack、レスポンス）が、受信時と異なる別の HTTP セッションで返される通信パターンを意味する。WS-BPEL を用いた非同期型メッセージ交換は、相互に `invoke`、`receive` のアクティビティを繰り返すことにより実現される。

4. 5. 4 WS-BPEL 実装における共通規定

すべてのメッセージ交換パターンで、受信側にあたる WS-BPEL では、必須のレスポンス、もしくは受領 Ack を受信した際に、受信したメッセージに含まれる共通ヘッダ部のシステムエラー報告の値がエラー（障害）か、否かを検知しなければならない。エラーが含まれている場合は、後述の規定で処理する。

4. 5. 5 「リクエスト型受領 Ack あり」実装の WS-BPEL 実装規定

掲題メッセージ交換パターンでは、開始側がリクエスト送信の HTTP セッションの戻りで受領 Ack を受信することになる。開始側が WS-BPEL プロセスである場合は、`invoke` アクティビティにより記述する。一方、応答側が WS-BPEL プロセスである場合は、`receive` アクティビティ と `reply` アクティビティ の

組み合わせで記述する。(表 4. 1 を参照)

表 4. 1 「リクエスト型受領 Ack あり」実装時の WS-BPEL 実装規定

	WS-BPEL での実装パターン、実装規定
開始側	invoke (リクエスト送信、受領 Ack 受信) ⇒ 受領 Ack チェック
応答側	receive (リクエスト受信) ⇒ 受領 Ack の生成 ⇒ reply (受領 Ack 返信)

4. 5. 6 「リクエスト・レスポンス型同期型レスポンス」実装の WS-BPEL 実装規定

【共通】

掲題メッセージ交換パターンでは、開始側はレスポンスをリクエストの送信と同一の HTTP セッションで受信する。開始側が WS-BPEL プロセスである場合、invoke アクティビティを用いて記述する。一方、応答側が WS-BPEL プロセスである場合は、receive アクティビティと reply アクティビティの組み合わせで記述する。(表 4. 2 を参照)

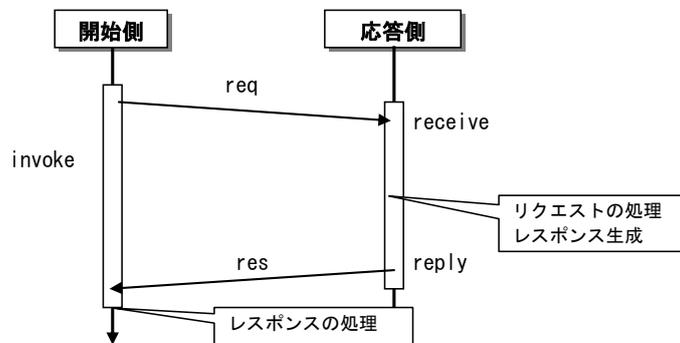


図 4. 4 リクエスト・レスポンス型同期型レスポンス

表 4. 2 「リクエスト・レスポンス型同期型レスポンス」実装時の WS-BPEL 実装規定

	WS-BPEL での実装パターン、実装規定
開始側	invoke (リクエスト送信、レスポンス受信) ⇒ レスポンス受信後の処理
応答側	receive (リクエスト受信) ⇒ リクエストに基づく処理、レスポンス生成 ⇒ reply (レスポンス返信)

このメッセージ交換パターンは、リクエストに対する応答側でのレスポンス返信開始までの処理が短時間（目安として数十秒以内）で終了する場合の利用を前提とする。開始側の invoke アクティビティでレスポンスの受信タイムアウト監視を設定することも可能であるが、一部、実装に依存する。そのため、SOAP 通信レベル（高信頼性通信の場合は、そのレベル）以下のレイヤで、HTTP レベルのセッションタイムアウト検出を行うことにより、代替のタイムアウト監視を行うことが可能となる。（invoke アクティビティにおける受信タイムアウト監視が、実装に依存する場合、その利用はポータビリティの点から推奨しない。）

4. 5. 7 「リクエスト・レスポンス型同期型受領 Ack+非同期型レスポンス」実装の WS-BPEL 実装規定

【共通】

掲題メッセージ交換パターンでは開始側、応答側共に、リクエストに対する受領 Ack、レスポンスに対する受領 Ack をそれぞれ同一 HTTP セッションで同期型に返信する。さらにリクエストとレスポンスは別の HTTP セッションで非同期型に送受信する。これは、応答側にてリクエスト受付からレスポンス返信までに時間がかかる場合に採用すべきメッセージ交換パターンである。

このメッセージ交換パターンを WS-BPEL で実現する場合、開始側、応答側共に invoke アクティビティ、receive アクティビティ、reply アクティビティの組み合わせで記述される。開始側、応答側のどちらかをポータルサイト、業務ユニットの機能として実現する場合、その内部で非同期に実施されるリクエストの処理とレスポンスの処理を連携させる必要がある。

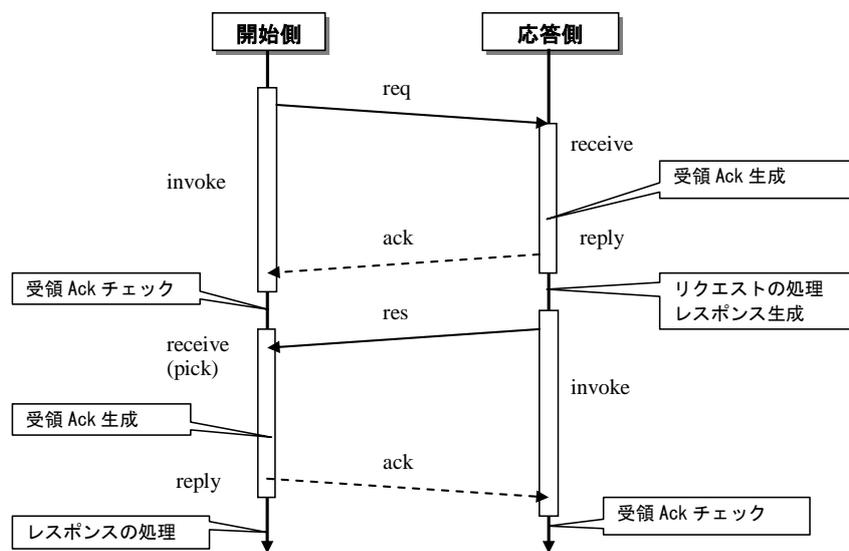


図 4. 5 リクエスト・レスポンス型同期型受領 Ack+非同期型レスポンス

表 4. 3 「リクエスト・レスポンス型同期型受領 Ack+非同期型レスポンス」実装時の WS-BPEL 実装規定

WS-BPEL での実装パターン、実装規定	
開始側	invoke (リクエスト送信、受領 Ack 受信) ⇒ 受領 Ack チェック ⇒ receive(pick) (レスポンス受信) ⇒ 受領 Ack 生成 ⇒ reply (受領 Ack 返信) ⇒ レスポンス受信後の処理
応答側	receive (リクエスト受信) ⇒ 受領 Ack 生成 ⇒ reply (受領 Ack 返信) ⇒ リクエストに基づく処理、レスポンス生成 ⇒ invoke (レスポンス返信、受領 Ack 受信) ⇒ 受領 Ack チェック

4. 5. 8 SOAP レベルや高信頼性通信レベルにおける障害検知をした場合の WS-BPEL 実装規定

【共通】

障害は invoke アクティビティ、reply アクティビティなどにおける SOAP 処理系（高信頼性通信の場合は、その処理系）の障害という形で検知される。従って、呼出元の BPM 機能である WS-BPEL 処理系により fault が throw されるので、invoke アクティビティ、reply アクティビティに対応した fault handler で異常系処理を記述する。6. に定義する様に異常系処理には、補償・回復処理等を組み込んではならない。

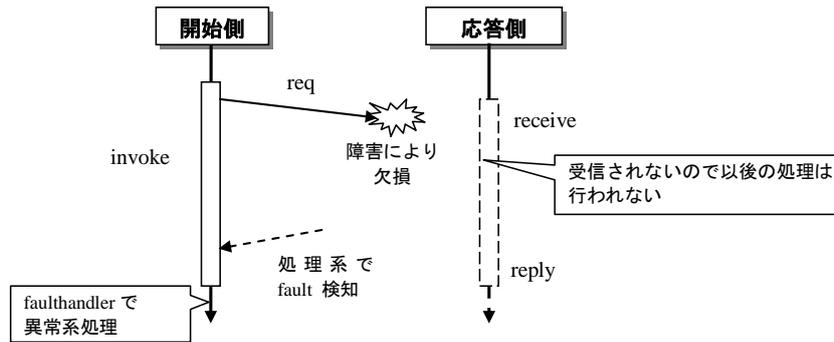


図 4. 6 invoke による SOAP レベル（高信頼性通信の場合は、そのレベル）障害検知

4. 5. 9 受信タイムアウトを検出した場合の実装規定

【共通】

受信タイムアウトとは、開始側で検出する場合の障害で、同期型と非同期型で対応が異なる。同期型のメッセージ受信がタイムアウトする場合は、4. 5. 8 と同じ方法で対応する。これに対して、非同期型の場合、受信側で WS-BPEL の receive を使用すると、WS-BPEL 処理系でタイムアウトを検知することはできない。従って、WS-BPEL が受信側となった場合の非同期型の受信タイムアウトについては WS-BPEL に対する実装を規定せず、呼出元の業務ユニットでタイムアウト監視を行うことを推奨する。なお、関連事項をガイドラインに記載する。

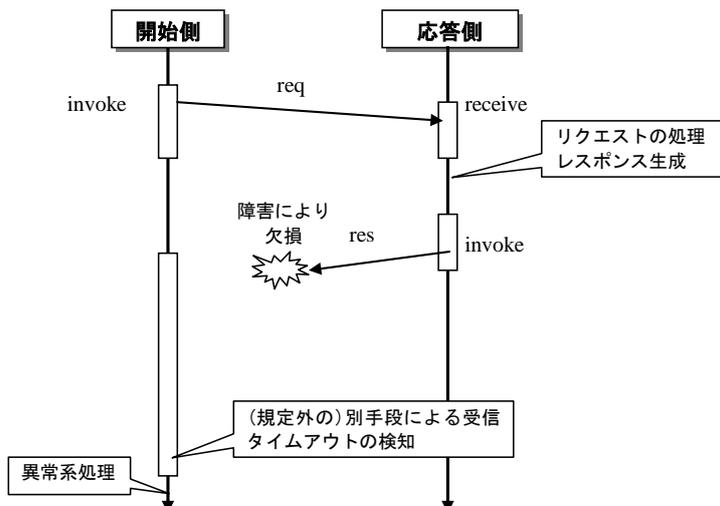


図 4. 7 非同期メッセージ受信に対するタイムアウト検知

4. 5. 10 エラーを示すレスポンス、もしくは受領 Ack を受信した場合の実装規定

【共通】

4. 5. 4にて定義しているように、受信側にあたる WS-BPEL では、必須のレスポンス、もしくは受領 Ack を受信した際に、受信したメッセージに含まれる共通ヘッダ部のシステムエラー報告の値がエラー（障害）か、否かを検知しなければならない。エラーを意味する値の場合、もしくは XML Schema 上の違反により該当情報が含まれていない場合、これをエラーとして扱い、fault を throw して、アクティビティに対応した fault handler で異常系処理を記述する。6. に定義する様に異常系処理には、補償・回復処理等を組み込んではならない。

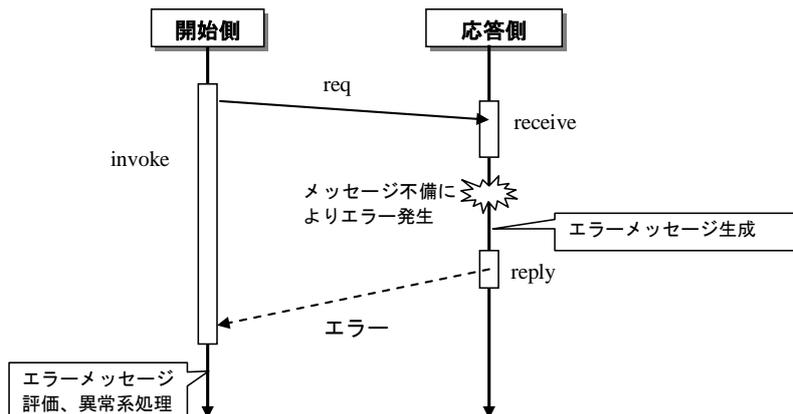


図 4. 8 エラーメッセージによる障害検知

4. 6 追加規定事項

4. 6. 1 WS-BPEL 仕様の採用するバージョン

【共通】

これは、4. 4の記述に準ずる。

4. 6. 2 WS-BPEL 用の WSDL 記述に関する規定事項

【共通】

WSDL1.1 について、WS-BPEL との関係に関しては、下記規定を遵守しなければならない。

- ・ メッセージ受信時に起動すべき WS-BPEL の決定を一意に行うため、受信側の WS-BPEL 内で定義される partnerlink は、各々排他的、かつ固有の受信 Endpoint を指定しなければならない。そして WS-BPEL を含めて送信側で、当該指定のものを一意に指定する必要がある。
- ・ Endpoint 要件として、WSDL 記述上の Port Type は、input および output を必須とする。
- ・ 外部に対して呼び出しインタフェースとして定義・公開される WSDL 記述と、WS-BPEL に関する定義に使用する WSDL の記述とは分離して定義しなければならない。
- ・ WS-BPEL に関する定義に使用する WSDL の記述は、他のものと同様に、WS-I Basic Profile 1.0 の規定に従わなければならない。WS-I Basic Profile 1.0 の Errata では WSDL 拡張要素が WSDL 文書中のどこにでも置けるようになっているが、地域情報 PF 仕様では WS-BPEL 関係の定義を以下のように<wsdl:documentation>の後(<wsdl:Types>の前)に記述することとする。

```
<wsdl:definitions ...>
  <wsdl:documentation ...> ... </wsdl:documentation>
```

<!--ここに WSDL 拡張要素として WS-BPEL 関係の定義を記述する。 -->

```
<wsdl:types ...> ... </wsdl:types>
<wsdl:message ...> ... </wsdl:message>
<wsdl:portType ...> ... </wsdl:portType>
<wsdl:binding ...> ... </wsdl:binding>
<wsdl:service ...> ... </wsdl:service>
</wsdl:definitions ...>
```

<<準拠ルール>>

【共通】

- (CS-R040001) : BPM 機能は、OASIS WS-BPEL V2.0 に準拠しなければならない。(必須) [4. 4. 2]
- (CS-R040002) : WS-BPEL 定義に使用する WSDL は、WS-I Basic Profile に準拠しなければならない。
(必須) [4. 6. 2]
- (CS-R040003) : BPM 機能は、サービス呼出 I/F に関する receive、reply、invoke を使用する。
(必須) [4. 4. 3]
- (CS-R040004) : BPM 機能は、実行制御に関する sequence、flow、if、while を使用する。(必須) [4. 4. 4]
- (CS-R040005) : BPM 機能は、throw を使用する。(必須) [4. 4. 4]
- (CS-R040006) : BPM 機能は、変数処理に関する assign を使用する。(必須) [4. 4. 5]
- (CS-R040007) : BPM 機能は、fault handler を使用する。(必須) [4. 4. 6]
- (CS-R040008) : BPM 機能は、コリレーションセットを使用する。(必須) [4. 2. 3]
- (CS-R040009) : BPM 機能は、assign では XML のすべてのデータを割り当てることができなければならない。
(必須) [4. 4. 5]
- (CS-R040010) : BPM 機能は、XML の部分データを変数に割り当てるために Xpath 1.0 を使用する。
(必須) [4. 4. 5]
- (CS-R040011) : BPM 機能は、flow は、link なしでもよい。(オプション) [4. 4. 4]
- (CS-R040012) : BPM 機能は、Xpath に関するバリデーションを使用してもよい。(オプション) [4. 4. 5]
- (CS-R040013) : BPM 機能は、BP 実施の依頼元からの指定で BP 実行フローを制御できなければならない。
(必須) [4. 3. 2]
- (CS-R040014) : BPM 機能は、呼出先業務ユニットの処理結果で BP 実行フローを制御できなければならない。
(必須) [4. 3. 3]
- (CS-R040015) : BPM 機能は、応答メッセージ受信時にメッセージ内の共通ヘッダ部のシステムエラー報告をチェックしなければならない。(必須) [4. 5. 4]
- (CS-R040016) : BPM 機能では、例外規定以外は fault handler の異常系処理として、デフォルトの補償・回復処理等を起動してはならない。(必須) [4. 5. 9]
- (CS-R040017) : BPM 機能は、BP 開始や応答のための Endpoint を呼出元から一意に識別しなければならない。(必須) [4. 6. 2]
- (CS-R040018) : BPM 機能は、partner link により一意に Endpoint を指定しなければならない。
(必須) [4. 6. 2]

用語の定義

項番	用語	意味
1	ビジネスプロセスマネージメント	経営目標を実現する業務、ビジネスプロセスを IT の支援を受けて短期間・適正に設計し、その改善サイクルを継続的に進める経営手法のこと。
2	ビジネスプロセス (BP)	業務のこと。ただし地域情報 PF 仕様では、より明確に「1 つ以上のサービス実行を含み、一連の動作として定義される処理」と定義する。概ねの場合、サービスの高付加価値化を目的として定義される。
3	ビジネスプロセスインスタンス (BP インスタンス)	具体的に実行中のビジネスプロセスのこと。
4	ビジネスプロセス管理機能 (BPM 機能)	BP として定義された処理内容を、要求に応じて実行可能なインスタンスとして生成し、生成された複数のプロセスインスタンスを並列に実行管理するソフトウェア基盤、ソフトウェア基盤機能である。
5	WS-BPEL	BP に関する定義、実行を行うための XML によるスクリプト言語で、OASIS により国際標準として勧告されている。
6	WS-BPEL プロセス	WS-BPEL で記載された記述が起動し、実行状態のプロセスとして定義されるもの。
7	コリレーションセット	BPM 機能が、受信メッセージに対し、割り付けるべき BP インスタンスを特定するためのデータ項目。
8	メッセージ交換パターン (MEP : Message Exchange Patterns)	メッセージ群の交換の種類を定義するもの。
9	受信タイムアウト	規定時間内に開始側においてレスポンス (res) メッセージを受け取れないこと。
10	補償・回復処理	異常が発生した場合に、既に行われた処理をキャンセルする等を行い、一連の処理が行われる前の状態に戻す処理。
11	障害検知	異常を捕捉すること。
12	即時対応処理	障害検知を受け、それを通知する等の問題解決を伴わない対応処理。
13	異常系処理	障害検知、即時対応処理を受けて、障害を解決するための一連の処理。
14	同期	受信内容に対する返信 (受領 Ack、レスポンス) が、受信時と同一の HTTP セッションで返される通信パターン。
15	非同期	受信内容に対する返信 (受領 Ack、レスポンス) が、受信時と異なる別の HTTP セッションで返される通信パターン。

5. プラットフォーム通信標準における認証・認可・セキュリティ機能

5. 1 対象領域

本仕様では、異なるサイト間（自治体間、自治体・民間間等）に関連する、プラットフォーム通信標準仕様における、認証・認可・セキュリティ仕様について記述する。

- (1) 異なるサイト間におけるセキュリティ上の脅威と対策技術
- (2) 異なるサイト間における認証・認可・セキュリティ仕様

本仕様で規定する PF 認証・認可・セキュリティ機能の中で、サイト内とサイト間で共通に使用できる仕様に関しては、サイト内でも使用してよい。

5. 2 異なるサイト間におけるセキュリティ上の課題と対策技術

地域情報 PF に関連するセキュリティ上の課題と対象技術について下記に記述する。異なるサイト間（自治体間、自治体・民間間等）におけるセキュリティ上の課題を図 5.1 に示す。

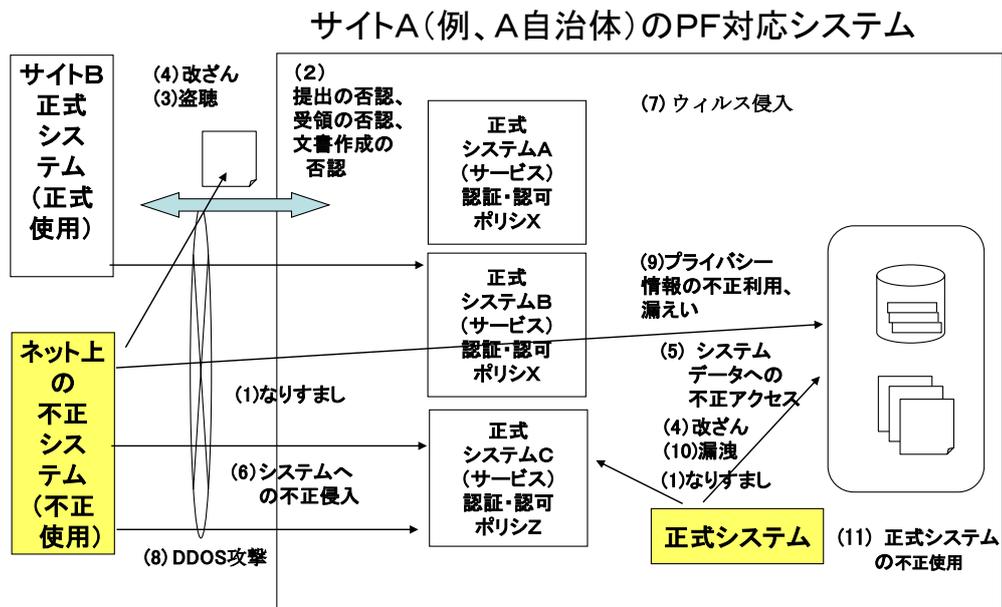


図 5. 1 異なるサイト間（自治体間、自治体・民間間等）におけるセキュリティ上の課題

異なるサイト間におけるセキュリティ上の課題に対する対策技術を下記に示す。

(a) 認証技術

- ・ 認証（Authentication）技術とは、本人であることを判定し証明する技術である。
（ここでいう本人とは、「クライアントのユーザ」「マシン」「サービス」の抽象的表現）
- ・ 対策対象： (1)なりすまし、(6)不正侵入

(b) 秘匿性確保技術

- ・ 秘匿性確保技術とは、通信の内容を第三者が参照できなくする技術である。
- ・ 対策対象： (3)盗聴、(10)漏洩

(c) 電子署名と検証技術

- ・ 電子署名と検証技術とは、情報に対し電子的な署名を行い、情報の改ざん検知や署名者の確認

を可能にする技術である。

- ・対策対象： (1)なりすまし、(4)改ざん
- (d) サービス認証・サービス認可連携技術
 - ・サービス認証・サービス認可連携技術とは、ワンストップサービス用のサービス認証・サービス認可情報の管理/伝播方法およびそのモデルに関する技術でシングルサインオンなどが実現できる。
 - ・対策対象： (1)なりすまし、(5)データの不正アクセス、(6)不正侵入
- (e) プライバシ情報公開技術
 - ・プライバシ情報公開技術とは、認証・認可の際やサービス処理におけるプライバシ情報の不正利用防止方法・交換許諾方法の技術である。
 - ・対策対象： (9)プライバシ情報の不正利用、(10)漏洩
- (f) 監査証跡技術
 - ・正式なシステムを不正に使用していないかの情報を統一的に収集し監査したい場合に、使用する技術である。
 - ・対策対象： (11)正式システムの不正使用
- (g) ウィルス対策技術
 - ・感染したビジネス文書やウィルスそのものの侵入検知や感染したデータの駆除等の技術である。
 - ・対策対象： (7)ウィルス侵入
- (h) DDoS 攻撃対策技術
 - ・複数のサイトから特定サイトへの集中アクセスによるシステムダウンやサービス停止の攻撃への対策技術である。負荷集中の監視やネットワーク機器の IP フィルタリング設定等の技術である。
 - ・対策対象： (8)DDoS 攻撃

5. 3 異なるサイト間における認証・認可・セキュリティ仕様

本節では、異なるサイト間（自治体間、自治体・民間間等）におけるセキュリティ上の課題に関する、対策技術の仕様を規定する。

5. 3. 1 PF サイト認証仕様

PF サイト認証仕様には、サイト間通信において、サービスのリクエスト側がサービス側の通信先マシン（サイト）を認証するサーバ認証と、サービス側がアクセスしてくるリクエスト側の PF 通信の通信元マシン（サイト）を認証するクライアント認証とがある。PF サイト認証仕様として、下記の国際標準仕様を採用する。

異なるサイト間での PF 通信では、PF サイト認証仕様は必須であり、サーバ間での通信においては TLS サーバ認証および TLS クライアント認証を必須とする。またリクエスト側がサーバではなく PC などの場合には TLS クライアント認証の代用として HTTP Basic 認証も可能とする。TLS のバージョンは、中間サーバとの接続も踏まえ、通信経路の盗聴、改ざんを防止するため、TLS 1.1 または TLS 1.2 を採用する。

○TLS サーバ認証

TLS で暗号通信を実施する場合、必須でサポートされている機能である。

○TLS クライアント認証

TLS クライアント認証により、Web サービスクライアントを認証する。TLS クライアント証明書は Web サービスクライアントごとに発行する。

○HTTP Basic 認証 (TLS クライアント認証が使えない場合の代用として使用する)

HTTP プロトコルで規定されているベーシック認証。ユーザ毎に、ID とパスワードをサーバ側で設定し、クライアントのアクセス時に指定し認証する方法。

5. 3. 2 PF 秘匿性確保仕様

PF 秘匿性確保仕様とは、通信路上のデータの内容を第三者から参照させないための仕様である。秘匿性確保のためには、通信路上の2点間の秘匿性確保と、End-to-Endでの秘匿性の確保が必要となる。しかし、PF 秘匿性確保仕様ではサイト間における秘匿性確保を対象としていることから、通信路上の2点間のPF 秘匿性確保仕様を異なるサイト間では必須とする。

(1) 通信路上の2点間の秘匿性確保仕様

ポータル、自治体、民間の間で、電子申請や公文書等を伝送される場合、データの漏洩を防止するため、**TLS 1.1 または TLS 1.2** 通信仕様を、通信路暗号として使用する。

(2) End-to-Endでの秘匿性確保仕様

自治体間においてWeb サービスが中継機関を介して連携される場合、本来であれば自治体でのみ参照すべき情報が中継機関にて参照される可能性がある。End-to-Endの秘匿性確保を行うことで、自治体間で連携する情報を中継機関に対して秘匿することが可能となる。

本機能を実現する標準仕様として、書類の形式に応じて、下記の仕様を選択する。尚、End-to-Endでの秘匿性確保仕様はオプションである。

- ・書類がXMLの場合、XML Encryptionを使用する。
- ・書類がファイルの場合、PKCS#7、または、XML Encryptionを使用する。

End-to-Endの秘匿性確保仕様の実現方式として表5.1に示す3つの方式が考えられる。ただし、公開鍵方式及びハイブリッド鍵方式の導入にあたっては、暗号用途の公開鍵基盤(PKI)の整備が前提となる。

表 5. 1 End-to-Endの秘匿性確保仕様の実現方式

実現方式	概要	メリット	デメリット
公開鍵方式	【送信側】受信側の公開鍵でメッセージを暗号化 【受信側】自身の秘密鍵でメッセージを復号化		<ul style="list-style-type: none"> ・処理が他の方式と比べて低速 ・公開鍵基盤 (PKI) の整備が必要
共通鍵方式	【送信側】受信側と事前に共有しておいた鍵 (共通鍵) でメッセージを暗号化 【受信側】共通鍵でメッセージを復号化	<ul style="list-style-type: none"> ・処理が他の方式と比べて高速 	<ul style="list-style-type: none"> ・各自治体で接続先ごとに共通鍵を管理する必要がある
ハイブリッド鍵 (公開鍵+共通鍵) 方式	【送信側】共通鍵でメッセージを暗号化。受信側の公開鍵で共通鍵を暗号化し、受信側に送付 【受信側】自身の秘密鍵で共通鍵を復号し、共通鍵でメッセージを復号化	<ul style="list-style-type: none"> ・処理が他の方式と比べて高速 	<ul style="list-style-type: none"> ・公開鍵基盤 (PKI) の整備が必要

5. 3. 3 PF 本人電子署名・検証仕様

PF 本人電子署名・検証仕様は、通信路の仕様ではないが、申請者が申請する申請書を異なるサイト間で交換する際に使用し、情報の改ざん検知や署名者の検証を可能にする仕様である。電子申請等、本人が作成したこと、および、改ざん防止が必要な場合に使用する。PF 本人電子署名・検証仕様は、オプションである。

なお、電子申請では、なりすましによる申請、データ改ざん、送信否認などのセキュリティ脅威が考えられる。これらの脅威を防ぐ場合には、申請書に電子署名を付与することが有効であり、下記の仕様を選択する。

- ・実現する技術としては、XML Signature 規格を使用する。
電子申請する本人の端末で申請データに電子署名を行い、電子署名のある電子申請を、オンライン申請する。申請データを受信した申請先は、その申請書の電子署名の検定を行う。
- ・住民が自治体に申請する書類の電子署名は、下記の JPKI を用いた技術標準を採用する。
JPKI に関しては、公的個人認証サービスポータルサイト (<http://www.jpki.go.jp/>) を参照。

5. 3. 4 PF 自治体組織電子署名・検証仕様

PF 自治体組織電子署名・検証仕様は、通信路の仕様ではないが、自治体が発行する照会書類や公文書を、異なるサイト間で交換する際に使用し、情報の改ざん検知や署名者の検証を可能にする仕様である。PF 自治体組織電子署名・検証仕様は、オプションである。PF 自治体組織電子署名・検証仕様としては、自治体間の組織認証基盤である「LGPKI」仕様を採用する。詳細は地域公共団体における組織認証基盤 (LGPKI) の Web サイト (<http://www.lgpkj.jp/>) を参照。

なお、異なるサイト間で照会書類や公文書を交換する場合、書類の形式に応じて、下記の仕様を選択する。

- ・書類が XML の場合、XML Signature を使用する。
- ・書類がファイルの場合、PKCS#7、または、XML Signature で署名する。
PKCS#7 の電子署名の場合は、ファイルと署名データを添付ファイルとして送信する。

5. 3. 5 PF サービス認証・サービス認可連携仕様

PF サービス認証・サービス認可連携仕様は、複数サイトとの間の連携処理にサービス認証やサービス認可処理の一元化が必要な場合に使用する。PF サービス認証・サービス認可連携仕様は、オプションである。なお、この機能は、サイトを跨る都度認証処理が必要になることでワンストップサービスが中断されることを防ぐことためのものであり、セキュリティ要件に応じ、サービス認証とサービス認可仕様を採用する。

(1) サービス認証

サービス認証は、異なるサイト間を跨いだサービスへのアクセス認証を伝播する仕様である。サービス認証仕様には、アグリゲーション型認証連携とプライバシー保護型認証連携が存在し、それぞれのセキュリティレベルを考慮し、採用する仕様を選択すること。

①サービス認証（アグリゲーション型認証連携）

アグリゲーション型認証連携は、ポータル等のサービスを統合するサイトが、自サイトの ID および認証情報と個別サイトが提供する認証情報を紐付けて保持し、自サイトでの認証後、紐付けされた個別の ID と認証情報によりバックの個々サイトのサービスを呼び出す方法である。

図 5.2 に、アグリゲーション型認証連携のモデルを示す。

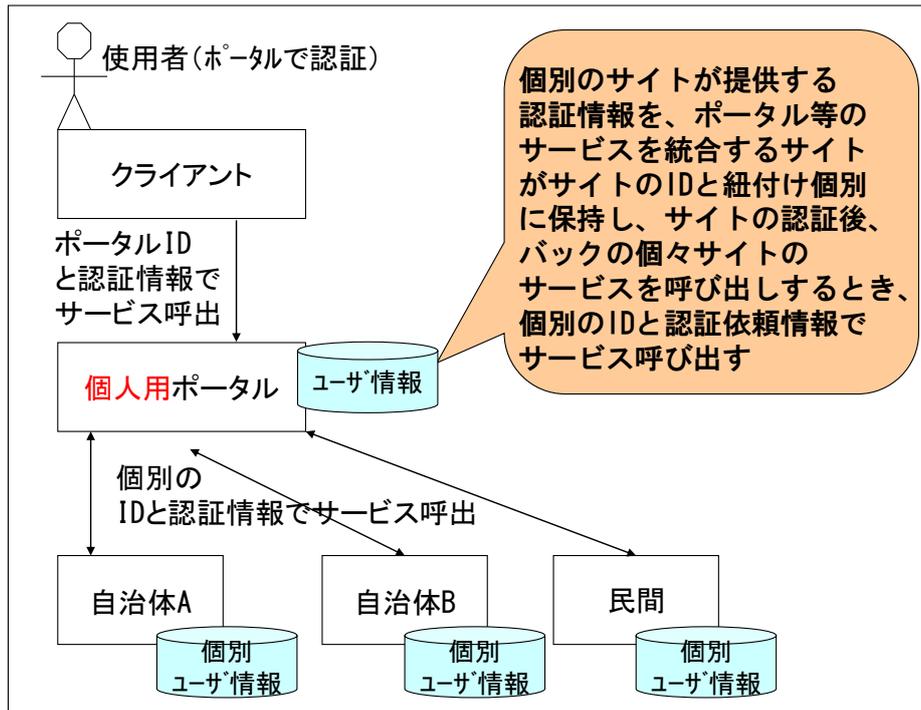


図 5. 2 アグリゲーション型認証連携

②サービス認証（プライバシー保護型認証連携）

プライバシー保護型認証連携は、ID 体系の異なるサイト間のサービス認証において、サービス間で認証情報の交換を可能とする機能（認証連携機能）と、認証情報を統合した場合に、複数サービスで使用されているプライバシー情報の名寄せを防止する機能（名寄せ防止機能）から成る。本機能を実現する標準仕様として、SAML V2.0 を採用する。

(a) 認証連携機能

ID 体系の異なる様々なサービスが連携する際に、サービス間でサービス認証情報およびプライバシー情報を交換することによってシングルサインオンを実現する機能である。

(b) 名寄せ防止機能

名寄せを防止するには、他のサービスで提供される認証情報の伝播を防ぐ必要があり、例えば認証用の ID とは別の ID を使用するなどで情報の連鎖を切ることが可能になる。サービス利用者に認証サーバ内で認証に使用する ID とは異なる仮 ID を発行し、各サービスに割り当てる。認証情報を交換する際は、認証サーバから連携サービスに割り当てられた仮 ID を受け取り、これを使用する。

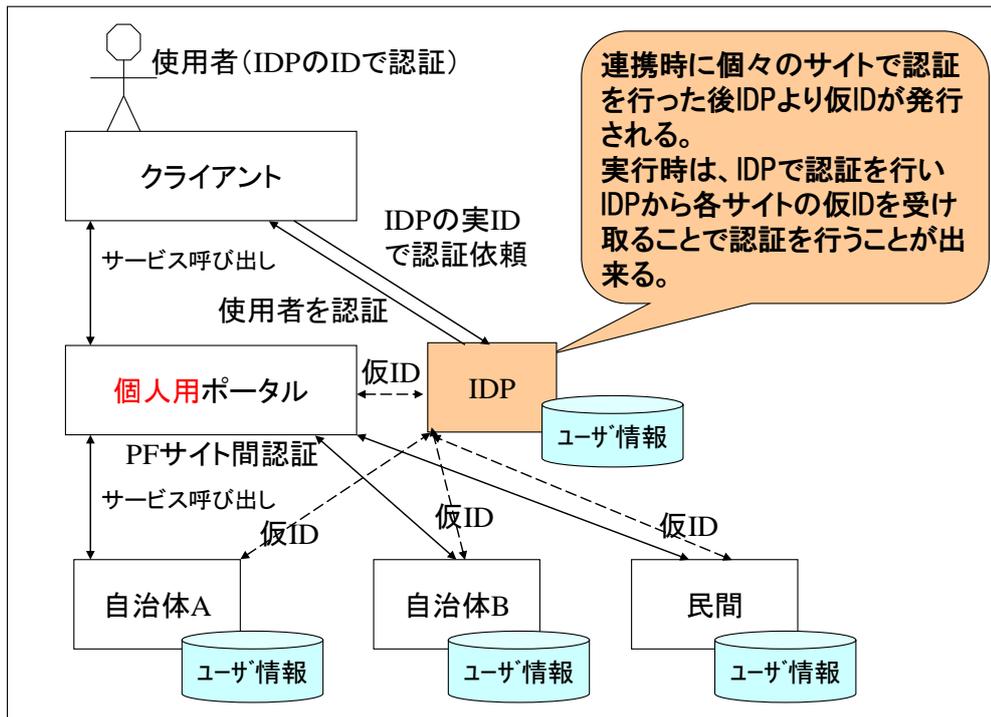


図5.3 プライバシ保護型認証連携技術

(2) サービス認可 (権限管理基盤)

権限管理基盤ではサービス認可の機能を提供するが、その機能は権限情報提供機能と、権限情報同期機能から構成される。

(a) 権限情報提供機能

権限情報提供機能では、アクセス制御ルールを共通化し一元管理する。また、権限管理基盤を提供することにより、共通化したアクセス制御ルールによる判定を可能にする。

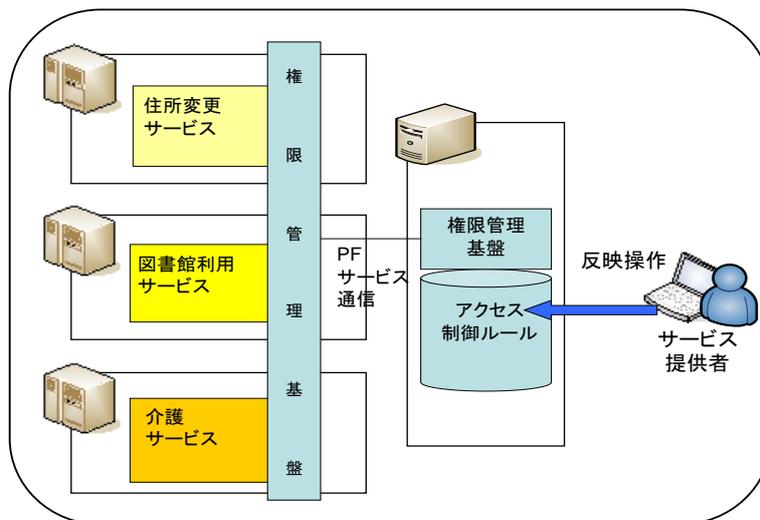


図5.4 権限情報提供技術のイメージ図

(ア) アクセス制御ルールの特標準化

- ・アクセス制御ルールは、標準化された言語を採用して共通化する。
- ・アクセス制御は、サイト内で統一した判定基準を採用する。

(イ) 統合的なアクセス制御ルールの管理

- ・すべてのアプリケーションに対して統合的（一元的）に編集し、一括したアクセス制御の変更ができるようにする。
- ・各アプリケーション向けインターフェースは、統一化して提供する。
- ・通信プロトコルは、相互接続に向けて標準化された仕様（SOAP）を採用する。

なお、自治体内での権限管理情報の一元管理では、LDAP、RDB 等が使われる。

(b) 権限情報同期機能

権限情報同期機能では、「権限情報の連携」および「権限情報の整合性の維持」によりサイト間のアクセス制御ルールの管理および判定を実現する。

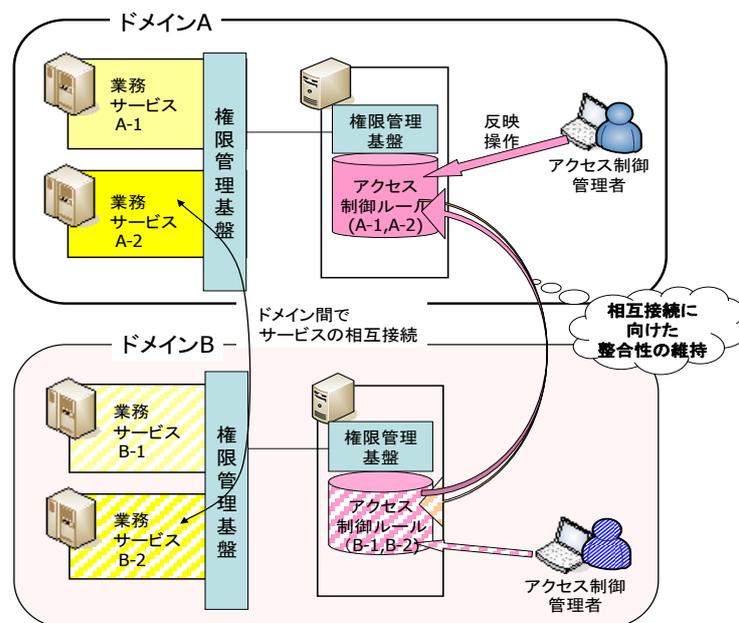


図5. 5 権限情報同期技術のイメージ図

(ア) 権限情報の連携

- ・権限情報は、他サイトで定義された権限情報に対しても適用できるように、サイト間で受け渡す際に標準化された言語に変換する。

(イ) 権限情報の整合性の維持

- ・権限情報は、他サイトで定義された権限情報に対しても適用できるように、自サイトの権限情報と他サイトの権限情報との関連に必要な情報の保持および管理ができるようにする。

5. 3. 6 PF プライバシ情報公開仕様

PF プライバシ情報公開仕様とは、複数サイトとの間の連携に際して認証や認可処理の一元化が必要な場合に使用するプライバシ情報の漏洩防止方法・交換許諾方法に関する仕様である。プライバシ情報を扱う際のポリシーの記述仕様として P3P V1.0 を採用する。PF プライバシ情報公開仕様は、オプションである。

(1) プライバシ情報公開の要件

複数サイト間でプライバシ情報を交換する場合、事前にエンドユーザに対して利用目的や提供先サイトなどを提示して許諾を得なければならない場合がある。また、プライバシ情報は、その交換範囲を、エンドユーザからの許諾に基づき制御しなければならない場合がある。

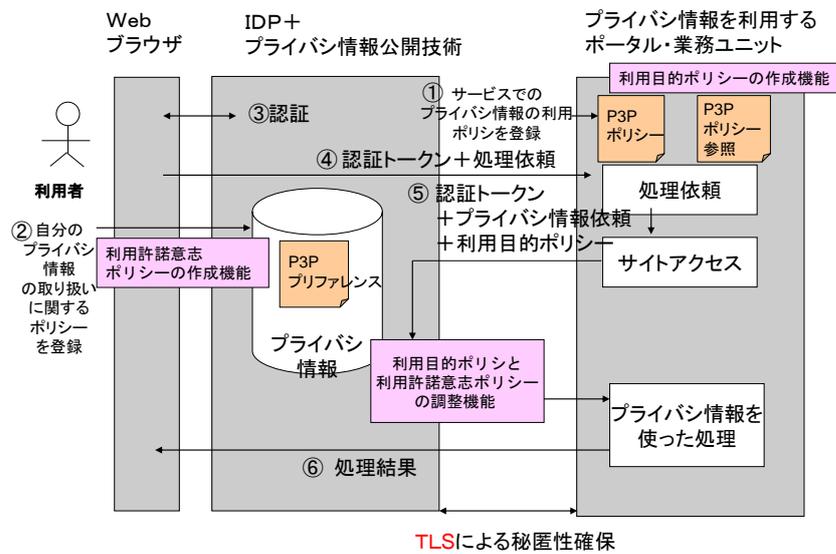


図 5. 6 P3P の PF 通信機能 (SOAP) への適用例

また、プライバシ情報を保有するサイトとエンドユーザがアクセスするサイトが異なる場合であっても、アクセスする側のサイトでエンドユーザの許諾を取得し、プライバシ情報を保有するサイトまで許諾情報を連携することでプライバシ情報の交換を制御することが可能である。

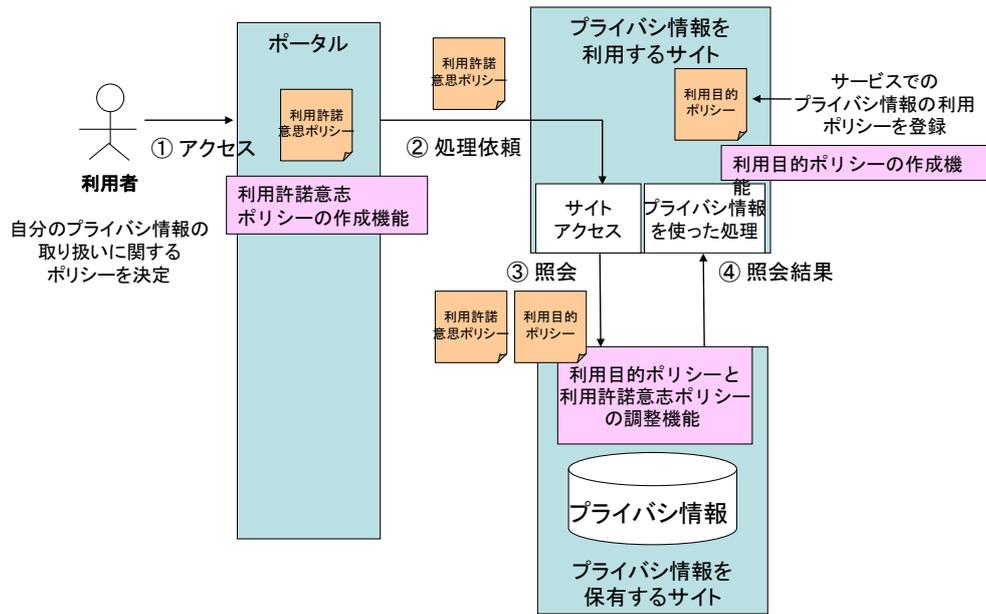


図 5. 7 許諾情報を連携させる場合の適用例

(2) プライバシ情報公開の機能

プライバシー情報の漏洩防止および交換許諾のための仕様として、図 5. 7 の P3P の個人情報公開技術の適用例を実現する次の機能を具備すること。

(a) 利用目的ポリシーの作成機能

提供するサービスの内容に応じて、使用するプライバシー情報の内容、およびその利用目的を記述した、サービスの提供者におけるプライバシー情報の利用目的ポリシー (P3P ポリシーと、P3P ポリシー参照で記述) を作成する機能である。

(b) 利用許諾意志ポリシーの作成機能

利用するサービスの内容に応じて、利用されるプライバシー情報の内容、およびその利用目的に応じて、許諾の意志を記述した、サービスの利用者におけるプライバシー情報の利用許諾意志ポリシー (P3P プリファレンスの記述) を作成する機能である。

(c) 利用目的ポリシーと利用許諾意志ポリシーの調整機能

プライバシー情報を提供するサイトが、プライバシー情報を利用するサイトから送付された利用目的ポリシーと、利用許諾意志ポリシー間の調整を行う機能。利用許諾意志ポリシーは、予めプライバシー情報を提供するサイトがエンドユーザから取得することもできるが、プライバシー情報を利用するサイトがエンドユーザから取得して送付することもできる。

ポリシーが適合すればプライバシー情報をサービス提供者に提供し、ポリシーが適合しない場合にはプライバシー情報を提供しない。

なお、本書では上記の機能についての仕様等についての規定は行わず、機能のみの規定とする。

5. 3. 7 PF 監査証跡仕様

PF 監査証跡仕様は、個々に点在するセキュリティの監査証跡情報や複数サイト間に跨るセキュリティ監査証跡情報を統一的に管理し、監査する際の基本的な仕様（考え方）について規定したものである。PF 監査証跡仕様はオプションである。

なお、本仕様は、自治体内等の単一サイト内における監査証跡について規定するものではない。

(1) PF 監査証跡仕様の対象範囲

PF 監査証跡仕様にはシステム監査とアクセス記録が存在する。

- (a) システムに対する不正アクセスや不正利用等を検知するためにシステム管理者が実施する監査（システム監査）
- (b) 自治体が保有する住民情報に対する不正アクセスや不正利用等を検知するために住民本人が実施する監査（アクセス記録）

尚、アクセス記録については、以下の要件が存在する。

- ・自治体間連携におけるアクセス記録を出力できること
- ・自治体間連携で用いる ID により、特定の住民に対するアクセス記録を収集できること
- ・住民が参照可能なのは、自己情報に対するアクセス記録のみとなるよう制御できること

(2) PF 監査証跡仕様の機能

PF 監査証跡仕様に基づき、サイト内の監査情報を、外部のサイトに公開するサービスを、監査証跡情報提供サービスと呼ぶ。以下に、監査証跡情報提供サービスにおけるシステム構成例を示す。

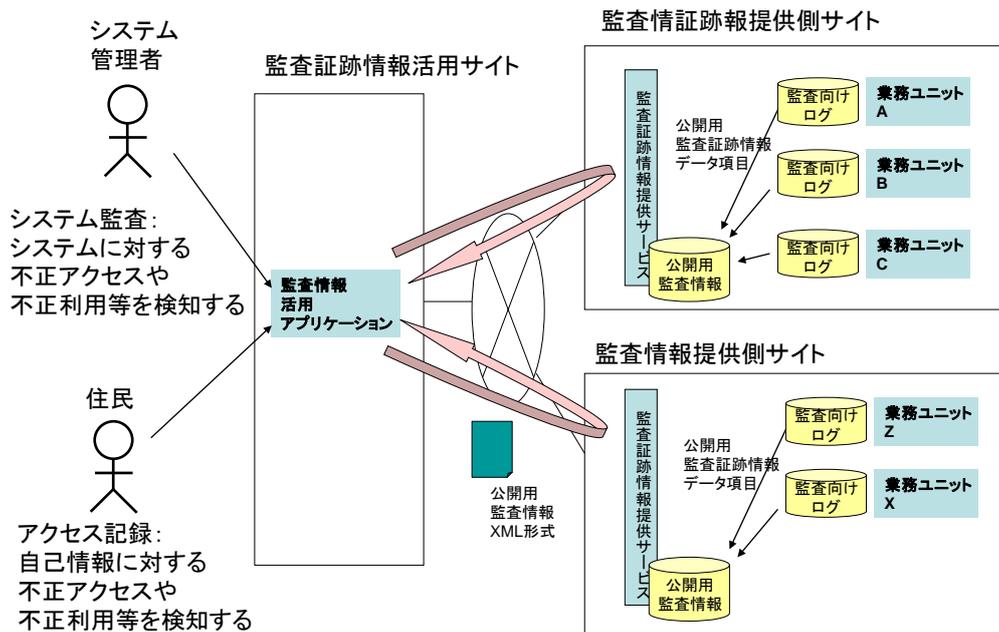


図 5. 8 監査証跡情報提供サービス機能のシステム構成例

また監査証跡情報提供サービスに関連するアプリケーションについて表 5.2 に示す。

表 5. 2 監査証跡情報提供サービスに関連するアプリケーション

アプリケーション名	機能概要
業務ユニット	監査対象の監査ログを生成するアプリケーション。
監査証跡提供サービス	監査証跡取得機能を実現するサービス。サイト間で公開する公開用監査情報を自治体内のログから取得蓄積管理し、PF 通信機能での取得依頼に基づき、情報を提供するサービス。
監査情報活用アプリケーション	監査ログの収集サービスにより収集された監査ログを、監査を目的に閲覧や分析するためのアプリケーション

ヘルスケア分野における関連情報としては、監査ログ生成の契機として、RFC3881 で定義されたイベントカテゴリが、また、ヘルスケア分野における監査証跡のメッセージ標準規約が以下の URL にある。

<http://www.jahis.jp/standard/seitei/st06-002/st06-002.htm>

PF 監査証跡仕様としてのサイト間で共有する共通監査情報項目については、今後、サイト間で実施する監査要件を明確化する検討を実施し、その結果に基づき、情報項目を規定していく。

<<準拠ルール>>

【サイト間】

- (CS-R050001) : サイト間で PF 通信機能を使用する際は、「PF サイト認証仕様」を使用する。(必須) [5. 3. 1]
- (CS-R050002) : サイト間で PF 通信機能を使用する際は、「PF 秘匿性確保仕様」を使用する。(必須) [5. 3. 2]
- (CS-R050003) : サイト間で電子申請する場合は「PF 本人電子署名・検証仕様」を使用してよい。
(オプション) [5. 3. 3]
- (CS-R050004) : 自治体間で、文書の照会を行う場合、「PF 自治体組織電子署名・検証仕様」を使用する。
(必須) [5. 3. 4]
- (CS-R050005) : 複数サイト間の連携処理にプライバシー情報を保護し、サービス認証やサービス認可処理の一元化が必要な場合、PF サービス認証・サービス認可連携仕様を使用してよい。
(オプション) [5. 3. 5]
- (CS-R050006) : サイト間でプライバシー情報を交換する場合、PF プライバシ情報公開仕様を使用してよい。
(オプション) [5. 3. 6]
- (CS-R050007) : サイト間で正式なシステムを不正に使用していないかの情報を統一的に収集し監査したい場合、PF 監査証跡仕様を使用してよい。(オプション) [5. 3. 7]

用語辞書

項番	用語	意味
1	認証技術	本人であることを判定し証明する技術 対策対象： なりすまし、不正侵入
2	秘匿性確保技術	データの内容を第三者が参照できなくする技術 対策対象： 盗聴、漏洩
3	電子署名と検証技術	電子署名と検証技術とは、情報に対し電子的な署名を行い、情報の改ざん検知や署名者の確認を可能にする技術 対策対象： なりすまし、不正侵入、データの不正アクセス

項番	用語	意味
4	プライバシー情報公開技術	プライバシ情報公開技術とは、認証・認可におけるプライバシー情報の漏洩防止方法・交換許諾方法の技術 対策対象： プライバシ情報の不正利用や漏洩
5	監査証跡技術	正式なシステムを不正に使用していないかの情報を統一的に収集し監査したい場合に、使用する技術 対策対象： 正式システムの不正使用

6. プラットフォーム通信機能における MEP と異常系処理

本章では、プラットフォーム通信機能におけるサービス通信でのメッセージ交換パターン (MEP : Message Exchange Patterns) と異常系処理について説明する。

業務サービスの依頼やそれに伴う業務データの受け渡しは、サービス基盤上の処理系同志、あるいはその処理系と業務サービス間において為される業務メッセージの交換で行われる。しかし一般には、通信によるメッセージ交換は様々な要因で失敗し、業務上の障害にも成り得る。こうした障害を検知し、適切に対応するという点で、到達の確認、処理完了を示すレスポンスの送受信が必要となり、これら組み合わせから、様々なメッセージ交換のパターンが定義される。本章では、この様なメッセージ交換パターンと障害対応の異常系処理について、下記構成で説明する。

- (1)用語の定義
- (2)各メッセージ交換パターンの解説と本標準仕様での推奨
- (3)推奨メッセージ交換パターンにおける障害検知方法
- (4)異常系処理
- (5)データ交換システムパターンとメッセージ交換パターンの関係
- (6)その他の規定

6. 1 概念、用語の定義

6. 1. 1 基本用語定義

- ・メッセージ交換パターン (MEP : Message Exchange Patterns) : 後述するメッセージ群の交換の種類を定義するものである。
- ・MEP 処理系 : BPM (Business Process Management) 処理系、WS-BPEL (Web Services Business Process Execution Language 2.0) 処理系等の狭義の処理系のこと。
- ・MEP 実行系 : MEP を処理するソフトウェアの総称のこと。前述 MEP 処理系が中心となるが、MEP 処理系を利用する業務ユニットレベルのものも含め、広義に定義する。
- ・エンティティ : OSI (Open Systems Interconnection) の定義に基づき、メッセージを扱うものをエンティティと呼ぶ。具体的には、サービス、プロセスを意味する。

6. 1. 2 メッセージの種類に関する定義

本章で扱うメッセージの種類を、以下の様に定義する。

- ・要求メッセージ、リクエスト (req) : 業務サービス依頼を意味する要求メッセージであり、地域情報 PF 仕様における MEP 開始の送信メッセージ。
- ・応答メッセージ、レスポンス (res) : リクエストに対する業務サービス処理結果を返す際のメッセージ。
- ・受領 Ack (Acknowledgement) (ack) : ビジネスプロセス、もしくは業務サービスでリクエスト、レスポンスを受信したことを送信元に通知するためのメッセージ。この受領 Ack は、デフォルトとして「受信直後の業務ユニットレベルの処理前」に送付されるものとする。これは、TCP/IP、高信頼通信機能などの受領確認とは異なり、通常の業務メッセージとして表現される受領確認である。詳細は 6. 6 に記す。

6. 1. 3 メッセージを扱うエンティティの定義

エンティティは役割に応じて以下の様に定義する。

- ・ 開始側 (initiator) : 要求メッセージを送信するサービス、あるいは BPM 機能内のプロセスのことで、この仕様では、「発呼側」と呼ぶこともある。
- ・ 応答側 (responder) : 要求メッセージを受信し、必要に応じてレスポンスを送信するサービス、BPM 機能内のプロセス、あるいは、統合 DB 機能のことで、この仕様では「着呼側」と呼ぶこともある。
- ・ メッセージ送信側 : 特に SOAP (Simple Object Access Protocol) 通信レベル (高信頼性通信の場合は、そのレベル) から下位レベルでメッセージを送信する側のことで、上記「開始側」とは異なる。(「応答側」も「メッセージ送信側」に成り得る。) なお、高信頼性通信機能では、単に「送信側」と呼ぶこともある。
- ・ メッセージ受信側 : 特に SOAP 通信レベル (高信頼性通信の場合は、そのレベル) から下位のレベルでメッセージを受信する側のことで、上記「応答側」とは異なる。(「開始側」も「メッセージ受信側」に成り得る。) なお、高信頼性通信機能では、単に「受信側」と呼ぶこともある。

6. 1. 4 応答の振る舞いに関する類型定義

応答の振る舞い方として幾つかの分類が可能である。以下に列挙する。

- ・ リクエスト型 : リクエスト (req) のメッセージだけ、もしくはリクエスト (req) と受領 Ack (ack) のメッセージだけで、それに対する応答が無いもの。この場合、受領 Ack (ack) をレスポンス (res) のメッセージとは扱わない。
- ・ リクエスト・レスポンス型 : リクエスト (req) のメッセージに対するレスポンス (res) のメッセージが存在するもの。ただし、受領 Ack (ack) をレスポンス (res) のメッセージとは扱わない。
- ・ 同期 : 受信内容に対する返信 (受領 Ack、レスポンス) が、受信時と同一の HTTP (Hypertext Transfer Protocol) セッションで返される通信パターン。HTTP リクエストに対する HTTP レスポンスによる応答がこれにあたる。
- ・ 非同期 : 受信内容に対する返信 (受領 Ack、レスポンス) が、受信時と異なる別の HTTP セッションで返される通信パターン。HTTP リクエストのみに業務メッセージを載せて相互に通信することがこれにあたる。

6. 1. 5 高信頼性通信機能と MEP の関係

高信頼性通信機能は、通信路の信頼性を向上させる。これに対して、MEP ではその上位概念を扱う。そのため、以下の様な基本方針に従う。

- ・ 高信頼性通信機能と MEP については基本的には異なるレイヤとして扱う。
- ・ MEP の受領 Ack は業務ユニットレベルの送達確認であり、高信頼性通信機能のものとは意味・役割が異なる。

6. 1. 6 表記法説明

各メッセージ交換パターンを概説する上で、一般的に利用される UML (Unified Modeling Language) のシーケンス図を用いる。表記に当っては下記制約、特記事項を設ける。

- ・ 後述する図中のシーケンス図の長方形は、通常の意味では、通信における 1 つのセッション境界を示す。

ただし、同じ形式でトランザクション境界や、オブジェクトの生存期間を意味する場合もある。

- ・ 上記を受けて、2つの長方形を続けて記す場合は、非同期を意味する。
- ・ 図中で表現されるメッセージは、6. 1. 2で定義されたもの以外は表現されない。

6. 2 プラットフォーム通信機能における推奨メッセージ交換パターンの定義

6. 2. 1 概説

【共通】

メッセージ交換パターンは、リクエスト型、リクエスト・レスポンス型、同期、非同期等で種々のパターンが存在する。プラットフォーム通信機能では SOAP1.1 の Req-Res パターンとの合致、および、実装の容易さを考慮し、下記3パターンを採用する。メッセージ交換パターンは、メッセージ送受信に対する業務要件に依存するため、下記のいずれかを選択するかは、業務設計時に決定する。

- (1) リクエスト型受領 Ack あり
- (2) リクエスト・レスポンス型同期型レスポンス
- (3) リクエスト・レスポンス型同期型受領 Ack + 非同期型レスポンス

6. 2. 2 メッセージ交換パターン「リクエスト型受領 Ack あり」

このパターンは要求メッセージのみで構成され、業務データを含む応答メッセージの返信が無いパターンである。ただし、開始側が、リクエスト送信の HTTP セッションの戻りで受領 Ack を受信する。応答側での受領を確認はできるが、処理結果が開始側に返されないため、要求メッセージに基づく処理結果については開始側で確認できない制約は存在する。

- ・ 開始側：要求メッセージの送信 ⇒ 受領 Ack 受信 ⇒ 受領 Ack チェック
- ・ 応答側：(要求メッセージの受信) ⇒ 受領 Ack の生成 ⇒ 受領 Ack 返信 ⇒ 要求に基づく処理

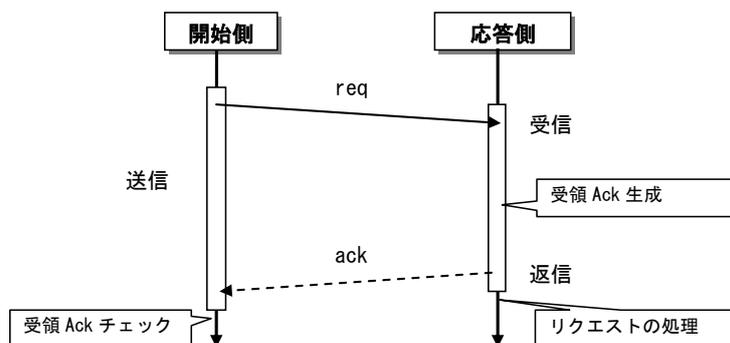


図 6. 1 「リクエスト型同期型受領 Ack あり」の処理概要

6. 2. 3 メッセージ交換パターン「リクエスト・レスポンス型同期型レスポンス」

リクエスト・レスポンス型は要求メッセージとそれに対する業務データを含む応答メッセージの組で構成される。受領 Ack の有無、および、受領 Ack と応答メッセージの送受信の仕方等で種々のパターンが定義できるが、このパターンにおいては要求メッセージ、応答メッセージに対する受領 Ack を行わない。開始側は応答メッセージをリクエスト送信の HTTP セッションの戻りで受信する。このパターンは、要求

メッセージに対する応答側での応答メッセージ返信開始までの処理が、開始側の HTTP セッションのタイムアウト制限時間よりも短い時間に終了し得る場合に使用する。開始側の HTTP セッションのタイムアウト制限時間よりも長く処理が掛かる場合は、すべて開始側でタイムアウトとして検出される。

- ・ 開始側：要求メッセージの送信 ⇒ 応答メッセージの受信 ⇒ 受信後の処理
- ・ 応答側：要求メッセージの受信 ⇒ 要求に基づく処理、応答メッセージの生成 ⇒ 返信

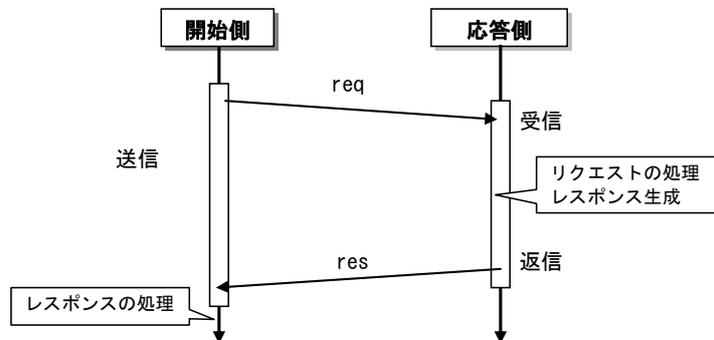


図 6. 2 「リクエスト・レスポンス型同期型レスポンス」の処理概要

6. 2. 4 メッセージ交換パターン「リクエスト・レスポンス型同期型受領 Ack+非同期型レスポンス」

このパターンでは要求メッセージに対する受領 Ack を同一の HTTP セッションで返信し、応答メッセージに対する受領 Ack も同様に同一の HTTP セッションで返信する。また原則として、要求メッセージと応答メッセージは別の HTTP セッションで非同期型として送受信するが、例外も存在する。応答側が MEP 処理系で構成され、一時に複数の再呼出しを行う場合は、呼び出し関係を維持しなければならないため、応答側は連続して存在しなければならない場合もある。詳細についてはガイドラインにて記す。

変則的に応答側にて応答メッセージに対する受領 Ack を無視する場合も許容する。要求メッセージの送信に障害が発生した場合、受領 Ack のタイムアウト検出により異常系処理を実施する。

- ・ 開始側：要求メッセージ送信 ⇒ 受領 Ack 受信 ⇒ 受領 Ack チェック ⇒ 応答メッセージの受信 ⇒ 受領 Ack 生成 ⇒ 受領 Ack 返信 ⇒ 応答メッセージ受信後の処理
- ・ 応答側：要求メッセージ受信 ⇒ 受領 Ack 生成 ⇒ 受領 Ack 返信 ⇒ 要求に基づく処理 ⇒ 応答メッセージ生成 ⇒ 応答メッセージ返信 ⇒ 受領 Ack 受信 ⇒ 受領 Ack チェック

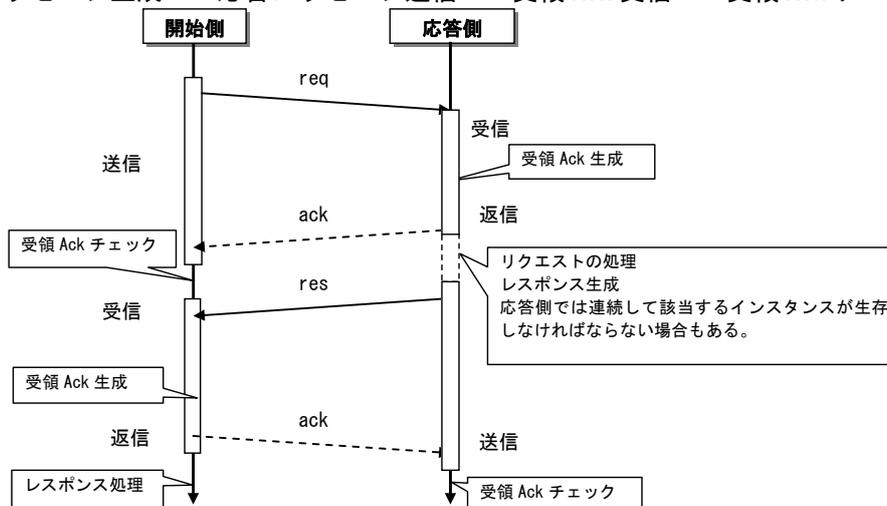


図 6. 3 「リクエスト・レスポンス型同期型受領 Ack+非同期型レスポンス」の処理概要

6. 3 プラットフォーム通信機能の通信障害とその検知方法定義

本節では、障害検知に関する定義、プラットフォーム通信機能の中で発生する障害類型と、実装すべき検知方法等を定義する。また、障害検知と関連する異常系処理は 6. 4 にて定義する。

6. 3. 1 障害検知に関する定義

障害に対しては、下記 3 段階で処理が実施される。

- ・ 障害の検知
- ・ 即時対応処理
- ・ 異常系処理

障害検知とは、異常を捕捉することである。即時対応処理とは、障害検知を受け、それを通知する等の問題解決を伴わない対応処理である。異常系処理とは、障害検知、即時対応処理を受けて、障害を解決するための一連の処理である。プラットフォーム通信機能における障害検知、即時対応処理については、以下の基本方針に基づく。

- ・ 障害検知に関しては、発生する真の異常を各々捕捉するのではなく、それにより誘発され、通信各機能層で認識される障害を捕捉することで検知する場合も含む。
- ・ 即時対応処理に関しては、基本的に通信の開始側で責任を持つ。また、通信の応答側では業務ユニットレベルでの拒絶、内部発生障害の通知以外の明示的な処理は行わない。

以下、プラットフォーム通信機能における障害検知、即時対応処理の各詳細方法について説明する。定義された「障害検知方法」、「即時対応処理」については、プラットフォーム通信機能の中で実装されなければならない。

6. 3. 2 プラットフォーム通信機能の通信障害の類型

プラットフォーム通信機能のアーキテクチャ上で発生しうる、通信障害は以下のように分類される。

(分類 1) TCP/IP、HTTP レベルで検知される障害

- ・ 通信相手のサーバダウン
- ・ ネットワークの物理的断絶
- ・ プロトコル処理障害
- ・ その他の TCP/IP、HTTP の階層で捕捉される一切のもの

(分類 2) SOAP 通信レベル（高信頼性通信の場合は、そのレベル）で検知される障害

- ・ SOAP 送受信機能の不備
- ・ XML (Extensible Markup Language) のシンタクス異常
- ・ XML 上の取り決め違反
- ・ 受信タイムアウト (SOAP 通信レベルでの受信タイムアウト)
- ・ SOAP 通信レベルのプロトコル処理障害
- ・ その他、SOAP 通信、高信頼性通信で捕捉される一切のもの

(分類 3) MEP 実行系レベル以上で検知される障害

- ・ 受信タイムアウト (WS-BPEL 処理系レベルのタイムアウト)
- ・ 受領 Ack、レスポンスでの障害報告の受信

- ・業務ユニットレベルでの拒絶
- ・業務ユニットレベルの内部発生障害
- ・非同期時受信タイムアウトを含む業務ユニットレベルの処理タイムアウト

6. 3. 3 TCP/IP、HTTP レベルの障害検知、即時対応処理方法（メッセージ送信側）

【共通】

- (1) SOAP 通信レベル（高信頼性通信の場合は、そのレベル）より下位レベルとなる TCP/IP、HTTP レベルの障害については、SOAP 処理系（高信頼性通信の場合は、その処理系）の障害として検知する。従って採用する SOAP 処理系は、下位レベルで検知した障害が通知されなければならない。

6. 3. 4 TCP/IP、HTTP レベルの障害検知、即時対応処理方法（メッセージ受信側）

【共通】

- (1) 通常は、通信プロトコルにおける検知として扱われる。
- (2) 障害検知の結果、実施される即時対応処理では、プロトコルの規定に従い、メッセージ送信側に障害を通知する。ただし、メッセージ送信側に障害を通知できない場合は、そのままにする。
- (3) 即時対応処理として、メッセージ受信側の上位レベルに通知することはオプションである。

6. 3. 5 SOAP 通信レベル（高信頼性通信も含む）の障害検知、即時対応処理方法（メッセージ送信側）

【共通】

- (1) SOAP 処理系（高信頼性通信の場合は、その処理系）自身の障害は、その SOAP 処理系（高信頼性通信の場合は、その処理系）の障害として扱う。メッセージ送信側の SOAP 処理系で障害検知した結果、即時対応処理として、メッセージ送信側内部で適切な異常系処理を行い、その上位レベルに障害を通知しなければならない。
- (2) 業務レスポンス、受領 Ack に関係なく、メッセージ受信側と同期返信を期待する場合、SOAP 通信レベルもしくはそれより下位レベルのプロトコルで設定されるタイムアウトに伴う障害は SOAP 処理系で検知する。従って、採用する SOAP 処理系では、タイムアウト検知が可能でなければならない。これによりメッセージ受信側の TCP/IP、HTTP レベルの障害、並びに SOAP 処理系（高信頼性通信の場合は、その処理系）自身の障害は、メッセージ送信側の SOAP 処理系（高信頼性通信の場合は、その処理系）のタイムアウトにより検知される。
- (3) 高信頼性通信機能における再送を前提とした受領確認、障害報告に対しては、即時対応処理として上位レベルへの通知を直ちに実施せず、高信頼性通信機能に従い、処理を継続する。その後、以下の通り処理する。
 - － メッセージ送信側ではすべての再送に失敗した場合に限り、一度だけ障害を上位レベル（送信側の MEP 実行系）に通知しなければならない。

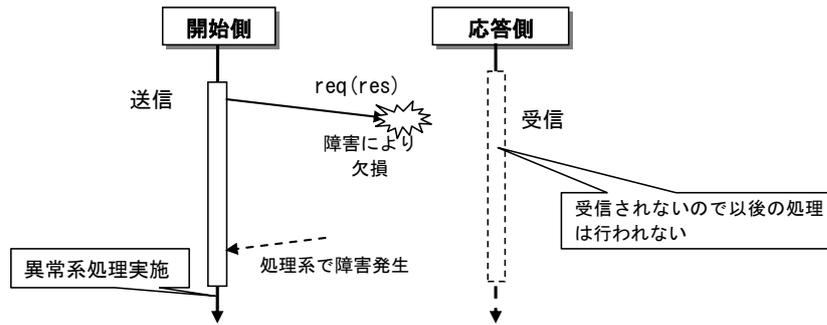


図 6. 4 メッセージ送信側の SOAP 通信レベル（高信頼性通信の場合は、そのレベル）
障害の検知

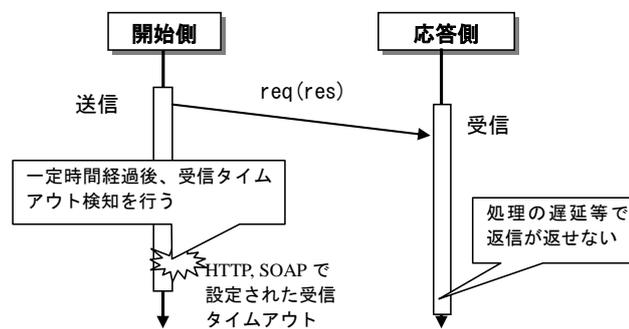


図 6. 5 メッセージ送信側の SOAP 通信レベル（高信頼性通信の場合はそのレベル）
タイムアウト検知

6. 3. 6 SOAP 通信レベル（高信頼性通信も含む）の障害検知、即時対応処理方法（メッセージ受信側）

【共通】

- (1) メッセージ受信側の SOAP 処理系で、受信したメッセージに対する受領 Ack もしくはレスポンス (res) の同期返信時に障害検知した場合、下記の即時対応処理を行う。
 - 後にメッセージ送信側に対して障害通知が可能である場合は、メッセージ送信側に障害を通知する。
 - メッセージ送信側に障害の通知ができない場合は、通信を行わなくとも良い。（後にメッセージ送信側の SOAP 処理系でタイムアウトにより検知される。）
- (2) 高信頼性通信機能における再送を前提とした受領確認、障害報告に対しては、即時対応処理として上位レベルへの通知を直ちに実施せず、高信頼性通信機能に従い、処理を継続する。その後、以下の通り処理する。
 - メッセージ受信側では送受信に成功した場合、受信したメッセージを上位レベル（メッセージ受信側の MEP 実行系）に通知する。

6. 3. 7 MEP 実行系レベル以上の障害検知、即時対応処理方法（開始側）

(1) 開始側の MEP 実行系レベルでは、複数のケースが存在する。最初に共通的に検知する方法を定義の上で、障害の個別ケースについて検知方法を定義する。開始側の MEP 実行系レベルでの障害は、下記の 5 事象に集約される。

- ① 受信タイムアウトが発生するケース
- ② 応答側の MEP 実行系から障害を示す通知もしくは受領 Ack を受信したケース
- ③ 業務ユニットレベルを含む、MEP 実行系で障害が発生するケース
- ④ 応答側の業務ユニットレベルのレスポンスに対して受信タイムアウトが発生するケース
- ⑤ 下位レベルから障害を通知されるケース

【共通】

(2) 開始側の MEP 実行系レベルで共通的に実施される検知方法

以下の 2 つの検知方法が用意されなければならない。

- ① (検知方法 1) すべてのメッセージ交換パターンで、開始側の MEP 実行系レベルで必須のレスポンス、または受領 Ack を受信した場合、受信したメッセージの共通ヘッダ部内の項目（XML 定義においては要素）の No. 10 結果情報の値がエラー（障害）か、否かを検知しなければならない。
- ② (検知方法 2) 開始側の MEP 実行系レベルの上位に位置づけられる業務ユニットレベルで、応答が無い場合の受信タイムアウトを検知する。なお、業務ユニットレベルで受信タイムアウト検知をした場合に、MEP 実行系レベルで WS-BPEL プロセスへの後処理が必要となる。これはガイドラインに記載する。

(3) 受信タイムアウトが発生するケース

受信タイムアウトが発生するケースは、メッセージ交換パターン「リクエスト・レスポンス型同期型レスポンス」のレスポンス受信時と、メッセージ交換パターン「リクエスト・レスポンス型同期型受領 Ack+非同期型レスポンス」の受領 Ack 受信時が相当する。

これらのケースでは、共通的に実施される(検知方法 1)(検知方法 2)を必要としない。これら同期返信の場合、タイムアウト検知は、同期処理を実現する SOAP 通信レベル（高信頼性通信の場合は、そのレベル）以下の処理系で対応できるため、6. 3. 5 の SOAP 処理系（高信頼性通信の場合は、その処理系）での障害として扱うことができる。ただし、オプションとして非同期メッセージの受領 Ack 受信を、例えば HTTP セッションタイムアウトの様な SOAP 以下の処理系のタイムアウトよりも短く設定したい場合、直接、MEP 実行系レベルの受信タイムアウトにより検知することも可能である。

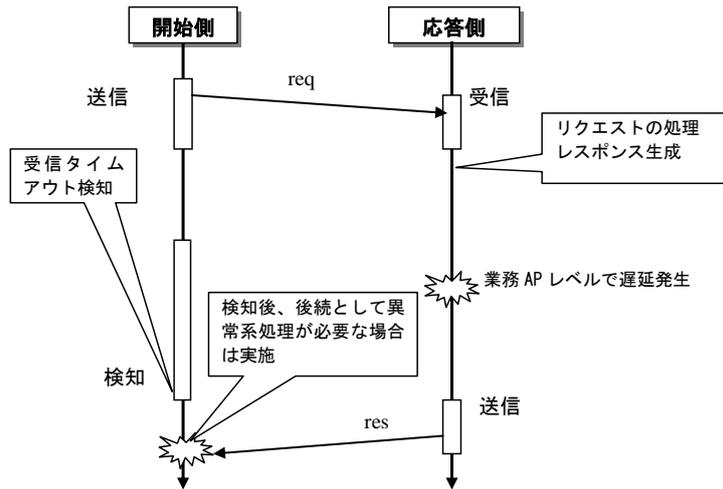


図 6. 6 非同期メッセージ受信に対するタイムアウト検知

(4) 応答側の MEP 実行系から障害を示す通知もしくは受領 Ack を受信したケース

このケースでは、共通的に実施される（検知方法 1）の利用が前提となる。検知の上で即時対応処理を実施する。このケースに対する例外規定はない。

特記事項として、共通的に実施される検知方法で障害検知されないにも関わらず、業務メッセージ内の独自領域に障害状態が表現される場合が有り得る。これは、業務ユニットレベルの処理を呼び出す際に、相応の理由で、業務ユニットレベルでシステム異常等を検知した際に発生する。この場合、通常は業務ユニットレベルで後処理を行うものの、業務ユニットレベルが呼び出す MEP 処理系における検知、異常系処理が必要な場合は、MEP 処理系での障害の検知、異常系処理に委託するため、業務ユニットレベルの処理においてシステム制御情報に障害を示す値をセットし直し、共通的に実施される検知方法で、改めて検知させることも可能である。これは、4. 3. 3の BP のフロー制御が関係する。

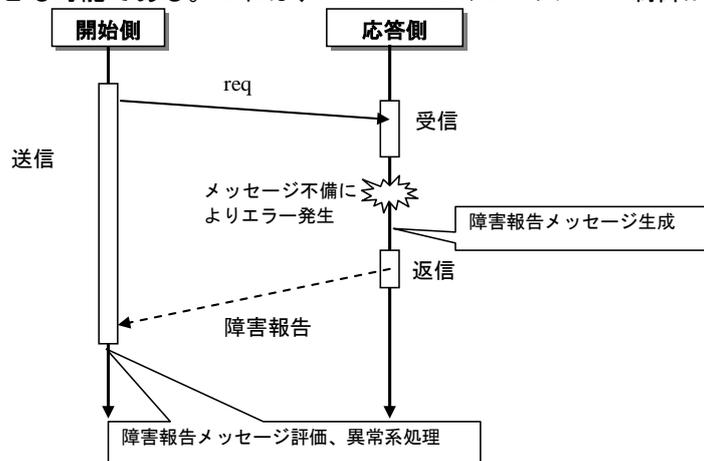


図 6. 7 障害報告メッセージによる障害検知の例

(5) 業務ユニットレベルを含む、MEP 実行系で障害が発生するケース

このケースでは、共通的に実施される（検知方法 1）（検知方法 2）は特に必要としない。開始側の MEP 実行系自身の障害を検知した場合は、応答側の MEP 実行系自身の障害を検知した場合と全く同等の即時対応処置を実施しなければならない。

その後の異常系処理は、呼出元の業務ユニットレベルで対応する。これは、当該呼出元の業務ユニットが、独立した別のサービス呼出により起動している場合が考えられるためである。その様な場合、障

害発生のお知らせを連鎖的に遡流させることで、不整合の発生を防止する。

(6) 応答側の業務ユニットレベルのレスポンスに対して受信タイムアウトが発生するケース

このケースは、メッセージ交換パターン「リクエスト・レスポンス型同期型受領 Ack+非同期型レスポンス」における非同期型レスポンスで発生する。その原因の1つに、応答側の業務処理遅延などに伴い、レスポンスが遅れる等がある。

このケースの検知方法としては、共通的に実施される(検知方法2)で対応する。また、オプションとして直接、MEP 実行系レベルの受信タイムアウトにより検知することも利用してよい。

(7) 下位レベルから障害を通知されるケース

このケースは SOAP 処理系で検知した障害を報告するものである。そして、共通的に実施される(検知方法1)を必要とする場合、必要としない場合の2通りが存在する。このケースの即時対応処理は、受信タイムアウトが発生するケースと同様である。

6. 3. 8 MEP 実行系レベル以上の障害検知、即時対応処理方法 (応答側)

(1) 応答側の MEP 実行系レベルでは、2つのケースが存在する。最初に共通的に検知する方法を定義の上で、障害の個別ケースについて検知方法を定義する。

【共通】

(2) 応答側の MEP 実行系レベル以上で共通的に実施される検知方法

以下の検知方法をオプションとして推奨する。

- (検知方法1) 応答側の MEP 実行系レベル、もしくはその上位レベルである業務ユニットでは、受信したメッセージの意味的内容について、実際の処理を実施する前段階で、その妥当性をチェックする。意味的な内容とは、形式的な内容である XML の構文ではなく、メッセージに含まれる値の範囲等の、意味に関連した事項である。

(3) 業務ユニットレベルで障害を検知したケース

このケースでは、開始側の業務ユニット以下に問題が含まれるケース、応答側の MEP 実行系レベルで問題が含まれるケースの2つがある。応答側の MEP 実行系レベルで問題ない場合、即時対応処理として、共通ヘッダ内のシステムエラー報告の値に業務ユニットレベルの障害を示す値をセットした応答メッセージを開始側に送信する。これに対して、応答側の MEP 実行系レベルで問題があり、送信が不可能である場合は、即時対応処理は行わず、開始側の業務ユニットレベルで、受信タイムアウトが発生・検知させる。

(4) MEP 処理系レベルで障害を検知したケース

業務ユニットレベルが利用する MEP 処理系で障害を検知した場合は、即時対応処理として、その MEP 処理系固有の方法で可能な限り障害を示す受領 Ack、応答メッセージを開始側に返す。

6. 4 プラットフォーム通信機能における異常系処理

異常系処理とは障害検知、即時対応処理を受けて、障害を解決するための一連の処理である。地域情報 PF において、現版の規定として定義される異常系処理は、基本的には、ビジネスプロセスの停止・中断とする。現版の規定下では、トランザクション等の問題解決を図るための規定事項が確定できる段階にない。この状況でビジネスプロセスに関連した問題解決を自動化させると、各実装の固有方式に依存

することになり、データ管理上の不整合を不必要に招くことも考えられる。

そこで現版の規定下では、異常系処理全体において問題解決のプロセスそのものは、基本的にはマニュアル操作で行うことを取り決め事項とする。そして、マニュアル操作を容易にするため、異常を検知した状況そのものを保存することを目的としてビジネスプロセスの停止・中断に留めるものとする。ただし、一部の例外規定も設ける。以下の規定事項を記す。

【共通】

- (1) 即時対応処理により、開始側の業務ユニットに障害発生した旨が報告されなければならない。
- (2) 障害が発生したビジネスプロセスは、例外規定の条件を満足しない限り、即時対応処理の後、停止・中断しなければならない。
- (3) 現版仕様の範囲内で、例外規定の条件を満足しない限り、障害発生に対して、自動化された補償・回復処理を行ってはならない。
- (4) 現版仕様の範囲内で、例外規定の条件を満足しない限り、障害発生したビジネスプロセスの扱い（問題解決、再開、終了等）はシステム運用管理者がマニュアル操作で対応しなければならない。
- (5) 停止させたビジネスプロセスは、監査の結果、続行しても処理上の不整合が発生しないことをシステム運用管理者が確認した場合のみ、再開できる。
- (6) オプションとして定義されるモニタリング機能等の利用が可能である場合、障害状況をログに出力することを推奨する。

【サイト内】

- (7) 例外規定として、自治体内の同サイト間通信であり、同じ運用ポリシー下にあり、異常系処理が一切の不整合を発生させない保証がある場合に限り、自動化された補償・回復処理を行ってもよい。

異常系処理における問題解決等については、ガイドラインに記載する。

6. 5 データ交換システムパターンと、メッセージ交換パターンの関係

2. 3. 2 (3) にデータ交換システムパターンの[Type1]～[Type5]迄、5種類が定義されている。このデータ交換システムパターンは複数のメッセージ交換パターンを組み合わせることで、実現される。しかし、流通データの保全性の関係から、各々のデータ交換システムパターンにて採用できるメッセージ交換パターンについては制約が存在している。ここでは、各データ交換システムパターンを構成する際に採用すべきメッセージ交換パターンについて規定する。

6. 5. 1 データ交換システムパターン[Type1]

[Type1]は、MEP 処理系を介さずに、直接、業務ユニット間でメッセージ交換をするもので、1つのメッセージ交換パターン実施 1-1 から構成される。

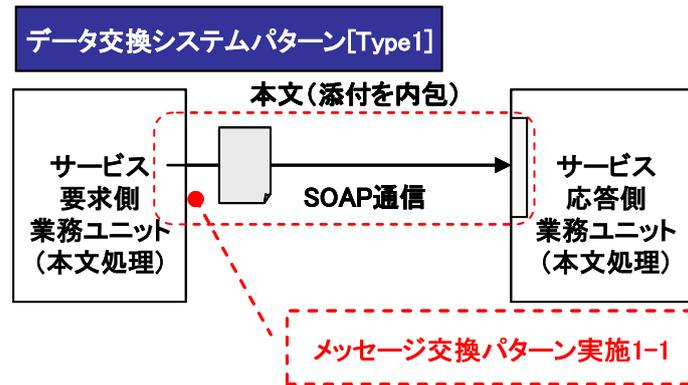


図 6. 8 データ交換システムパターン[Type1]を構成するメッセージ交換パターン

【共通】

- (1.1) メッセージ交換パターン実施 1-1 に関して、下記の規定を定義する。
- サービス応答側業務ユニットからの応答メッセージが不要な場合は、「リクエスト同期型受領 Ack あり」を採用する。
 - サービス応答側業務ユニットからの応答メッセージが必要な場合は、「リクエスト・レスポンス型同期型レスポンス」、もしくは「リクエスト・レスポンス型同期型受領 Ack+非同期型レスポンス」を採用する。後者の場合は「リクエスト同期型受領 Ack あり」を双方向で実施することになる。

6. 5. 2 データ交換システムパターン[Type2]

[Type2]も、MEP 処理系を介さずに、直接、業務ユニット間でメッセージ交換をするもので、1つのメッセージ交換パターン実施 2-1 から構成される。その概要は、[Type1] と概ね同じである。

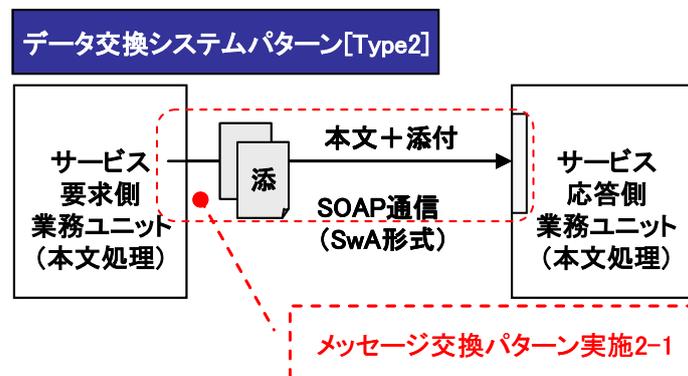


図 6. 9 データ交換システムパターン[Type2]を構成するメッセージ交換パターン

【共通】

- (2.1) メッセージ交換パターン実施 2-1 に関して、下記の規定を定義する。
- サービス応答側業務ユニットからの応答メッセージが不要な場合は、「リクエスト同期型受領 Ack あり」を採用する。
 - サービス応答側業務ユニットからの応答メッセージが必要な場合は、「リクエスト・レスポンス型同期型レスポンス」、もしくは「リクエスト・レスポンス型同期型受領 Ack+非同期型レスポンス」を採用する。後者の場合は「リクエスト同期型受領 Ack あり」を双方向で実施することになる。

6. 5. 3 データ交換システムパターン[Type3]

[Type3]は、直接、業務ユニット間でメッセージ交換をするが、データの受け渡しは、統合 DB 機能を介して行うもので、2つのメッセージ交換パターン実施 3-1、3-2 から構成される。なお、統合 DB 機能へのデータ提供は仕様規定外のため、メッセージ交換パターン実施には含まれない。

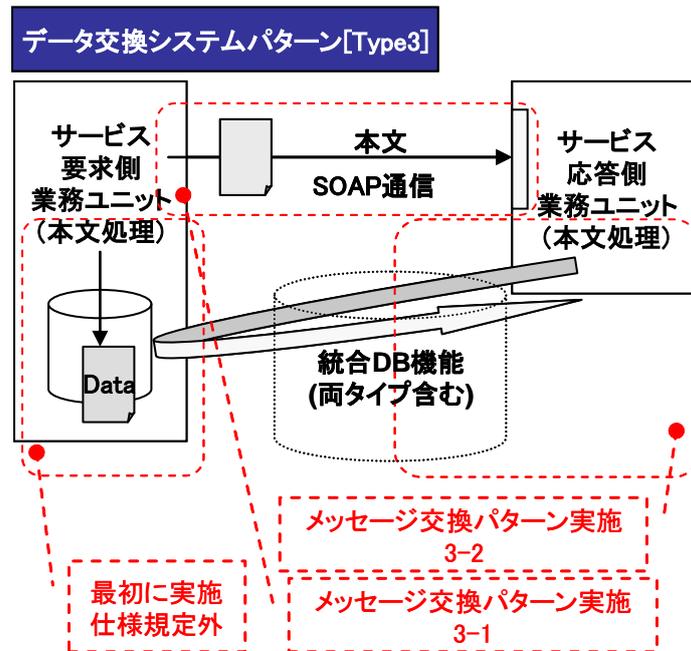


図6. 10 データ交換システムパターン[Type3]を構成するメッセージ交換パターン

【共通】

- (3.1) メッセージ交換パターン実施 3-1 に関して、下記の規定を定義する。
- サービス要求側業務ユニットは、サービス応答側業務ユニットからの応答（受領 Ack またはレスポンス）を確認するまでは、統合 DB 機能に提供したデータ（元データおよび統合 DB 機能のデータ）を更新・変更しないことを保障する必要がある。
 - サービス応答側業務ユニットからの応答を必須とする。
 - 「リクエスト・レスポンス型同期型レスポンス」を採用した場合、サービス応答側業務ユニットは、メッセージ交換パターン実施 3-2 からの応答を確認後に同期型の応答メッセージをサービス要求側業務ユニットに回答しなければならない。
 - 「リクエスト・レスポンス型同期型受領 Ack+非同期型レスポンス」を採用した場合、サービス応答側業務ユニットは、メッセージ交換パターン実施 3-2 からの応答を確認後に非同期側の応答メッセージをサービス要求側業務ユニットに回答しなければならない。
- (3.2) メッセージ交換パターン実施 3-2 に関して、下記の規定を定義する。
- 「リクエスト・レスポンス型同期型レスポンス」のみを採用できる。

6. 5. 4 データ交換システムパターン[Type4]

[Type4]は、BPM 機能を介して、業務ユニット間でメッセージ交換をするもので、2つのメッセージ交換パターン実施 4-1、4-2 から構成される。

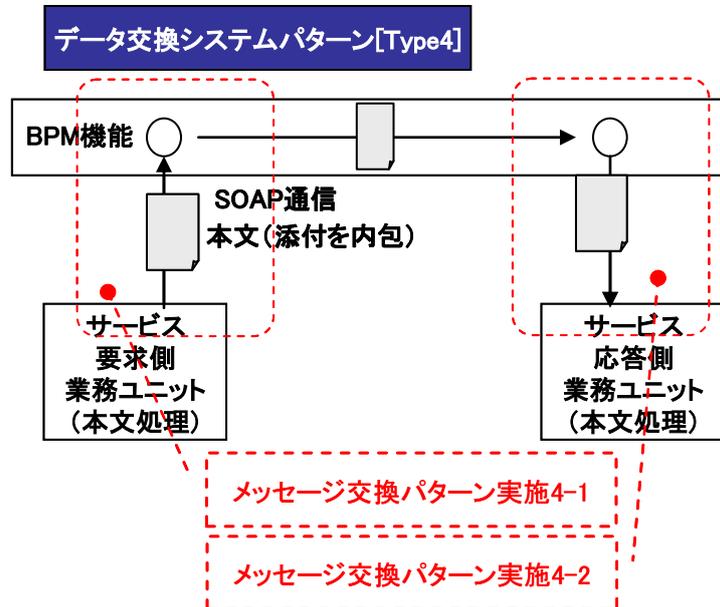


図 6. 1 1 データ交換システムパターン[Type4]を構成するメッセージ交換パターン

【共通】

(4.1) メッセージ交換パターン実施 4-1 に関して、下記の規定を定義する。

- サービス応答側業務ユニットからの応答メッセージが不要な場合は、「リクエスト型同期型受領 Ack あり」を採用する。
- サービス応答側業務ユニットからの応答メッセージが必要な場合は、「リクエスト・レスポンス型同期型レスポンス」、もしくは「リクエスト・レスポンス型同期型受領 Ack+非同期型レスポンス」を採用する。後者の場合は「リクエスト型同期型受領 Ack あり」を双方向で実施することになる。
- 「リクエスト・レスポンス型同期型レスポンス」を採用した場合、サービス応答側業務ユニットは、メッセージ交換パターン実施 4-2 からの応答を、BPM 機能による転送を通して、確認後に同期型レスポンスをサービス要求側業務ユニットに回答しなければならない。
- 「リクエスト・レスポンス型同期型受領 Ack+非同期型レスポンス」を採用した場合、サービス応答側業務ユニットは、メッセージ交換パターン実施 4-2 からの応答を、BPM 機能による転送を通して、確認後に非同期側レスポンスをサービス要求側業務ユニットに回答しなければならない。

(4.2) メッセージ交換パターン実施 4-2 に関して、下記の規定を定義する。

- サービス応答側業務ユニットからの応答メッセージが不要な場合は、「リクエスト型同期型受領 Ack あり」を採用する。
- サービス応答側業務ユニットからの応答メッセージが必要な場合は、「リクエスト・レスポンス型同期型レスポンス」、もしくは「リクエスト・レスポンス型同期型受領 Ack+非同期型レスポンス」を採用する。後者の場合は「リクエスト型同期型受領 Ack あり」を双方向で実施することになる。なお、どちらを選択するかには当たっては、メッセージ交換パターン実施 4-1 と同じ種類のものを選択しなければならない。詳しくは、ガイドラインに記す。

6. 5. 5 データ交換システムパターン [Type5]

[Type5]は、BPM 機能を介して、業務ユニット間でメッセージ交換をするが、データの受け渡しは、統合 DB 機能を介して行うもので、3つのメッセージ交換パターン実施 5-1、5-2、5-3 から構成される。なお、統合 DB 機能へのデータ提供は仕様規定外のため、メッセージ交換パターン実施には含まれない。

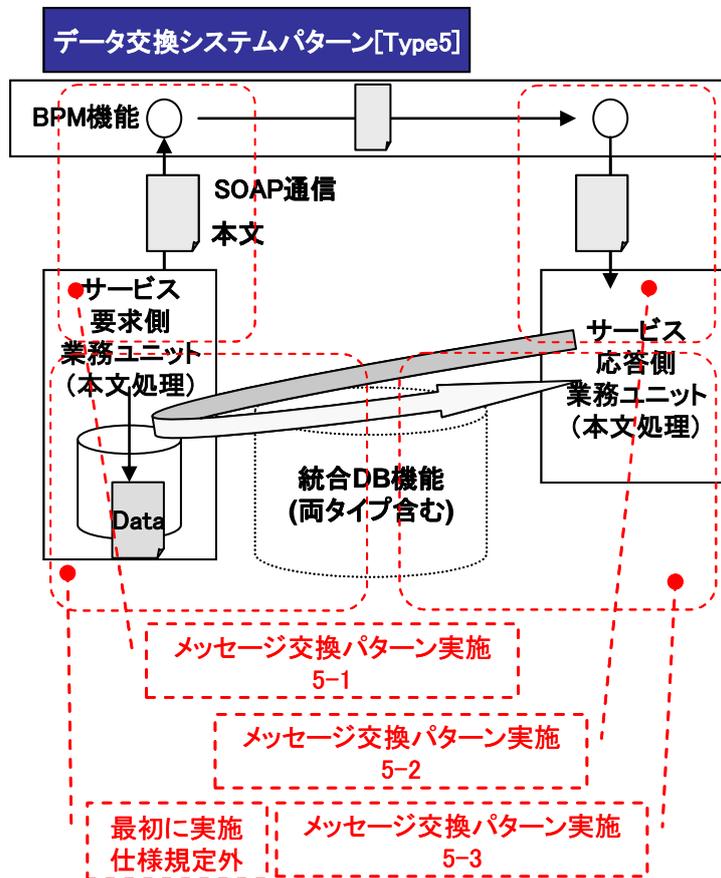


図6. 12 データ交換システムパターン [Type5] を構成するメッセージ交換パターン

【共通】

- (5.1) メッセージ交換パターン実施 5-1 に関して、下記の規定を定義する。
- サービス要求側業務ユニットは、サービス応答側業務ユニットからの応答（受領 Ack、または応答メッセージ）を確認するまでは、統合 DB 機能に提供したデータ（元データおよび統合 DB 機能のデータ）を更新・変更しないことを保障する必要がある。
 - サービス応答側業務ユニットからの応答を必須とする。
 - 「リクエスト・レスポンス型同期型レスポンス」を採用した場合、サービス応答側業務ユニットは、メッセージ交換パターン実施 5-2 からの応答を確認後に同期型の応答メッセージをサービス要求側業務ユニットに回答しなければならない。
 - 「リクエスト・レスポンス型同期型受領 Ack+非同期型レスポンス」を採用した場合、サービス応答側業務ユニットは、メッセージ交換パターン実施 5-2 からの応答を確認後に非同期側の応答メッセージをサービス要求側業務ユニットに回答しなければならない。
- (5.2) メッセージ交換パターン実施 5-2 に関して、下記の規定を定義する。
- サービス応答側業務ユニットからの応答メッセージが不要な場合、「リクエスト型同期型受領 Ack あ

- り」を採用する。ただし、受領 Ack は、メッセージ交換パターン実施 5-3 が終了してから送信する。
- サービス応答側業務ユニットからの応答メッセージが必要な場合は、「リクエスト・レスポンス型同期型レスポンス」、もしくは「リクエスト・レスポンス型同期型受領 Ack+非同期型レスポンス」を採用する。後者の場合は「リクエスト型同期型受領 Ack あり」を双方向で実施することになる。なお、どちらを選択するかに当たっては、メッセージ交換パターン実施 5-1 と同じ種類のものを選択しなければならない。詳しくはガイドラインに記す。

(5.3) メッセージ交換パターン実施 5-3 に関して、下記の規定を定義する。

- 「リクエスト・レスポンス型同期型レスポンス」のみを採用できる。

6. 6 その他の規定事項

6. 6. 1 受領 Ack のメッセージ形式

【共通】

受領 Ack (ack) のメッセージ形式は、表 6. 1 に従う。これは、共通ヘッダと、リクエストやレスポンスを受領できたかの成否のための受領ステータス、受領日時から構成される。

表 6. 1 受領 Ack のメッセージの形式

No.	データ項目	データ型	最小文字数	最大文字数	最小出現回数	最大出現回数	説明
1	受領 Ack	受領 Ack 情報					この項目の最小出現回数、最大出現回数は、取込むスキーマで定義する。
2	共通ヘッダ	共通ヘッダ情報			1	1	「7.1 共通ヘッダ」のデータ項目
3	受領ステータス	string	1	1	1	1	"0"..成功/"1"..異常有り
4	受領日時				1	1	受領日時を入れる
	年	string	4	4	0	1	西暦年、半角数字 4 桁 (※1) (※2)
	月	string	2	2	0	1	半角数字 2 桁 (※1) (※2)
	日	string	2	2	0	1	半角数字 2 桁 (※1) (※2)
	時	string	2	2	0	1	半角数字 2 桁 (※1) (※2)
	分	string	2	2	0	1	半角数字 2 桁 (※1) (※2)
	秒	string	2	2	0	1	半角数字 2 桁 (※1) (※2)
5	備考	string	0	1024	0	1	備考情報を入れる

(※1) 受領日時の桁数は、右詰残り前を 0 で埋める形式とする。

例> 1 月の場合、<月>△1</月> (△ : 空白) → <月>01</月>

(※2) 受領日時として記録する必要のある日時情報の粒度がユースケースや業務に依存するため、年・月・日・時・分・秒といった項目の値は原則オプションで省略可能とする。また、値が無い場合の XML メッセージの表現は、本仕様の 3. 2. (6) の「表 3. 2. 2 データ項目の値が省略された場合の XML 表現方法」の「No. 1 タグを省略する」とする。

<<準拠ルール>>

【共通】

- (CS-R060001) : 「リクエスト型受領 Ack あり」を使用してもよい。(オプション) [6. 2. 1]
- (CS-R060002) : 「リクエスト・レスポンス型同期型レスポンス」を使用してもよい。(オプション) [6. 2. 1]
- (CS-R060003) : 「リクエスト・レスポンス型同期型受領 Ack+非同期型レスポンス」を使用してもよい。(オプション) [6. 2. 1]
- (CS-R060004) : 上記 3 つ以外のメッセージ交換パターンは使用してはならない。(必須) [6. 2. 1]
- (CS-R060005) : メッセージ送信側の MEP 処理系は、下位レベルで検知した障害が通知される。(必須) [6. 3. 3]
- (CS-R060006) : 開始側の MEP 実行系で必須のレスポンス、受領 Ack を受信した際、共通ヘッダ部内のシステムエラー報告の値を検査しなければならない。(必須) [6. 3. 7]
- (CS-R060007) : 開始側の業務ユニットで、応答に対するタイムアウトを検出しなければならない。(必須) [6. 3. 7]
- (CS-R060008) : 応答側の MEP 実行系、業務ユニットでは、受信メッセージの意味的内容を実処理の前段階で検査してもよい。(オプション) [6. 3. 8]
- (CS-R060009) : すべての即時対応処理により、開始側の業務ユニットに障害報告がされなければならない。(必須) [6. 4]
- (CS-R060010) : 障害発生ビジネスプロセスは、例外規定の条件を満足しない限り、即時対応処理の後、停止・中断しなければならない。(必須) [6. 4]
- (CS-R060011) : 例外規定の条件を満足しない限り、障害発生に対して自動化された補償・回復処理を行ってはならない。(必須) [6. 4]
- (CS-R060012) : 例外規定の条件を満足しない限り、障害発生したビジネスプロセスはシステム運用管理者がマニュアル操作で対応決定しなければならない。(必須) [6. 4]
- (CS-R060013) : 停止ビジネスプロセスは、監査し不整合が発生しないことを確認した場合のみ、再開しなければならない。(必須) [6. 4]
- (CS-R060014) : モニタリング機能へ、障害状況をログ出力する。(推奨) [6. 4]
- (CS-R060015) : データ交換システムパターン [Type1] では、規定メッセージ交換パターンのみ採用しなければならない。(必須) [6. 5]
- (CS-R060016) : データ交換システムパターン [Type2] では、規定メッセージ交換パターンのみ採用しなければならない。(必須) [6. 5]
- (CS-R060017) : データ交換システムパターン [Type3] では、規定メッセージ交換パターンのみ採用しなければならない。(必須) [6. 5]
- (CS-R060018) : データ交換システムパターン [Type4] では、規定メッセージ交換パターンのみ採用しなければならない。(必須) [6. 5]
- (CS-R060019) : データ交換システムパターン [Type5] では、規定メッセージ交換パターンのみ採用しなければならない。(必須) [6. 5]

【サイト内】

- (CS-R060020) : 同サイト間通信、且つ、同運用ポリシーの場合、異常系処理が不整合を発生させない場合に限り、自動化された補償・回復処理を行ってもよい。(オプション) [6. 4]

用語の定義

項番	用語	意味
1	メッセージ交換パターン (MEP : Message Exchange Patterns)	メッセージ群の交換の類型を定義するもの。
2	MEP 処理系	BPM 処理系、WS-BPEL 処理系等の狭義の処理系のこと。
3	MEP 実行系	MEP を処理するソフトウェアの総称のこと。前述 MEP 処理系が中心となるが、MEP 処理系を利用する業務ユニットレベルのものも含め、広義に定義する。
4	エンティティ	OSI の定義に基づき、メッセージを扱うもの。具体的には、サービス、プロセスを意味する。
5	リクエスト (req)、リクエスト (req) メッセージ、要求メッセージ	業務依頼を意味する要求メッセージであり、地域情報プラットフォームにおける MEP 開始の送信メッセージ。
6	レスポンス (res)、レスポンス (res) メッセージ、応答メッセージ	リクエストに対する業務処理結果を返す際のメッセージ。
7	受領 Ack (ack)	業務プロセス、もしくは業務サービスでリクエスト、レスポンスを受信したことを送信元に通知するためのメッセージ。
8	開始側 (initiator) のエンティティ、開始側	リクエストを送信するサービス、あるいは BPM プロセスのことで、この仕様では、「発呼側」と呼ぶこともある。
9	応答側 (responder) のエンティティ、応答側	リクエストを受信し、必要に応じてレスポンスを送信するサービス、BPM プロセス、あるいは、統合 DB 機能のことで、この仕様では「着呼側」と呼ぶこともある。
10	メッセージ送信側	SOAP 通信レベル (高信頼性通信の場合は、そのレベル) から下位レベルでメッセージを送信する側のことで、上記「開始側」とは異なる。
11	メッセージ受信側	SOAP 通信レベル (高信頼性通信の場合は、そのレベル) から下位のレベルでメッセージを受信する側のことで、上記「応答側」とは異なる。
12	リクエスト型	リクエスト (req) のメッセージだけ、もしくはリクエスト (req) と受領 Ack (ack) のメッセージだけで、それに対する応答が無いもの。
13	リクエスト・レスポンス型	リクエスト (req) のメッセージに対するレスポンス (res) のメッセージが存在するもの。ただし、受領 Ack (ack) はレスポンス (res) のメッセージとは扱わない。
14	同期	受信内容に対する返信 (受領 Ack、レスポンス) が、受信時と同一の HTTP セッションで返される通信パターン。
15	非同期	受信内容に対する返信 (受領 Ack、レスポンス) が、受信時と異なる別の HTTP セッションで返される通信パターン。
16	障害検知	異常を捕捉すること。
17	即時対応処理	障害検知を受け、それを通知する等の問題解決を伴わない対応処理。
18	異常系処理	障害検知、即時対応処理を受けて、障害を解決するための一連の処理。

19	受信タイムアウト	規定時間内に開始側においてレスポンス(res)メッセージを受け取れないこと。
----	----------	--

7. プラットフォーム通信仕様におけるメッセージ共通ヘッダ仕様

本章では、PF 通信で採用する電子封筒形式のうち、ビジネスメッセージ用で利用される共通ヘッダ仕様を規定する。共通ヘッダ仕様は、地域ポータル等、サイト間でのビジネスプロセス制御を実施する際の、全体制御に必要な仕様を規定している。

【サイト間】

サイト間での PF 通信の際は、メッセージに本章で規定する共通ヘッダを持たせること。

【サイト内】

メッセージの共通ヘッダを持たせることは、自治体内での適用は任意であるが、採用を推奨する。

7. 1 共通ヘッダで規定するデータ項目

PF 通信で交換される電子封筒形式のうち、ビジネスメッセージ用では、SOAP Body に共通ヘッダを、下記の目的に応じて活用するために規定している。

- (a) ビジネスプロセスフローを制御するための項目
- (b) PF 通信の送信制御や通信結果のための項目
- (c) モニタリング用の項目
- (d) 業務ユニットインターフェースの応答メッセージのための項目

ビジネスメッセージ用の電子封筒形式における共通ヘッダの位置づけを図 7. 1 に示す。

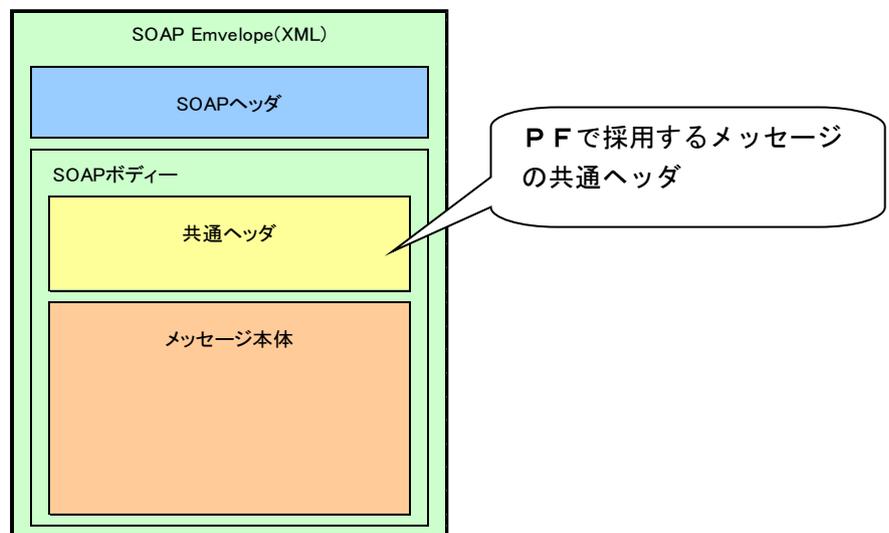


図 7. 1 共通ヘッダの位置づけ

7. 1. 1 共通ヘッダのデータ項目と項目用途

地域情報 PF 仕様において規定している共通ヘッダのデータ項目とその用途説明について、表 7. 1. 2 に示す。共通ヘッダのデータ項目は、項目の値の設定方法としては全てオプション（ユースケース毎に設定されたり、設定されなかったりする）である。

以降に示す「表 7. 1. 2 共通ヘッダのデータ項目説明」の表頭に設定している各項目の意味を、表 7. 1. 1 にて説明する。

表 7. 1. 1 共通ヘッダのデータ項目説明表の読み方

No.	表頭の項目	説明
1	データ項目	共通ヘッダのデータ項目名。XML 定義する際の要素名の候補として使う。
2	データ項目の値の必須・オプション	項目の設定値が必須か、もしくは省略される場合の種別（共通ヘッダとして定義しているデータ項目は、利用するユースケースにより値の設定に関する必須・オプションの位置づけが変わるため、全ての値の項目の設定について原則オプションとする）
3	共通ヘッダの項目を使うユースケース	<p>現状想定している共通ヘッダの項目を扱うユースケースの種別。</p> <p>以下のユースケースで、●がある項目を使用する。</p> <p>a : ビジネスプロセスフローを制御するために使用する項目</p> <p>b : PF 通信の送信制御や通信結果のために使用する項目</p> <p>c : モニタリング用（モニタリング機能を使用する場合）に使用する項目</p> <p>d : 自治体業務ユニットインターフェースの応答メッセージに使用する項目</p>
4	項目説明	<p>項目の説明に関して、次の 4 つの共通観点がある。</p> <p>但し本仕様では、全てのデータ項目について 4 つの観点のそれぞれを標準として規定していない。そのため、項目によっては説明がない観点もある。</p> <p>①目的：該当項目領域の利用目的</p> <p>②仕様：項目領域に値が設定されている場合、最低限守るべき仕様</p> <p>③開始側：MEP における開始側にて当該項目領域に対して実行処理を行なう場合、開始主体が最低限守るべき仕様</p> <p>④応答側：MEP における応答側にて、応答メッセージを作成するうえで当該項目に対して実行処理を行なう場合応答主体が最低限守るべき仕様</p>

表 7. 1. 2 共通ヘッダのデータ項目説明表

No.	データ項目	データ項目の値の 必須・ オプション	共通ヘッダの項目を使うユースケース				項目説明
			a	b	c	d	
1	共通ヘッダ						
2	To	オプション		●			<p>目的：送信先のENDPOINTを値として指定するための項目領域。例えば、動的なメッセージルーティングを利用する場合（BMR-GW利用の場合）、送信先の ENDPOINTを指定する。また、MEP非同期型通信の応答側では、開始側が「ReplyTo」領域に指示した戻り先のENDPOINTを値として設定する。</p> <p>仕様①：当該領域にENDPOINTとして指定される値はURLのみである。</p> <p>仕様②：URLは送信先として有効なものでなければならない。</p> <p>開始側：目的の用途をふまえ、動的なメッセージルーティングにより要求メッセージの送信先を指定するために、エンドポイントとなるURLを設定する。</p> <p>応答側①：MEP同期型通信で応答メッセージを作成する場合に、当該項目領域に値は設定しない。</p> <p>応答側②：MEP非同期型通信で応答メッセージを作成する場合に、開始側が要求メッセージの「ReplyTo」に指示した戻り先のENDPOINTを当該項目領域に設定する。応答側の処理結果を、開始側から指示された「戻り先ENDPOINT」に送信する。</p>
3	MsgID	オプション			●		<p>目的：開始側エンティティや応答側エンティティにおいて生成されるメッセージを個別に識別するため、メッセージ毎に固有に付与する値を設定する項目領域。付与した値は、メッセージのログ取得やモニタリングのための識別子として使う。</p> <p>仕様：識別できるユニークなIDで、要求メッセージと応答メッセージ、受領Ackでも異なる値とする。</p>
4	RelatesTo	オプション			●		<p>目的：モニタリングやエラー分析上、メッセージ間の関連性を示すために付与する値を設定する項目領域。</p> <p>仕様：要求メッセージの「MsgID」に指定された値とする。</p> <p>開始側：当該項目領域に値は設定しない。</p> <p>応答側：応答側が応答メッセージを作成する場合、要求メッセージの「MsgID」に指定された値を応答メッセージの当該項目領域に複製する。</p>
5	ReplyTo	オプション		●			<p>目的：MEP非同期型通信（かつ動的なメッセージルーティングを利用する）で応答メッセージを作成する場合に、その応答メッセージの送り先ENDPOINTの値を設定する項目領域。</p> <p>仕様①：当該領域にENDPOINTとして指定される値はURLのみである。</p> <p>仕様②：URLは送信先として有効なものでなければならない。</p> <p>開始側：MEP非同期型通信で要求メッセージを作成する場合、応答側の処理結果を送付する（戻り先）ENDPOINTを指定する。</p> <p>応答側：MEP同期型通信でもMEP非同期型通信でも、応答側は当該項目領域に何もしない。</p>
6	受付番号	オプション	●	●	●	●	<p>目的①：サービス処理申請における伝票番号等、業務処理（ビジネスプロセス管理）の単位毎に付与される識別子を設定する項目領域。</p> <p>目的②：案件番号として使い、全ての通信の処理で共通の案件情報として引き回される値を設定する項目領域。</p> <p>目的③：モニタリングのためにグローバルユニークとなる値を設定する項目領域。</p> <p>仕様①：サイト間でもサイト内でも一定のサービス処理範囲を定義し、その中でグローバル・ユニークな値が設定される。</p> <p>※注釈「グローバル・ユニークな値」とは、重複がなく固有であり変更・更新が行なわれることのない値という意味。</p> <p>仕様②：業務モデル標準での要件に応じて値が設定される。</p> <p>開始側：目的②を踏まえて当該項目領域に値を設定する場合に、処理の開始側として定義された主体はビジネスプロセス管理で規定される体系で識別子を付与する。</p> <p>応答側：要求メッセージの当該項目領域の値を応答メッセージに複製する。</p>
7	共通コリレーションセット	オプション	●				<p>目的：ビジネスプロセス管理の対象メッセージに、BPインスタンス識別情報を指定する項目領域。（ビジネスプロセス管理の対象メッセージに必須（本文の業務項目をコリレーションセットとする場合は不要））</p> <p>※注釈「コリレーションセット」とは「BPELでは、あるプロセスインスタンスに関連付けられているキーデータの値の集合」即ち、「本文の業務項目をコリレーションセットとする場合は不要」とは「メッセージ本文の業務項目に関連付けられたメッセージでは、業務番号（利用業務ユニット番号や識別番号の組み合わせ）で識別できるので本項目は不要」という意味。</p> <p>仕様：WS-BPELでのコリレーションセットの値が設定される。</p> <p>開始側：BPM機能独自の共通コリレーションセットの値を設定する。受付番号の値を複製して設定してもよい。</p> <p>応答側：メッセージの当該項目領域の値を応答メッセージに複製する。</p>

No.	データ項目	データ項目の値の 必須・ オプション	共通ヘッダの項目を使うユースケース				項目説明
			a	b	c	d	
8	ビジネスプロセス制御情報	オプション	●				<p>目的: ビジネスプロセス呼び出し側からフロー制御（本文（申請データ等）でフロー制御する場合は不要）するための値を設定する項目領域。</p> <p>仕様: ビジネスプロセス毎にフロー制御するコード体系を決定し、値を設定する。</p> <p>開始側: BPM機能に対して要求メッセージを作成する開始側の主体は、ビジネスプロセスの分岐を制御するための値を設定する。</p> <p>応答側: 処理アプリは本項目を設定しない。</p>
9	業務サービス結果情報	オプション	●			●	<p>目的: 業務サービスの処理ステータスを示す値を設定する項目領域。</p> <p>仕様①: 業務ユニットインタフェースや業務サービスインタフェースにおける値の詳細な設定方法は、業務モデル標準で規定する。</p> <p>仕様②: 統合DB機能における値の詳細な設定方法は、アーキテクチャ標準仕様の統合DB機能で規定する。</p>
10	結果情報	オプション				●	<p>目的: システム的なレスポンスの状態を示す値を設定する項目領域。</p> <p>仕様①: システム的なレスポンスが正常の場合（0）を設定する。</p> <p>仕様②: システム的なレスポンスが異常の場合（1）を設定する。</p> <p>開始側: MEP同期型通信および非同期通信において、開始側は当該領域に値を設定しない。</p> <p>応答側: MEP同期型通信および非同期通信において、応答側は値を設定する。</p>
11	システムエラー報告	オプション				●	<p>目的: システム的なレスポンスの状態が異常（1）の場合に、その異常ステータスの詳細を示す値を設定する項目領域。</p> <p>仕様①: 項目No. 10に（1）が設定された際に、この領域にエラーの詳細状態を示す値を設定する。</p> <p>仕様②: この領域に示すエラーの詳細状態を値をコード化する場合は、個別に決定し適用する。</p> <p>開始側: MEP同期型通信および非同期通信において、開始側は当該領域に値を設定しない。</p> <p>応答側: MEP同期型通信および非同期通信において、応答側は値を設定する。</p>

7. 1. 2 共通ヘッダのデータ項目のXMLスキーマ作成仕様

前述した共通ヘッダのデータ項目についてXMLスキーマを作成する際の仕様を本節にて示す。
 主な留意事項は次のとおり。

- ①. 共通ヘッダのXMLスキーマをプラットフォーム通信標準仕様として公開する。
 共通ヘッダを活用する場合は、公開したスキーマを適用することとする。
- ②. 表7. 1. 2に示したとおり、共通ヘッダのデータ項目の値の設定方法は、利用するユースケースにより要・不要が判断される。従って、ユースケースにより項目を使わない場合には値が省略されるため、項目の値の必須・オプションについては、原則オプションである。項目の値が省略されている場合、受付番号のXMLタグは必須のため、空タグ（本仕様の3. 2. (6)の「表3. 2. 2 データ項目の値が省略された場合のXML表現方法」のNo.3またはNo.4)で表現する。また、受付番号以外のXMLタグはオプションのため、値が省略された場合、XMLタグを省略（本仕様の3. 2. (6)の「表3. 2. 2 データ項目の値が省略された場合のXML表現方法」のNo.1)する。

表7. 1. 3 共通ヘッダのデータ項目のXMLスキーマ作成仕様

No.	XML要素名	XMLスキーマ仕様					
		XMLのデータ型	最小の文字数	最大の文字数	XMLタグの必須・オプション	最小出現回数	最大出現回数
1	共通ヘッダ	共通ヘッダ情報					
2	To	string	0	1024	オプション	0	1
3	MsgID	string	0	1024	オプション	0	1
4	RelatesTo	string	0	1024	オプション	0	1
5	ReplyTo	string	0	1024	オプション	0	1
6	受付番号	string	0	256	必須	1	1
7	共通コリレーションセット	string	0	256	オプション	0	1
8	ビジネスプロセス制御情報	string	0	100	オプション	0	1
9	業務サービス結果情報	string	0	10	オプション	0	1
10	結果情報	string	0	1	オプション	0	1
11	システムエラー報告	string	0	1024	オプション	0	1

7. 2 共通ヘッダの各項目の設定方法

7. 2. 1 共通ヘッダのデータ項目の値の設定

(1) ビジネスメッセージ処理を非同期に実施する場合の共通ヘッダの使い方

ビジネスメッセージ処理を非同期（6. で定義されたメッセージ交換パターン定義の中のリクエスト型受領 Ack ありや、リクエスト・レスポンス型同期型受領 Ack+非同期型レスポンスの場合）に実施する場合の共通ヘッダの使い方を、図 7.2 に示す。

ア) 例 1：業務ユニットから BPM 機能呼び出す場合

(1) 要求元の業務ユニット側の設定

- ・<受付番号>は、要求元の業務ユニットが採番した受付番号の値を設定。
- ・<共通コリレーションセット>は、受付番号と同様の値を設定（サービス実行BPM機能のBPインスタンスの識別用）

(2) サービス実行BPM機能側の設定

- ・<受付番号>は、要求元の業務ユニット側で設定された値を引き継ぐ。
- ・<業務サービス結果情報>に、サービス実行BPM機能側の業務的な処理結果を設定。
- ・<結果情報>に、システムのサービス呼び出し処理が正常(0)終了したか異常(1)かの結果情報を設定。異常の場合、<システムエラー報告>にエラーコードを設定。

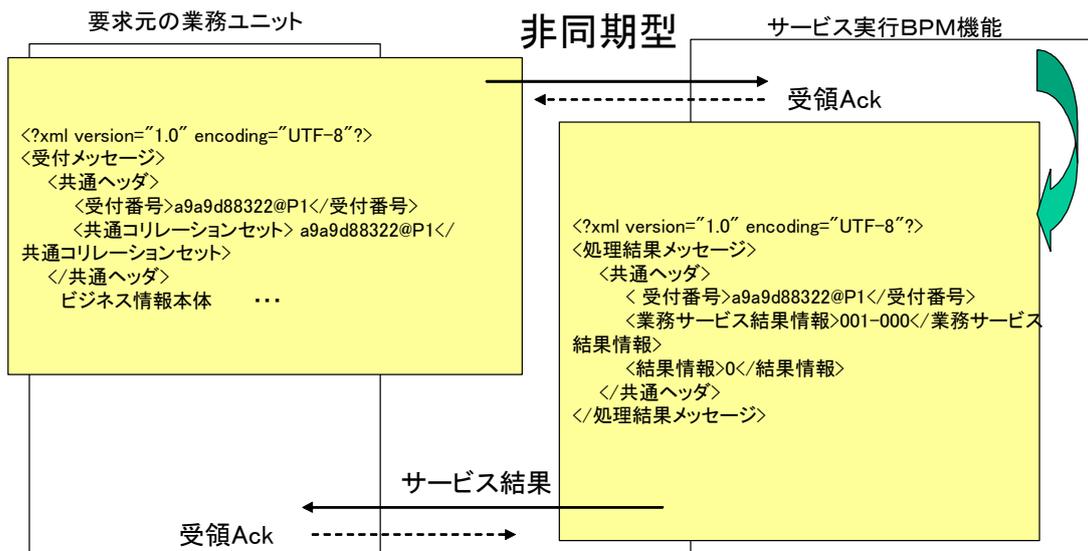


図 7. 2 共通ヘッダの使い方（業務ユニットから BPM 機能へ呼び出す場合）

イ) 例 2 : BPM 機能から人による審査がある業務ユニットを呼び出す場合

- (1) 要求元のBPM機能側の設定
 - ・<ReplyTo>に、非同期でのサービス実行ユニットのサービス結果を戻すENDPOINTを設定。
 - ・<受付番号>は、要求元のBPM機能呼び出したユニットが設定した値をそのまま引き継ぐ。
- (2) サービス実行ユニット側の設定
 - ・<業務サービス結果情報>に、サービス実行ユニットの業務的な処理結果を入れる。
 - ・<結果情報>に、システム的なサービスの呼び出し処理が正常(0)終了したか異常(1)かの結果情報を設定。異常の場合、<システムエラー報告>にエラーコードを設定。
 - ・<共通コリレーションセット>に、要求元のBPM機能の同一インスタンスを識別するために、要求元のBPM機能側で設定された値を引き継ぎ返す。
 - ・<To>に、要求元のBPM側で<ReplyTo>に設定された値を引き継ぐ。

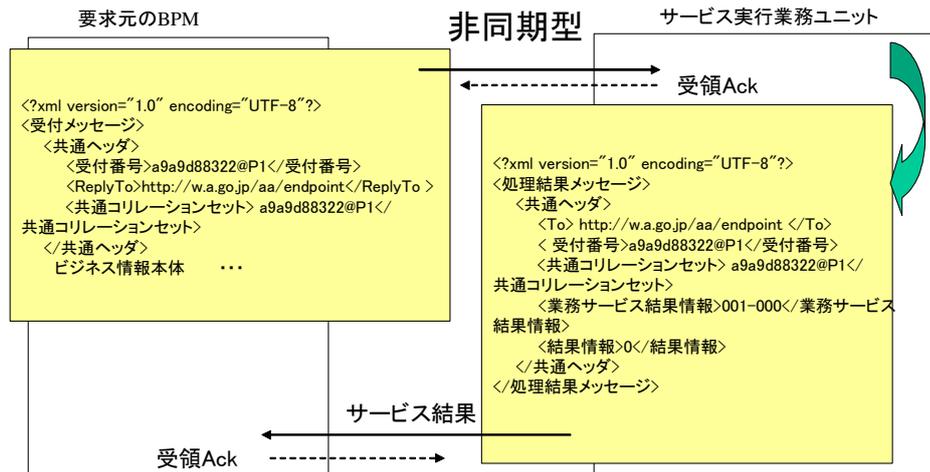


図 7. 3 共通ヘッダの使い方 (BPM 機能から人による審査がある業務ユニットを呼び出す場合)

ウ) 例 3 : BPM 機能から自動処理応答のある業務ユニットを呼び出す場合

- (1) 要求元のBPM機能の設定
 - ・<受付番号>は、一連の業務処理の延長でサービス通信する際は、値をそのまま引き継ぐ。
 - ・業務ユニットからのワンショットの検索では、受付番号は空値とする。
- (2) サービス実行ユニットの設定
 - ・<受付番号>と<共通コリレーションセット>の値は、要求元のBPM機能側で設定された値を引き継ぎ返す。
 - ・<結果情報>は、システム的なサービスの呼び出し処理が正常(0)終了したか異常(1)かの結果情報を設定して返す。異常の場合、<システムエラー報告>にエラーコードを入れる。
 - ・<業務サービス結果情報>に、サービス実行ユニットの業務的な処理結果を入れる。

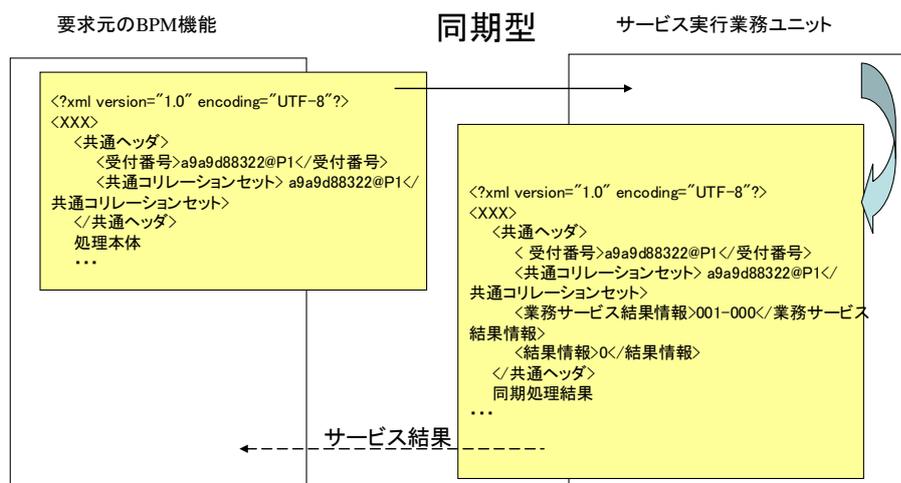


図 7. 4 共通ヘッダの使い方 (BPM 機能から自動処理応答のある業務ユニットを呼び出す場合)

エ) 例 4 : 業務ユニットから自動処理応答のある業務ユニットを呼び出す場合

- (1) 要求元の業務ユニットの設定
 - <受付番号>は、一連の業務処理の延長でサービス通信する際は、値をそのまま引き継ぐ。
 - 業務ユニットからのワンショットの検索では、受付番号は空値とする。
- (2) サービス実行ユニットの設定
 - <受付番号>の値は、要求元の業務ユニット側で設定された値を引継ぎ返す。
 - <結果情報>は、システム的なサービスの呼び出し処理が正常(0)終了したか異常(データ無しを含む)(1)かの結果情報を設定して返す。異常の場合、<システムエラー報告>にエラーコードを入れる。

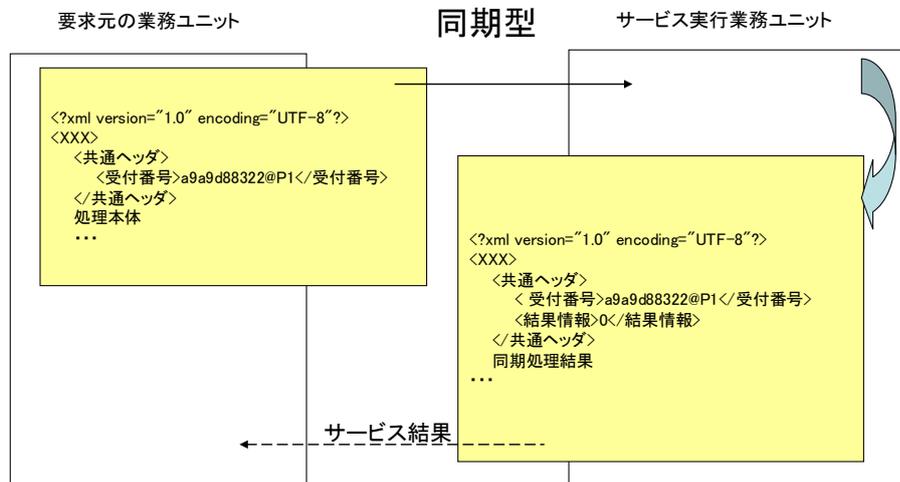


図 7. 5 共通ヘッダの使い方（業務ユニットから自動処理応答のある業務ユニットを呼び出す場合）

7. 2. 2 メッセージ ID (MsgID)

目的：メッセージ ID は、モニタリング機能を利用してビジネスプロセス間、ポータルとビジネスプロセス間、およびビジネスプロセスとサービス間など、ビジネスプロセスを介してやり取りされる業務メッセージを識別するために使用する。この業務メッセージを識別することで、サイト間での進捗状況トレース、メッセージトレース、および障害監視に使用する。

内容：メッセージ ID は、業務メッセージ生成・送信時に長さ 1024 以下のグローバルユニークな文字列で生成する（ただし、ローカル環境のみで使用する場合は、サーバ名等でもよい）。WS-BPEL 仕様の制約がない環境で生成するケースと、WS-BPEL プロセス内部の送信アクティビティにおいて、仕様の範囲内で生成するケースの 2 通りのフォーマットの例を下記に示す。

(1) WS-BPEL 仕様の制約がない環境で生成する場合

（サービス、ポータル、WS-BPEL 拡張機能等での生成）：

西暦(yyyy)、月(MM)、日(dd)、時分秒(hh:mm:ss)、およびミリ秒(zzz)で表される時刻表現と、重複を防ぐための 3 桁から 4 桁の整数で示されるカウンタ値および WS-BPEL 処理系の固有文字列を組み合わせた以下のフォーマットを規定する。カウンタ値は規定桁数の最大値へ到達したら次は 0 に戻すようカウントアップさせる。また、この方法で生成されるメッセージ ID には ‘_’（下線）を含んではならない。“yyyy-MM-dd-hh:mm:ss.zzz” + “-” + カウンタ値 + “@” + 固有文字列（ドメインを含む完全なサーバ名など）

(2) WS-BPEL の仕様の範囲内で生成する場合（WS-BPEL プロセス内での生成）：

WS-BPEL によるプロセスインスタンス開始時に受信したメッセージに含まれるメッセージ ID と

‘_’（下線）を連結した文字列を種 ID として、とその後ろに送信アクティビティ毎の固有文字列を連結させた以下のフォーマットを規定する。

種 ID + 送信アクティビティ固有文字列

利用箇所：要求元では、送信するメッセージの共通ヘッダに MsgID を設定し、要求先では返信のメッセージの共通ヘッダに要求元で設定した MsgID を Related に設定する。

7. 2. 3 受付番号

目的：一般に住民の申請手続きをポータルなどから実行される一連の申請手続き処理は、複数の業務ユニットや BPM 機能と連携して処理される。受付番号は、一連の申請手続き処理を識別するために用いられる。業務処理の実行トレース、障害監視などに使用する。

内容：業務開始時に処理を開始するアプリケーションが、グローバルユニークとなるように生成する。例えば、メッセージ ID の形式を活用する方法が考えられる。

利用箇所：一連の処理の起点となる業務ユニット（ポータル含む）が採番する番号を、共通ヘッダの受付番号に設定する。

7. 2. 4 To、ReplyTo と動的な業務メッセージ送信先の制御

業務メッセージの送信先を、受信した申請書類の内容や業務プロセスの処理結果に応じて動的に変更する要求がある。例えば、ポータルからの情報照会の申請書類から照会先自治体を取り出し、その自治体名に基づいて照会依頼を送信するようなケースが考えられる。あるいは、ある自治体からの業務メッセージの処理結果を後日返信する場合、業務メッセージの送信元に応じた返信先の動的変更が必要となる。地域情報 PF では、この動的な業務メッセージ送信先の制御機能を PF 共通機能「ユーティリティ機能」の「BMR-GW」で実現する。

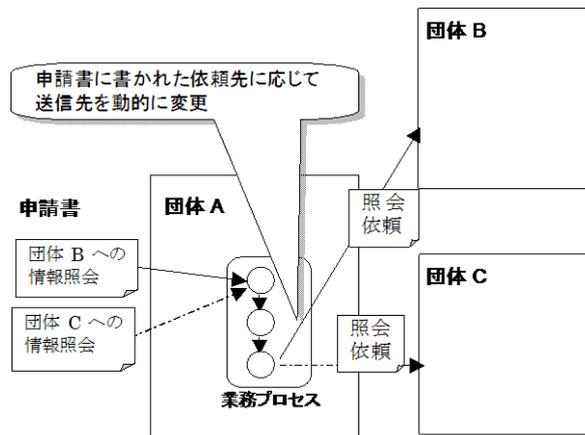


図 7. 6 動的な送信先変更

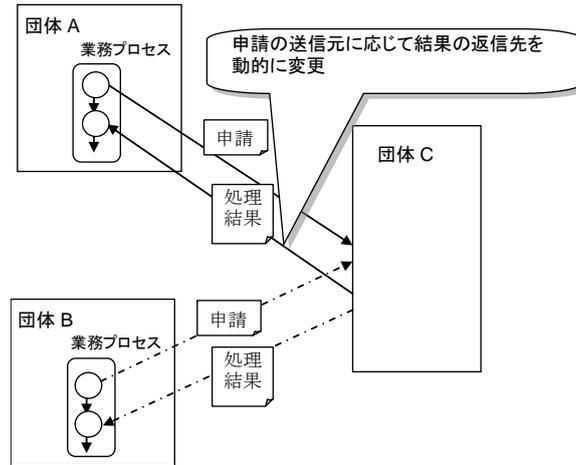


図 7. 7 送信元に応じた返信

また、WS-BPEL で記述されたビジネスプロセスで送信先の動的変更を実現する場合、WS-BPEL 仕様に従って EndpointReference を使用することが考えられる。しかし、ビジネスプロセス配備時に送信先を静的にバインドする WS-BPEL 処理系の場合、こうした機能が使用できない可能性がある。また、一般的な業務ユニットにおいても、複数の呼び出し元から非同期で呼び出される場合や、業務内容により呼び出し先の業務ユニットを動的に変える場合にも BMR-GW を利用する。

なお、送信元に応じた返信先の動的変更のためには、ビジネスメッセージに返信先を記述し、それに応じた返信を行うための規定が必要である。

本節では、上記のような動的な送信先制御のためにビジネスメッセージの共通ヘッダに記述すべき情報と、動的な送信先変更を実現するための方法について規定する。

送信・返信先の指定：

送信先、返信先の指定には、ビジネスメッセージ内の共通ヘッダに含まれる以下の情報を使用する。

To：送信先 URI。動的な送信先指定の場合は必ず設定する。

ReplyTo：この ReplyTo が含まれる業務メッセージの返信先 URI。

非同期による返信が必要な場合は必ず設定する。

また、元メッセージとの対応関係をとるために、返信時には以下の情報を使用する。

RelatesTo：この RelatesTo が含まれる業務メッセージの元メッセージとなる業務メッセージのメッセージ ID (MsgID) を設定する。

MsgID と RelatedTo の BPM 機能を使う場合の関係は、地域情報プラットフォームガイドライン第 3 章を参照。

<< 準拠ルール >>

【サイト間】

(CS-R070001)： サイト間での PF 通信の際は、メッセージに本章で規定する共通ヘッダを持たせること。(必須) [7.1]

【サイト内】

(CS-R070002)： メッセージの共通ヘッダを持たせることは、自治体内での適用は任意であるが、採用を推奨する。(推奨) [7.1]

用語の定義

項番	用語	意味
1	共通ヘッダ	ビジネスメッセージの共通ヘッダ、サービス通信の制御や BP 制御のための情報から構成される。
2	コリレーションセット	BPM 機能において、ビジネスプロセスインスタンスを識別する ID
3	ビジネスメッセージルーティングゲートウェイ (BMR-GW) 機能	メッセージヘッダ部（詳細は、PF 通信標準仕様の「7. プラットフォーム通信仕様におけるメッセージ共通ヘッダ仕様」を参照）に記載された送信先 (To タグ項目の値) に、メッセージ本体を動的に送信代行する Proxy として動作する機能

付録 1. ビジネス電文の例

プラットフォーム通信標準仕様の「3. 2 XML 定義仕様 (5) XML に関する取り決め」の⑥において、「名前空間はルート要素で必ず宣言する」とされている。この規定は XSD については必須であるが、ビジネス電文 (SOAP メッセージ) に対しては必ずしも適用されない。SOAP メッセージにおけるルート要素は SOAP の Envelope 要素となるが、この場合の名前空間の宣言は必ずしも Envelope 要素で行われなくてもよい。

実際のビジネス電文においては、次のような名前空間の宣言パターンがあり得る。

[パターン 1]

全ての名前空間の宣言を Envelope 要素で行い、業務メッセージのタグと共通ヘッダのタグに名前空間接頭辞を付けるパターン

[パターン 2]

業務メッセージと共通ヘッダの名前空間の宣言を SOAP の Body 部で行い、業務メッセージのタグと共通ヘッダのタグにデフォルト Namespace を記述するパターン

[パターン 3]

業務メッセージと共通ヘッダの名前空間の宣言を SOAP の Body 部で行い、業務メッセージのタグにデフォルト Namespace を記述し、共通ヘッダのタグに名前空間接頭辞を付けるパターン

[パターン 4]

業務メッセージと共通ヘッダの名前空間の宣言を SOAP の Body 部で行い、業務メッセージのタグと共通ヘッダのタグに名前空間接頭辞を付けるパターン

以下に、自治体業務におけるこれらのビジネス電文例を示す。

(1) リクエストの例

- ① 全ての名前空間の宣言を Envelope 要素で行い、業務メッセージのタグと共通ヘッダのタグに名前空間接頭辞を付けるパターン

```
POST /spf/services/JutogaiKanriJ HTTP/1.0
Content-Type: text/xml; charset=utf-8
Accept: application/soap+xml, application/dime, multipart/related, text/*
User-Agent: Axis/1.4
Host: XXXXXXX:8001
Cache-Control: no-cache
Pragma: no-cache
SOAPAction: "boss1_1-2010-02_1"
Content-Length: 610
```

```
<?xml version="1.0" encoding="UTF-8"?>
<soap:Envelope
  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:ns0="urn:applic:xmlns:pf:lgxml:schema:2010-02"
  xmlns:ns1="urn:applic:xmlns:pf:common:schema:2010-01">
  <soap:Header/>
```

```

<soap:Body>
  <ns0:識別番号メッセージ>
    <ns1:共通ヘッダ>
      <ns1:受付番号>20100122001</ns1:受付番号>
    </ns1:共通ヘッダ>
    <ns0:利用業務ユニット>02</ns0:利用業務ユニット>
    <ns0:識別番号>19</ns0:識別番号>
  </ns0:識別番号メッセージ>
</soap:Body>
</soap:Envelope>

```

- ② 業務メッセージと共通ヘッダの名前空間の宣言を SOAP の Body 部で行い、業務メッセージのタグと共通ヘッダのタグにデフォルト Namespace を記述するパターン

```

POST /spf/services/JutogaiKanriJ HTTP/1.0
Content-Type: text/xml; charset=utf-8
Accept: application/soap+xml, application/dime, multipart/related, text/*
User-Agent: Axis/1.4
Host: XXXXXX:8001
Cache-Control: no-cache
Pragma: no-cache
SOAPAction: "boss1_1-2010-02_1"
Content-Length: 610

<?xml version="1.0" encoding="UTF-8"?>
<soap:Envelope
  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soap:Header/>
  <soap:Body>
    <識別番号メッセージ xmlns="urn:applic:xmlns:pf:lgxml:schema:2010-02">
      <共通ヘッダ xmlns="urn:applic:xmlns:pf:common:schema:2010-01">
        <受付番号>20100122001</受付番号>
      </共通ヘッダ>
      <利用業務ユニット>02</利用業務ユニット>
      <識別番号>19</識別番号>
    </識別番号メッセージ>
  </soap:Body>
</soap:Envelope>

```

- ③ 業務メッセージと共通ヘッダの名前空間の宣言を SOAP の Body 部で行い、業務メッセージのタグにデフォルト Namespace を記述し、共通ヘッダのタグに名前空間接頭辞を付けるパターン

```

POST /spf/services/JutogaiKanriJ HTTP/1.0
Content-Type: text/xml; charset=utf-8
Accept: application/soap+xml, application/dime, multipart/related, text/*

```

```

User-Agent: Axis/1.4
Host: XXXXXX:8001
Cache-Control: no-cache
Pragma: no-cache
SOAPAction: "boss1_1-2010-02_1"
Content-Length: 610

<?xml version="1.0" encoding="UTF-8"?>
<soap:Envelope
  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soap:Header/>
  <soap:Body>
    <識別番号メッセージ xmlns="urn:applic:xmlns:pf:lgxml:schema:2010-02">
      <ns1:共通ヘッダ xmlns:ns1="urn:applic:xmlns:pf:common:schema:2010-01">
        <ns1:受付番号>20100122001</ns1:受付番号>
      </ns1:共通ヘッダ>
      <利用業務ユニット>02</利用業務ユニット>
      <識別番号>19</識別番号>
    </識別番号メッセージ>
  </soap:Body>
</soap:Envelope>

```

- ④ 業務メッセージと共通ヘッダの名前空間の宣言を SOAP の Body 部で行い、業務メッセージのタグと共通ヘッダのタグに名前空間接頭辞を付けるパターン

```

POST /spf/services/JutogaiKanriJ HTTP/1.0
Content-Type: text/xml; charset=utf-8
Accept: application/soap+xml, application/dime, multipart/related, text/*
User-Agent: Axis/1.4
Host: XXXXXX:8001
Cache-Control: no-cache
Pragma: no-cache
SOAPAction: "boss1_1-2010-02_1"
Content-Length: 610

<?xml version="1.0" encoding="UTF-8"?>
<soap:Envelope
  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soap:Header/>
  <soap:Body>
    <ns0:識別番号メッセージ xmlns:ns0="urn:applic:xmlns:pf:lgxml:schema:2010-02">
      <ns1:共通ヘッダ xmlns:ns1="urn:applic:xmlns:pf:common:schema:2010-01">
        <ns1:受付番号>20100122001</ns1:受付番号>
      </ns1:共通ヘッダ>
    </ns0:識別番号メッセージ>
  </soap:Body>
</soap:Envelope>

```

```

        </ns1:共通ヘッダ>
        <ns0:利用業務ユニット>02</ns0:利用業務ユニット>
        <ns0:識別番号>19</ns0:識別番号>
    </ns0:識別番号メッセージ>
</soap:Body>
</soap:Envelope>

```

(2) レスポンスの例

- ① 全ての名前空間の宣言を Envelope 要素で行い、業務メッセージのタグと共通ヘッダのタグに名前空間接頭辞を付けるパターン

```

HTTP/1.1 200 OK
Server: Apache-Coyote/1.1
X-Powered-By: Servlet 2.4
Content-Type: text/xml;charset=utf-8
Date: Mon, 22 Nov 2010 10:02:35 GMT
Connection: close

<?xml version="1.0" encoding="UTF-8"?>
<soap:Envelope
  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:ns0="urn:applic:xmlns:pf:lgxml:schema:2010-02"
  xmlns:ns1="urn:applic:xmlns:pf:common:schema:2010-01">
  <soap:Header/>
  <soap:Body>
    <ns0:住登外情報メッセージ>
      <ns1:共通ヘッダ>
        <ns1:受付番号>1290420294015</ns1:受付番号>
        <ns1:業務サービス結果情報>0</ns1:業務サービス結果情報>
        <ns1:結果情報>0</ns1:結果情報>
      </ns1:共通ヘッダ>
      <ns0:住登外情報>
        <ns0:識別番号>0000000000</ns0:識別番号>
        <ns0:住民種別>3</ns0:住民種別>
        <ns0:住民状態></ns0:住民状態>
        . . . . . (略) . . . . .
      </ns0:住登外情報>
    </ns0:住登外情報メッセージ>
  </soap:Body>
</soap:Envelope>

```

- ② 業務メッセージと共通ヘッダの名前空間の宣言を SOAP の Body 部で行い、業務メッセージのタグと共通ヘッダのタグにデフォルト Namespace を記述するパターン

```

HTTP/1.1 200 OK

```

```
Server: Apache-Coyote/1.1
X-Powered-By: Servlet 2.4
Content-Type: text/xml;charset=utf-8
Date: Mon, 22 Nov 2010 10:02:35 GMT
Connection: close
```

```
<?xml version="1.0" encoding="UTF-8"?>
<soap:Envelope
  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soap:Header/>
  <soap:Body>
    <住登外情報メッセージ xmlns="urn:applic:xmlns:pf:lgxml:schema:2010-02">
      <共通ヘッダ xmlns="urn:applic:xmlns:pf:common:schema:2010-01">
        <受付番号>1290420294015</受付番号>
        <業務サービス結果情報>0</業務サービス結果情報>
        <結果情報>0</結果情報>
      </共通ヘッダ>
      <住登外情報>
        <識別番号>0000000000</識別番号>
        <住民種別>3</住民種別>
        <住民状態></住民状態>
        . . . . . (略) . . . . .
      </住登外情報>
    </住登外情報メッセージ>
  </soap:Body>
</soap:Envelope>
```

③ 業務メッセージと共通ヘッダの名前空間の宣言を SOAP の Body 部で行い、業務メッセージのタグにデフォルト Namespace を記述し、共通ヘッダのタグに名前空間接頭辞を付けるパターン

```
HTTP/1.1 200 OK
Server: Apache-Coyote/1.1
X-Powered-By: Servlet 2.4
Content-Type: text/xml;charset=utf-8
Date: Mon, 22 Nov 2010 10:02:35 GMT
Connection: close

<?xml version="1.0" encoding="UTF-8"?>
<soap:Envelope
  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soap:Header/>
  <soap:Body>
    <住登外情報メッセージ xmlns="urn:applic:xmlns:pf:lgxml:schema:2010-02">
      <ns1:共通ヘッダ xmlns:ns1="urn:applic:xmlns:pf:common:schema:2010-01">
        <ns1:受付番号>1290420294015</ns1:受付番号>
      </ns1:共通ヘッダ>
    </住登外情報メッセージ>
  </soap:Body>
</soap:Envelope>
```

```

    <ns1:業務サービス結果情報>0</ns1:業務サービス結果情報>
    <ns1:結果情報>0</ns1:結果情報>
  </ns1:共通ヘッダ>
  <住登外情報>
    <識別番号>0000000000</識別番号>
    <住民種別>3</住民種別>
    <住民状態></住民状態>
    . . . . . (略) . . . . .
  </住登外情報>
</住登外情報メッセージ>
</soap:Body>
</soap:Envelope>

```

- ④ 業務メッセージと共通ヘッダの名前空間の宣言を SOAP の Body 部で行い、業務メッセージのタグと共通ヘッダのタグに名前空間接頭辞を付けるパターン

```

HTTP/1.1 200 OK
Server: Apache-Coyote/1.1
X-Powered-By: Servlet 2.4
Content-Type: text/xml;charset=utf-8
Date: Mon, 22 Nov 2010 10:02:35 GMT
Connection: close

<?xml version="1.0" encoding="UTF-8"?>
<soap:Envelope
  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soap:Header/>
  <soap:Body>
    <ns0:住登外情報メッセージ xmlns:ns0="urn:applic:xmlns:pf:lgxml:schema:2010-02">
      <ns1:共通ヘッダ xmlns:ns1="urn:applic:xmlns:pf:common:schema:2010-01">
        <ns1:受付番号>1290420294015</ns1:受付番号>
        <ns1:業務サービス結果情報>0</ns1:業務サービス結果情報>
        <ns1:結果情報>0</ns1:結果情報>
      </ns1:共通ヘッダ>
      <ns0:住登外情報>
        <ns0:識別番号>0000000000</ns0:識別番号>
        <ns0:住民種別>3</ns0:住民種別>
        <ns0:住民状態></ns0:住民状態>
        . . . . . (略) . . . . .
      </ns0:住登外情報>
    </ns0:住登外情報メッセージ>
  </soap:Body>
</soap:Envelope>

```

上記 4 パターンにおいて、リクエストのパターンとそれに対するレスポンスのパターンは異なってもよい。また、名前空間接頭辞（上記の例における ns0 や ns1）は、上記例と違う名称でもよい。

利用条件

本書は、本書の内容および表現が変更されないこと、および出典、著作権表示を明示することを前提に、無償でその全部または一部を複製、転記、引用して利用できます。なお、全体を複製された場合は、本利用条件を明示してください。

一般財団法人全国地域情報化推進協会が公開するドキュメントの内容は無保証で提供されます。ここに含まれる情報の利用について商品性、特定目的適合性や第三者権利の不侵害その他一切の、明示的、黙示的保障を行いません。

Copyright ©一般財団法人全国地域情報化推進協会 2007-2016 All rights reserved.