

平成16年度
流通サプライチェーン全体最適化促進事業

< 実証実験 >

「モジュール構成書」

平成17年3月

日本電気株式会社

改版履歴

日付	版数	改版内容
2005年3月31日	初版	新規

実証実験統括責任者

日本電気株式会社 : 曾根田 雄一

検 印

検 印

目次

1. 本書の内容と構成	1-1
2. ビジネスモジュールの java パッケージ構成	2-1
3. java パッケージ	3-1
3.1 jp.go.meti.dscm.adapter パッケージ群	3-1
3.2 jp.go.meti.dscm.mapper パッケージ群	3-2
3.3 jp.go.meti.dscm.constraint パッケージ群	3-4
3.4 jp.go.meti.dscm.dispatch パッケージ群	3-6
3.5 jp.go.meti.dscm.event パッケージ群	3-6
3.6 jp.go.meti.dscm.communication パッケージ群	3-8
3.7 jp.go.meti.dscm.tools パッケージ群	3-11
3.8 jp.go.meti.dscm.trace パッケージ群	3-12
3.9 jp.go.meti.dscm.xml パッケージ群	3-12
3.10 jp.go.meti.dscm.ejb パッケージ群	3-12
3.11 その他のユーティリティパッケージ群	3-13
4. ビジネスモジュール API 仕様	4-1
4.1 全体概要	4-1
4.2 業務 AP アダプタインターフェース仕様	4-2
4.2.1 概要	4-2
4.2.2 マッピング処理インターフェース	4-3
4.2.3 送信インターフェース	4-5
4.2.4 受信インターフェース	4-6
4.3 通信アダプタインターフェース仕様	4-9
4.3.1 概要	4-9
4.3.2 送信インターフェース	4-9
4.3.3 受信インターフェース	4-12
4.4 コラボレーション仕様	4-14
4.4.1 コラボレーション記述ファイル	4-14
4.4.2 イベント記述	4-15
4.4.3 通信プロファイル	4-16

1. 本書の内容と構成

本書は、ビジネスモジュール（以下、BMと略す場合がある）を構成する幾つかのサブモジュールの構成、およびビジネスモジュールの外部API仕様について述べたものである。

本書は、ビジネスモジュールに対して機能追加等を行うためにソースコードを修正する場合に、ビジネスモジュールの内部構造の概要を把握するとともに、APIを利用して通信アダプタ・業務APアダプタを開発ために参照することを想定して記述されている。

ビジネスモジュールは1つのJ2EEアプリケーションであり、本書ではその内部構成をJavaのパッケージ単位で説明する構成をとっている。各クラス単位での詳細については、別途ビジネスモジュールのJavaDocを参照されたい。

2. ビジネスモジュールの j a v aパッケージ構成

BMは、大きく下図の J a v aパッケージで構成される。

実際のシステム上の配置では、下記パッケージの大半は、ビジネスモジュールをインストールしたディレクトリ内の、lib/bpc.jar という Java アーカイブに含まれている。また、EJB 部分については bpc.ear という J2EE アーカイブに含まれ、これが前述の bpc.jar を内部から参照する構成となっている。

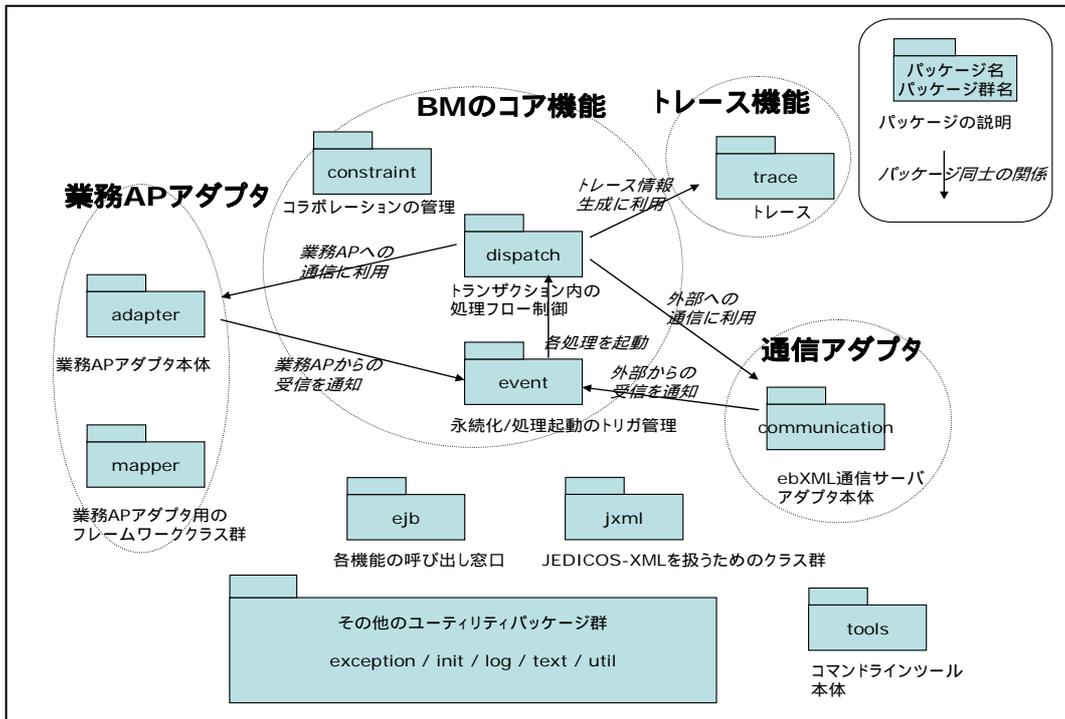


図 2 . 1 ビジネスモジュールの j a v aパッケージ構成

3. javaパッケージ

3.1 jp.go.meti.dscm.adapterパッケージ群

業務 AP アダプタを開発するためのフレームワークを提供する。このフレームワークを通じて、業務アプリケーションとBMの間の情報交換、および業務データ（固定長またはCSV）とJEDICOS - XMLメッセージ間のマッピングを行う。

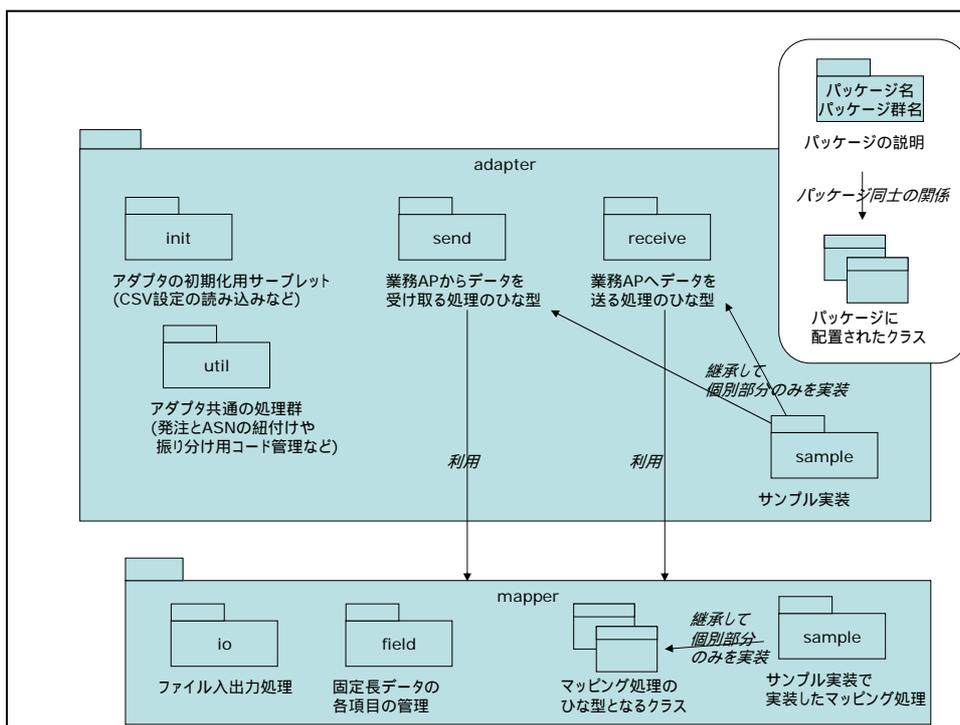


図 3.1 jp.go.meti.dscm.adapter パッケージ群（概念図）

表 3.2 jp.go.meti.dscm.adapter パッケージ群（一覧）

#	パッケージ名	説明
1	init	業務 AP アダプタの初期化(マッピング設定ファイル内容の読み込みなど)を行う。
2	util	業務 AP アダプタの共通処理(発注とASNの紐付けや振り分け用コード管理など)を行う。
3	send	業務アプリケーションからメッセージを受信し、BMにメッセージを送信するための機能を提供する。また、業務データファイルを読み込むためのユーティリティを提供する。

4	receive	B Mからメッセージを受信するための機能を提供する。
5	sample	<p>リファレンス実装を提供する。</p> <ul style="list-style-type: none"> ・ B Mから業務アプリケーションへ、J E D I C O S - X M L データを業務データに変換する。 ・ 業務アプリケーションから B Mへ、業務データを J E D I C O S - X M L データに変換する。 ・ 発注、発注データの受信を行う。 ・ 自由記述のデータを受信する。 ・ 業務アプリケーションと B M間の、データ送受信処理のひな型を生成する。

3.2 jp.go.meti.dscm.mapperパッケージ群

業務 AP アダプタフレームワークのうち、マッピングに関する機能を提供する。

マッピング設定ファイルの内容に基づいて、業務 AP アダプタの業務データ（固定長または C S V ）と業務メッセージ（ J E D I C O S - X M L ）の相互変換を行う。

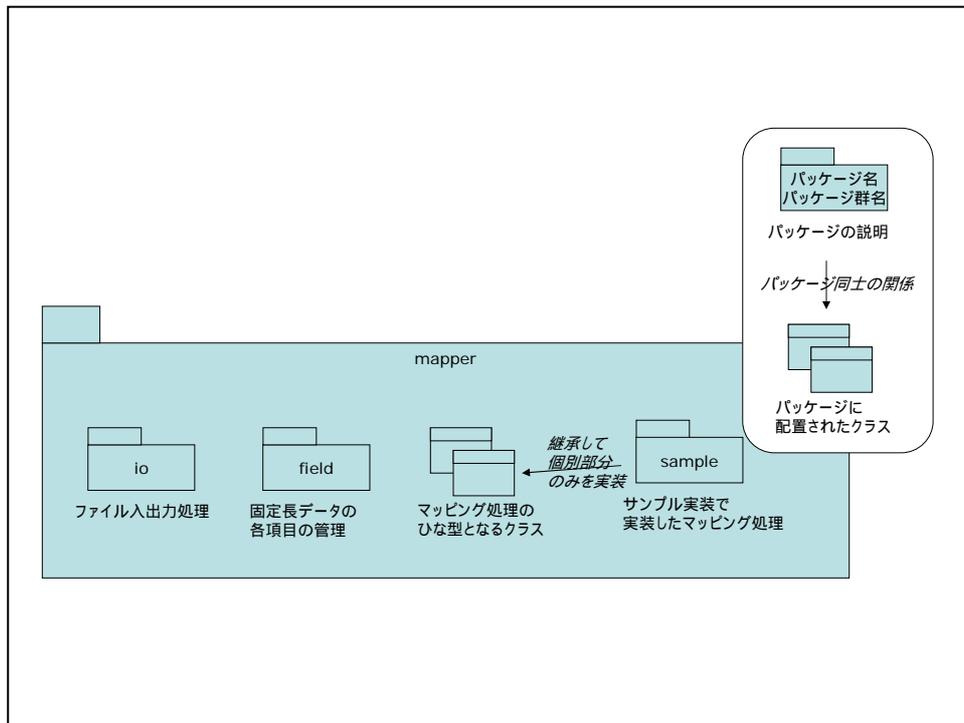


図 3 . 3 jp.go.meti.dscm.mapper パッケージ群 (概念図)

表 3 . 4 jp.go.meti.dscm.mapper パッケージ群 (一覧)

#	パッケージ名	説明
1	mapper	業務 A P アダプタの業務データと業務メッセージのマッピングを行う。
2	field	業務 A P アダプタの業務データのマッピング設定ファイル(CSVファイル)を読み込む。
3	io	業務 A P アダプタの業務データの入出力を行う。
4	sample	マッピング処理を実装したサンプル。業務 A P アダプタのサンプル実装から利用される。

3.3 jp.go.meti.dscm.constraintパッケージ群

BMの中核となる部分であり、予め定義されたコラボレーションを解釈し、それに従ったプロセス実行を制御する。

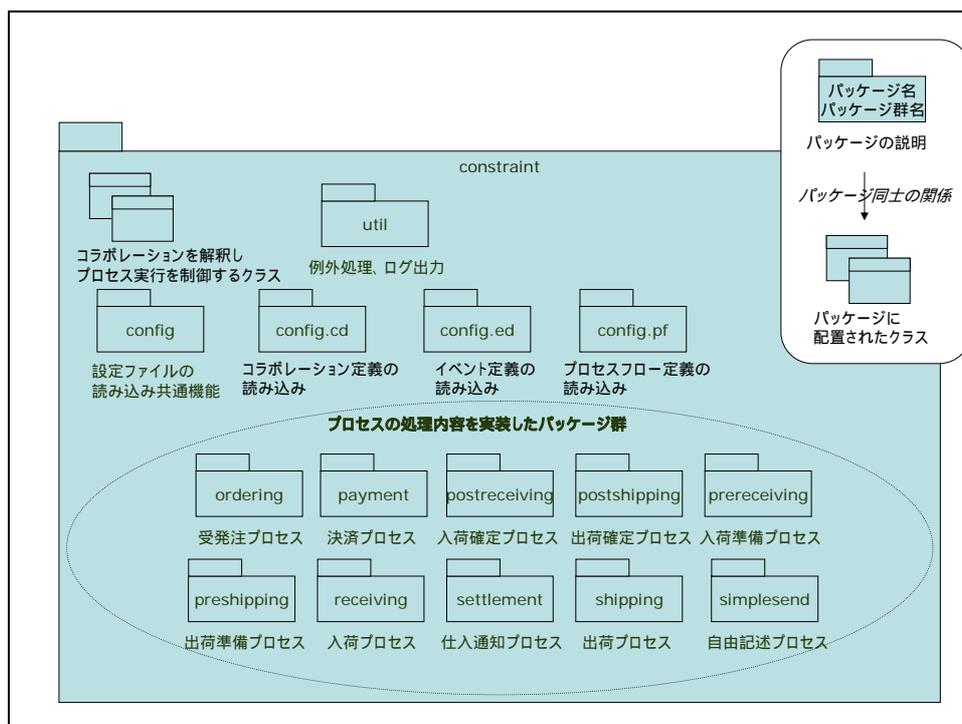


図 3.5 jp.go.meti.dscm.constraint パッケージ群 (概念図)

表 3.6 jp.go.meti.dscm.constraint パッケージ群 (一覧)

#	パッケージ名	説明
1	constraint	BMの中核となる部分であり、予め定義されたコラボレーションを解釈し、それに従ったプロセス実行順序の制御や、プロセス実行中の例外などの処理を行う。
2	config	設定ファイルを読み込むための共通機能を提供する。
3	config.cd	コラボレーション記述 (CollaborationDescriptor) を設定ファイルから読み込む。
4	config.ed	イベント記述 (EventDescriptor) を設定ファイルから読み込む。

5	config.pf	プロセスフロー定義（ProcessFlow）を設定ファイルから読み込む。
6	ordering	受発注プロセスを実装する。本プロセスにおいて特別に行う処理がないため、ビジネスモジュールではプロセス名の定義のみ行っている。
7	payment	決済プロセスを実装する。本プロセスにおいて特別に行う処理がないため、ビジネスモジュールではプロセス名の定義のみ行っている。
8	postreceiving	入荷確定プロセスを実装する。本プロセスにおいて特別に行う処理がないため、ビジネスモジュールではプロセス名の定義のみ行っている。
9	postshipping	出荷確定プロセスを実装する。本プロセスにおいて特別に行う処理がないため、ビジネスモジュールではプロセス名の定義のみ行っている。
10	prereceiving	入荷準備プロセスを実装する。本プロセスにおいて特別に行う処理がないため、ビジネスモジュールではプロセス名の定義のみ行っている。
11	preshipping	出荷準備プロセスを実装する。本プロセスにおいて特別に行う処理がないため、ビジネスモジュールではプロセス名の定義のみ行っている。
12	receiving	入荷プロセスを実装する。本プロセスにおいて特別に行う処理がないため、ビジネスモジュールではプロセス名の定義のみ行っている。
13	settlement	仕入通知プロセスを実装する。本プロセスにおいて特別に行う処理がないため、ビジネスモジュールではプロセス名の定義のみ行っている。
14	shipping	出荷プロセスを実装する。本プロセスにおいて特別に行う処理がないため、ビジネスモジュールではプロセス名の定義のみ行っている。
15	simplesend	自由記述プロセスを実装する。本プロセスにおいて特別に行う処理がないため、ビジネスモジュールではプロセス名の定義のみ行っている。
16	util	例外処理、ログ出力を行う。

3.4 jp.go.meti.dscm.dispatchパッケージ群

BMにおける業務処理の基本単位であるプロセスを呼び出し、ランザクション内の処理フロー制御を行なう。

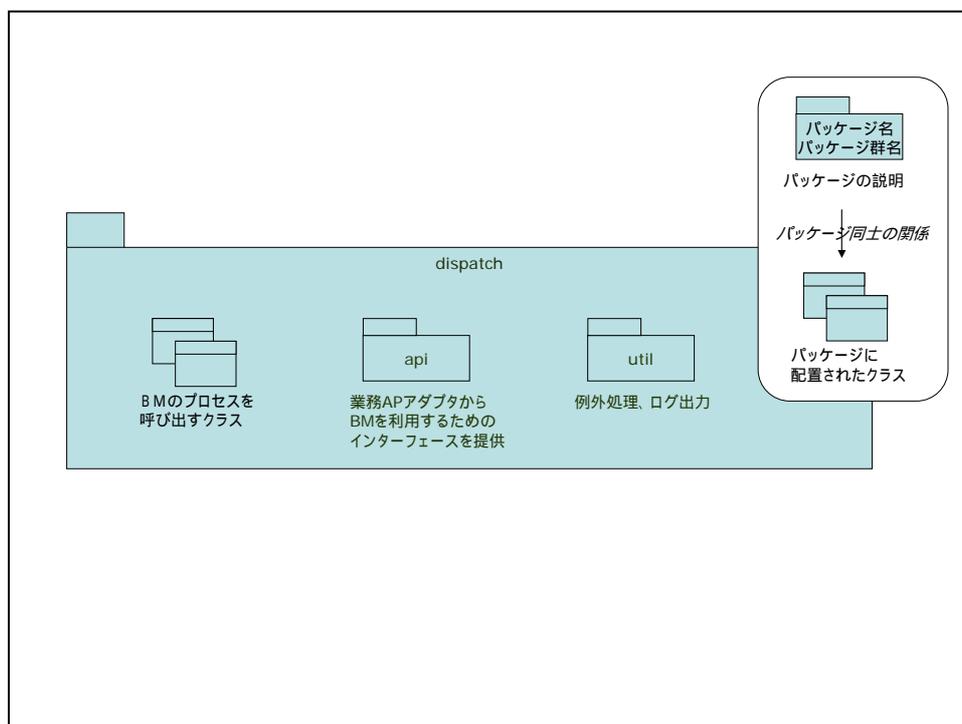


図 3.7 jp.go.meti.dscm.dispatch パッケージ群 (概念図)

表 3.8 jp.go.meti.dscm.dispatch パッケージ群 (一覧)

#	パッケージ名	説明
1	dispatch	BMにおける業務処理の基本単位であるプロセスを呼び出す。
2	api	eventパッケージからこのパッケージを利用するためのインターフェースを提供する。
3	util	例外処理、ログ出力を行う。

3.5 jp.go.meti.dscm.eventパッケージ群

プロセスの起動・終了の契機となる、イベントを管理する。
主な機能は以下の通り。

Event を登録し、これら Event が指定された開始時刻に、
EventListener を起動する。

- ebXML 通信サーバからの BpcDocument 受信
- 業務アプリケーションからの BpcDocument 受信

Process を起動する際、対応する Context および BpcDocument を渡す。

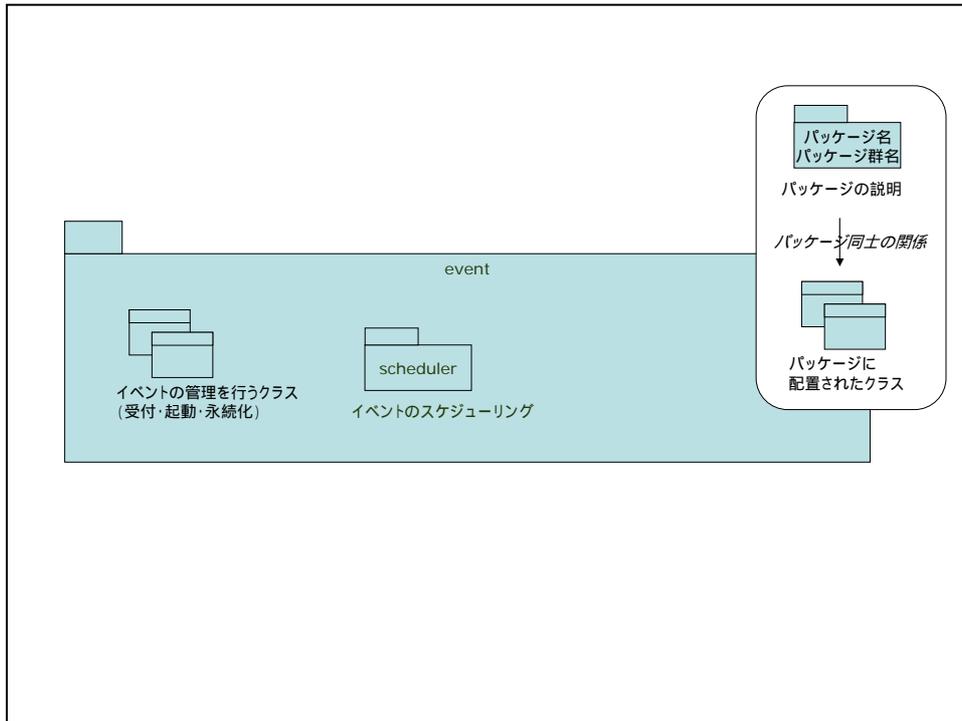


図 3 . 9 jp.go.meti.dscm.event パッケージ群 (概念図)

表 3 . 1 0 jp.go.meti.dscm.event パッケージ群 (一覧)

#	パッケージ名	説明
1	event	プロセスの起動・終了の契機となる、イベントを管理する。
2	scheduler	イベントのスケジューリングの機能を提供する。ビジネスモジュール本体とは別の J a v a プロセスとして動作し、指定時刻での起動および実行のタイムアウト、受信予定との違いを検出する。スケジューラはコマンドラインから起動される。

3.6 jp.go.meti.dscm.communication パッケージ群

e b X M L 通信機能を持つ通信サーバミドルウェア製品と B M の間の情報交換、および J E D I C O S - X M L メッセージの生成・解釈を行う。J E D I C O S - X M L 文書 (X M L テキスト) は、B P C オブジ

エクトと呼ばれる内部形式(JEDICOS - XMLのスキーマと同一の構造を持ったJavaオブジェクト)に変換される。

主な機能は以下の通り。

JEDICOS - XMLメッセージの送受信の管理。

- ・通信サーバからJEDICOS - XMLメッセージを受信。
- ・通信サーバへJEDICOS - XMLメッセージの送信を依頼。

JEDICOS - XMLメッセージとBPCオブジェクトの双方向の変換。

- ・JEDICOS - XMLメッセージをBPCオブジェクトへ変換する。
- ・BPCオブジェクトをJEDICOS - XMLメッセージへ変換する。

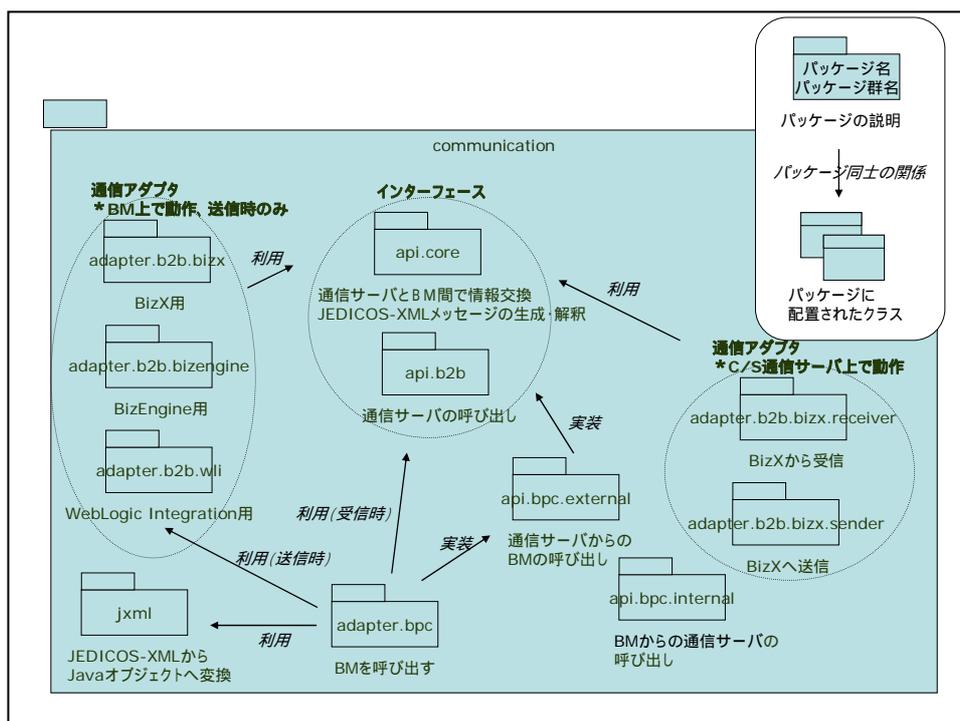


図 3 . 1 1 jp.go.meti.dscm.communication パッケージ群 (概念図)

表 3 . 1 2 jp.go.meti.dscm.communication パッケージ群 (一覧)

#	パッケージ名	説明
1	adapter.bpc	通信サーバからの受信時に、BM

			を呼び出すためのインターフェースを提供する。
2	api.bpc.external		通信サーバからのBPC呼び出し用APIを提供する。
3	api.bpc.internal		内部(BM)からの通信サーバ呼び出し用APIを提供する。
4	jxml		JEDICOS - XMLからJavaオブジェクトへの変換、および、バリデーションを行う。
5	adapter.b2b.bizx.receiver	通信アダプタ(C/S通信サーバ上で動作)	BizXから受信を行う。
6	adapter.b2b.bizx.sender		BizXへ送信を行う。
7	adapter.b2b.bizx	通信アダプタ(BM上で動作)	BizX用の通信アダプタ機能を提供する。送信時のみ利用。
8	adapter.b2b.wli		WebLogicIntegration用の通信アダプタ機能を提供する。送信時のみ利用。
9	adapter.b2b.bizengine		BizEngine用の通信アダプタ機能を提供する。送信時のみ利用。
10	api.b2b	インターフェース	通信サーバ呼び出しのためのAPIを提供し、EbxmlMessageを表すバリューオブジェクトを定義する。
11	api.core		ebXML通信機能を持つ通信サーバミドルウェア製品とBMの間の情報交換、およびJEDICOS - XMLメッセージの生成・解釈を行う。

3.7 jp.go.meti.dscm.toolsパッケージ群

ビジネスモジュール管理のためのコマンドラインのユーティリティを提供する。

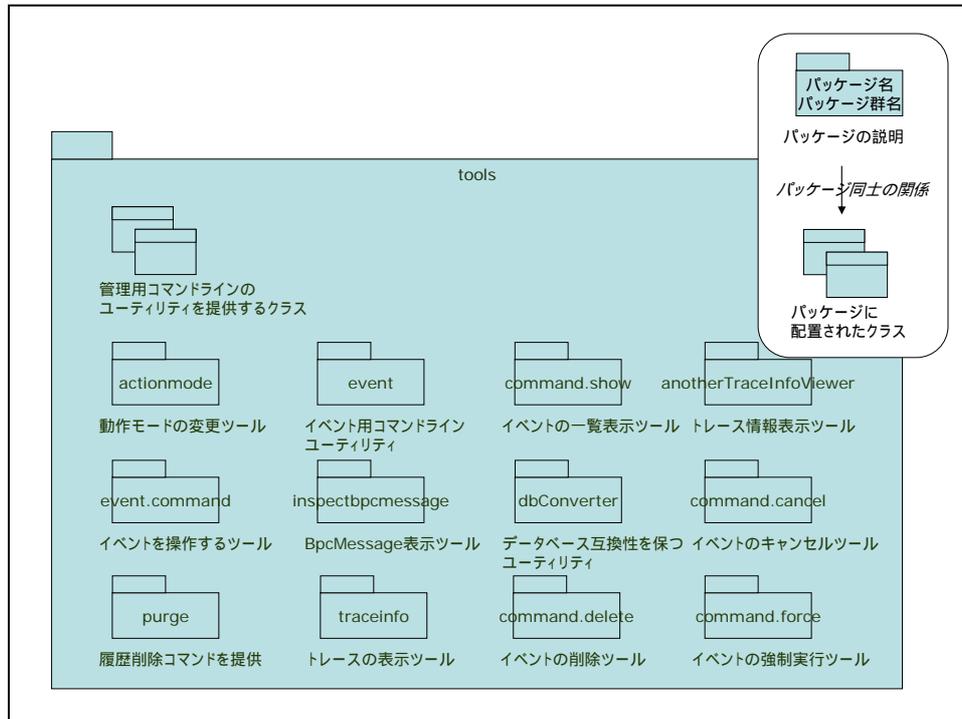


図 3 . 1 3 jp.go.meti.dscm.tools パッケージ群 (概念図)

表 3 . 1 4 jp.go.meti.dscm.tools パッケージ群 (一覧)

#	パッケージ名	説明
1	tools	コマンドラインの処理を行い、各機能に処理を振り分ける。
2	actionmode	B Mの動作モード(通常モード・メンテナンスモード)を変更するツールを提供する。
3	event	B Mのイベントの一覧表示、キャンセル、強制実行、削除をする。
4	event.command	eventパッケージの機能をコマンドラインから利用するための各種処理を行う。 ・イベントのキャンセル ・イベントの削除 ・イベントの強制実行

		・イベントの一覧表示
5	command.cancel	イベントをキャンセルする。
6	command.delete	イベントを削除する。
7	command.force	イベントを強制実行する。
8	command.show	イベントの一覧を表示する。
9	inspectbpcmessage	B p c M e s s a g eを表示する。
10	purge	指定された日付以前の履歴を削除する。
11	traceinfo	トレース情報を検索し、表示する。

3.8 jp.go.meti.dscm.traceパッケージ群

トレース機能を提供する。当該商品がどのような取引プロセスを経ているのかを確認できるように、トレースに必要な情報を生成しデータベースに記録する。

3.9 jp.go.meti.dscm.xmlパッケージ群

JEDICOS - XMLのスキーマから自動生成されたJavaクラス群。ビジネスモジュールは、JEDICOS - XML文書(XMLテキスト)を一旦この内部形式に変換して処理を行っている。

3.10 jp.go.meti.dscm.ejbパッケージ群

ビジネスモジュールの機能をEJBとして利用できるようにするためのラッピング処理を行っている。(EJB化することで、トランザクション管理、分散配置、セキュリティ管理などの拡張性が高まる)

ラッピング対象となる主な機能は下記の通り。

- ・業務APアダプタからBMに送信する際、BMに業務データを渡す。
- ・EVENT__SUBテーブルに対する操作を行なう。
- ・EVENTの実行を行なう。
- ・Ebxmlメッセージを受信する。
- ・BpcMessageを受信する。
- ・BpcMessageを送信する。

3.1.1 その他のユーティリティパッケージ群

#	パッケージ名	説明
1	jp.go.meti.dscm.exception	B Mで発生する例外の定義と、例外対応での共通処理を行なう。
2	jp.go.meti.dscm.init	B Mの起動時に行なわれる初期化処理に関する機能を提供する。 主な機能は下記の通り。 <ul style="list-style-type: none">・ Properties形式 (KEY=VALUE) での初期化項目の設定・ プログラム中から初期化項目の値を取得するためのクラスの提供 (抽象クラス)・ 各機能や各 Layer の依存関係を考慮し、初期化の順番を管理・ 初期化に失敗した場合の例外処理・ B 2 B 製品に依存しない初期化処理
3	jp.go.meti.dscm.log	ログ出力を行う。
4	jp.go.meti.dscm.text	文字エンコーディング、文字列操作 (検索・結合・変換等) を提供する。
5	jp.go.meti.dscm.util	ユーティリティ (ログ出力、処理時間計測、データベース接続など) を提供する。

4. ビジネスモジュールAPI仕様

4.1 全体概要

ビジネスモジュールは、既存の業務システムや通信サーバなどの外部のシステムと連携して処理を行う必要があるため、それらに対するインターフェースを持っている。ここで説明するインターフェース仕様には、以下の3つがある。本節の以降で、これらの仕様について詳細に説明する。

- 1) 業務APIアダプタインターフェース仕様
- 2) 通信アダプタインターフェース仕様
- 3) コラボレーション仕様

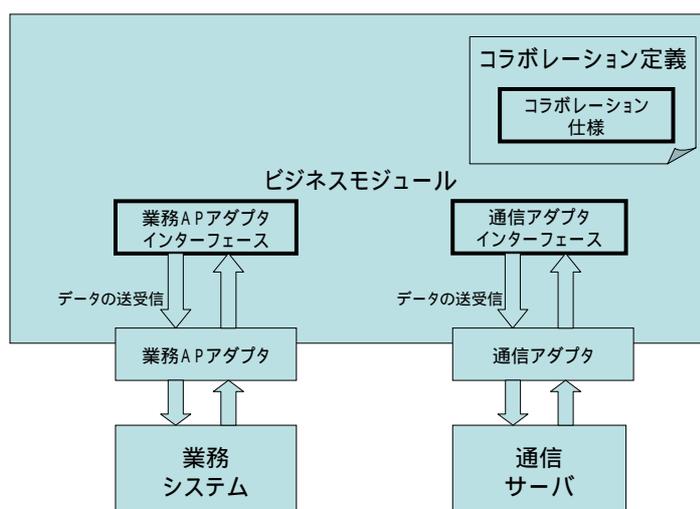


図 4.1 ビジネスモジュールの外部仕様の全体像

4.2 業務APアダプタインターフェース仕様

ビジネスモジュールは、外部の業務システムと連携するための機能を持っている。この機能は、製品に依存しない共通のインターフェースと、それを利用する業務システム別の業務APアダプタによって実現される。ここでは、業務APアダプタが提供するビジネスモジュール側のインターフェース（以下、業務APアダプタインターフェース）について記述する。

4.2.1 概要

業務APアダプタインターフェースは、大きく送信・受信の二つのインターフェースからなる。

送信インターフェースを提供しているのは、業務システムが作成したデータをJEDICOS - XMLに変換するためのToBpcMapperクラスおよび送信を担当するSendToBpcEJBである。

受信インターフェースを提供しているのは、JEDICOS - XMLメッセージを業務システム向けのデータに変換するためのFromBpcMapperクラスおよび業務APアダプタがビジネスモジュールからメッセージ到着の通知を受けるためのAbstractReceiveクラスである。

業務APアダプタインターフェースの詳細については、別途「業務アプリケーションアダプタ開発書」およびビジネスモジュールのJavaDoc（クラス仕様書）を参照されたい。

各クラスの正式名は以下のとおり。

```
jp.go.meti.dscm.ejb.SendToBpc  
jp.go.meti.dscm.adapter.receive.AbstractReceive  
jp.go.meti.dscm.mapper.ToBpcMapper  
jp.go.meti.dscm.mapper.FromBpcMapper
```

4.2.2 マッピング処理インターフェース

このインターフェース (`ToBpcMapper`・`FromBpcMapper`) は、`JEDICOS - XML`の項目と固定長あるいは`CSV`形式のデータレコードの項目とのマッピング処理を行うためのものである。`ToBpcMapper`は送信時の処理に、`FromBpcMapper`は受信時の処理に用いられる。これらのクラスを含んだ`mapper`パッケージの主要なクラス図を以下に示す。

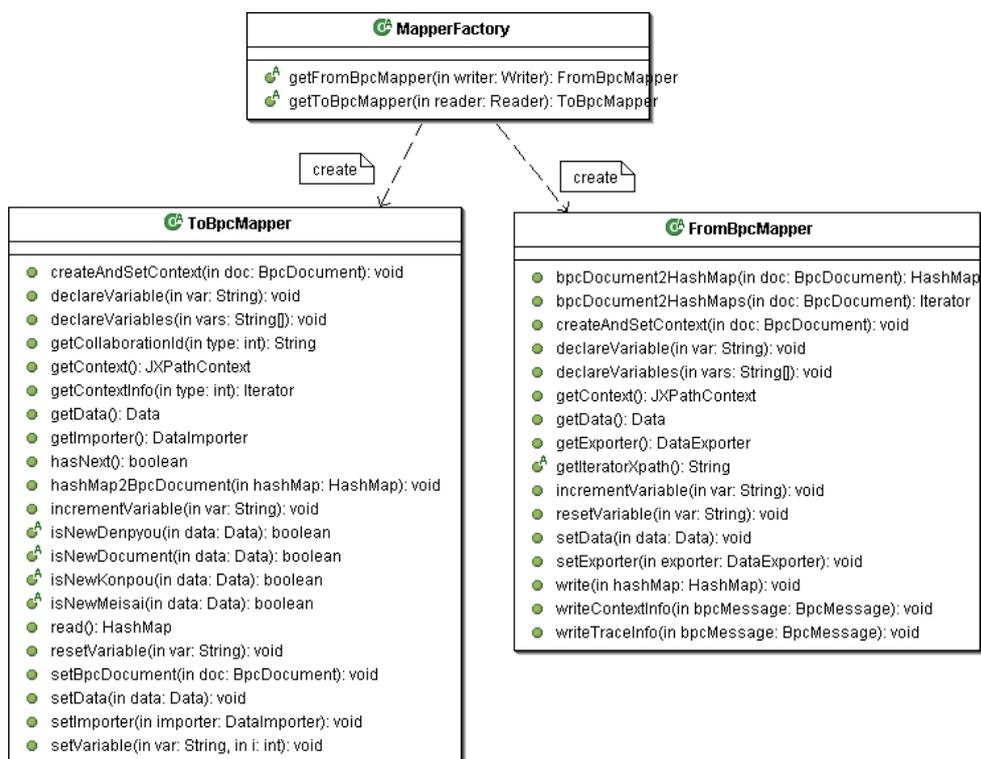


図 4.2 mapperパッケージのクラス図

上記クラスは、メッセージ項目マッピングツールによって自動生成されるため、業務APアダプタの開発者は、この自動生成されたクラス内に必要な個別処理を追記することになる。マッピングツールによる業務APアダプタ生成方法の概要を以下に示す。

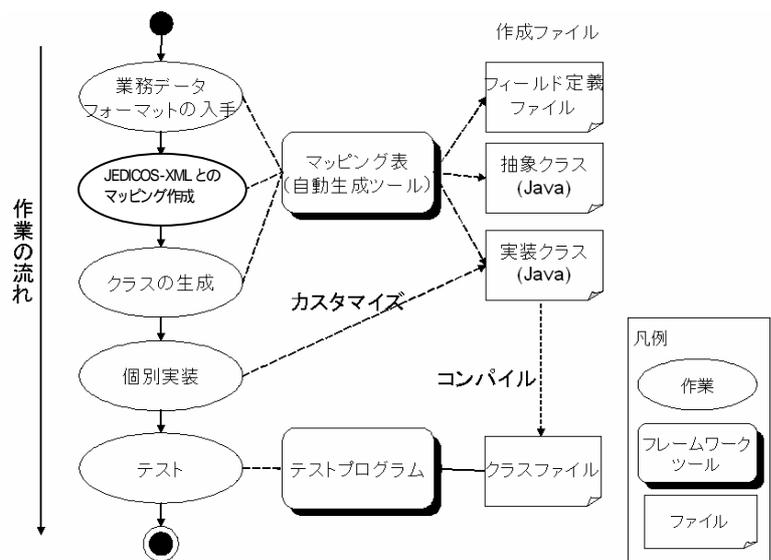


図 4.3 業務APアダプタ作成の流れと作成ファイル

4.2.3 送信インターフェース

業務APがビジネスモジュールへメッセージを送信するためには、業務APアダプタが必要なデータをSendToBpcEJB(SendToBpcBean)のsendメソッドに渡す必要がある。その後、EJBの中から、個別の変換処理を実装したToBpcMapperクラスが呼び出される。

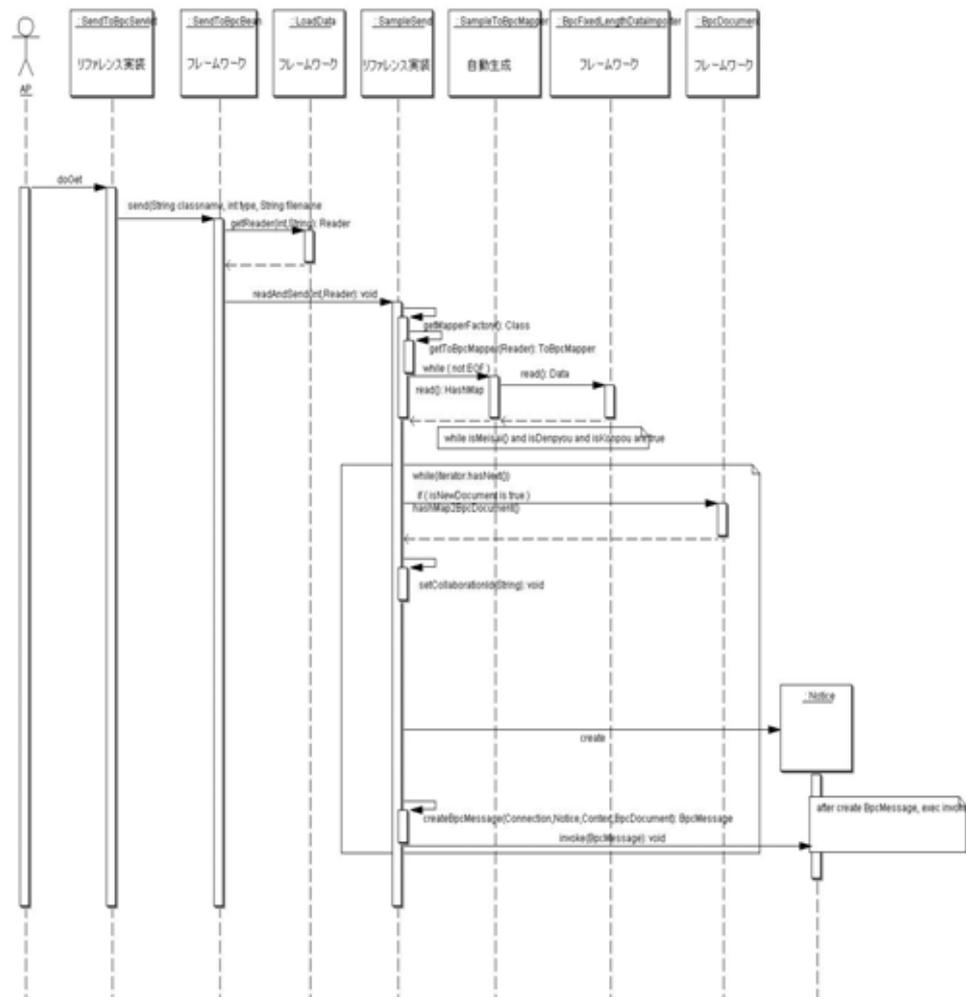


図 4.4 送信処理のシーケンス図

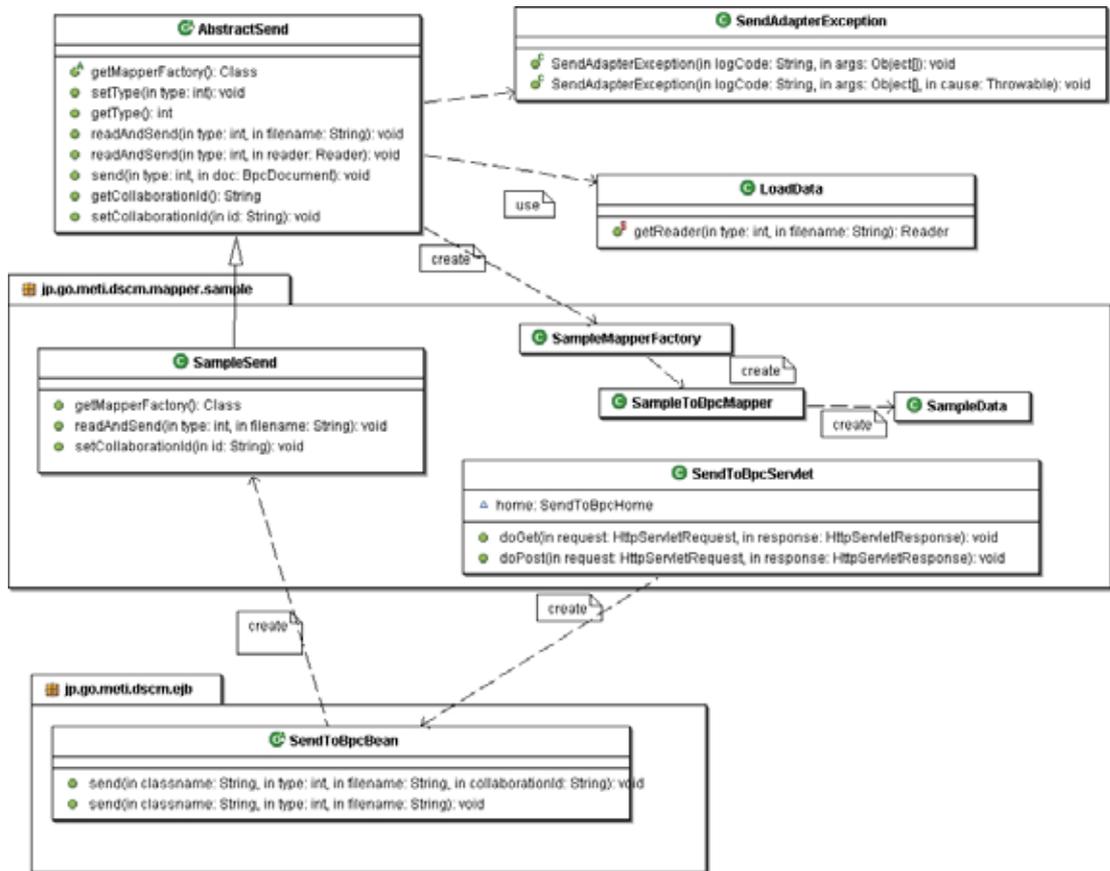


図 4.5 送信処理のクラス図
(送信業務 A Pアダプタのサンプル実装を含む)

4.2.4 受信インターフェース

ビジネスモジュールが受信したデータを業務システムに渡すためには、受信インターフェース `AbstractReceive#onNotify()` を実装した受信業務 A Pアダプタを作成し、イベント定義ファイル(次項にて記述)に登録する必要がある。

ビジネスモジュールがメッセージを受信すると、イベント定義を元に適切な受信業務 A Pアダプタの `onNotify()` が呼び出され、ビジネスモジュールからメッセージを受信することができる。

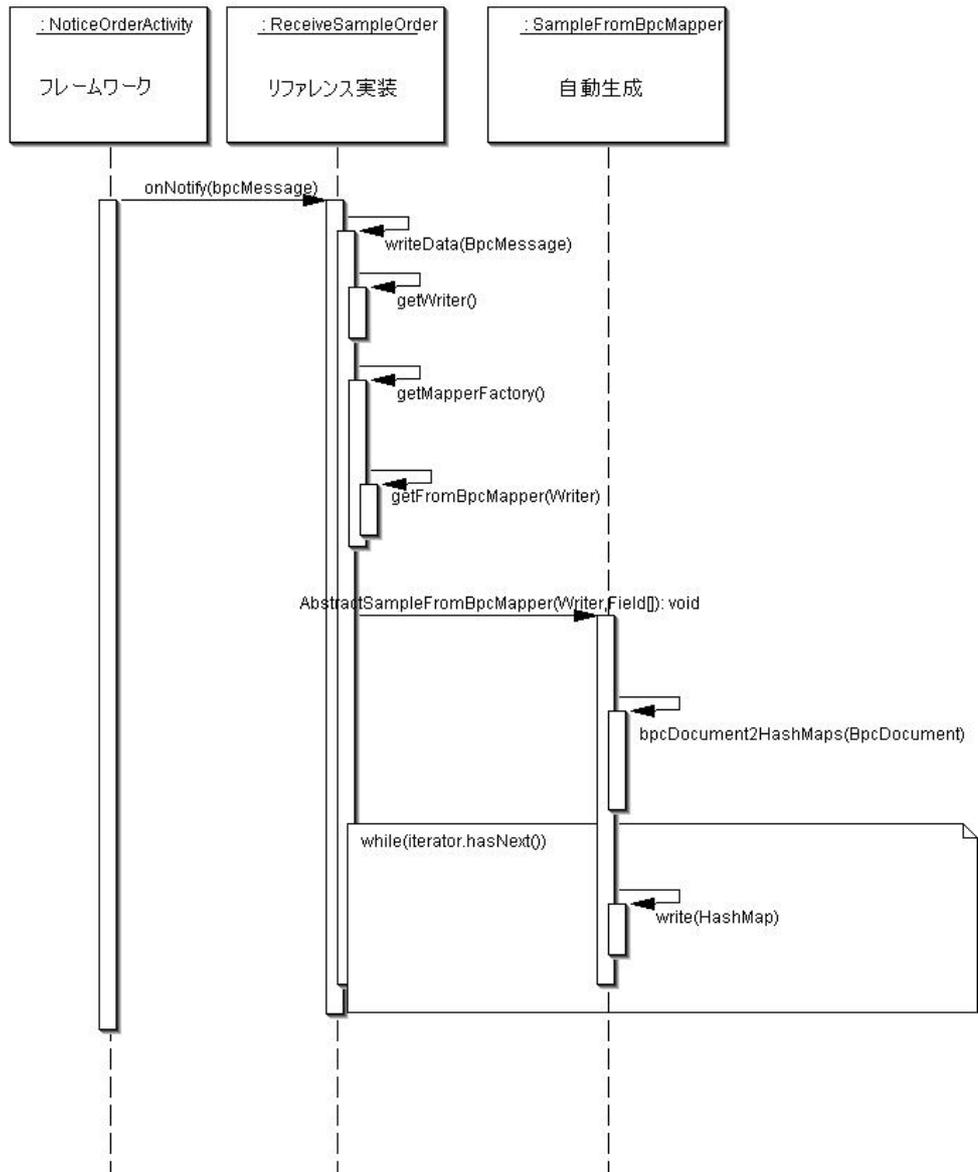


図 4.6 受信処理のシーケンス図

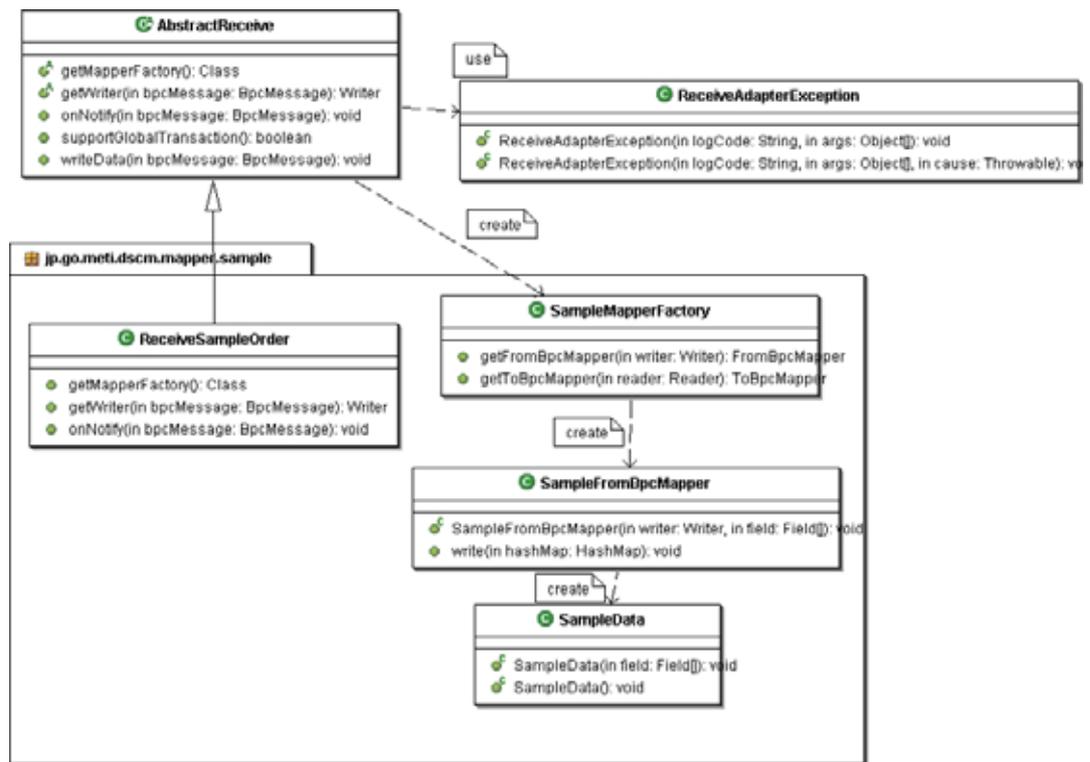


図 4.7 受信処理のクラス図
 (発注の受信業務 A P アダプタサンプルを含む)

4.3 通信アダプタインターフェース仕様

ビジネスモジュールは、e b X M L M S 2 . 0 に準拠した通信を行うため、外部の通信サーバと連携するための機能を持っている。この機能は、製品に依存しない共通のインターフェースと、製品に依存したインターフェースの実装（通信アダプタ）からなる。ここでは、共通のインターフェース（以下、通信アダプタインターフェース）について記述する。

4.3.1 概要

通信アダプタインターフェースは、大きく送信・受信の二つのインターフェースからなる。ビジネスモジュールは、E b x m l S e n d e r および E b x m l M e s s a g e というインターフェースを経由して、通信アダプタに対して送信先や送受信する内容を渡すことが出来る。

パッケージ名も含めたクラス名は以下のようなになる。

```
jp.go.meti.dscm.communication.api.b2b.EbxmlSender  
jp.go.meti.dscm.communication.api.b2b.EbxmlMessage
```

4.3.2 送信インターフェース

ビジネスモジュールは、外部に向けてメッセージを送信する際に、以下の処理を行う。

- (1) R e g i s t r y を通じて E b x m l S e n d e r インターフェースを実装したオブジェクトを探す。R e g i s t r y がどのオブジェクトを返却するかは、ビジネスモジュールの設定ファイルに記述される（リスト4.8）。
- (2) 発見されたオブジェクトに対して、M e s s a g e オブジェクトを渡すことで送信を依頼する。

リスト 4.8 b p c _ b 2 b . p r o p e r t i e s 設定ファイル

```
bpc.communication.ebxml_sender_class_name=mydomain.EbxmlSenderImpl
```

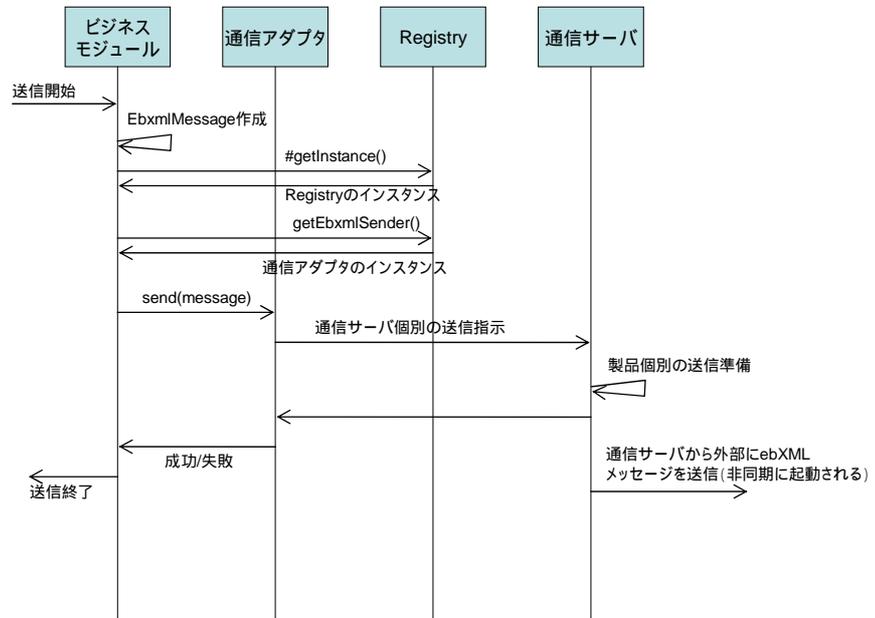


図 4.9 送信処理のシーケンス図

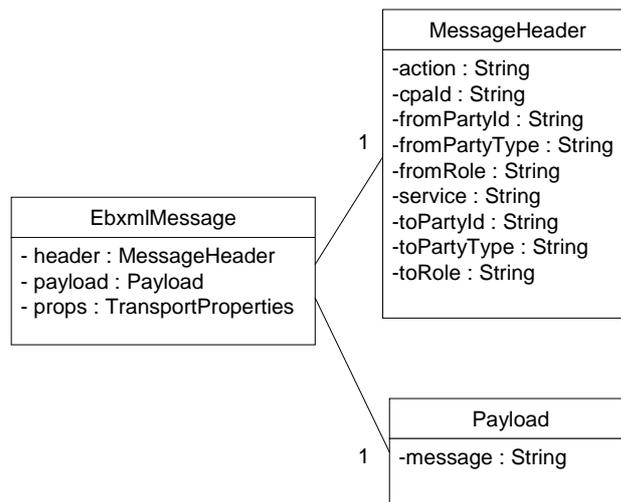


図 4.10 EbxmlMessageと関連クラスの格納する情報一覧

リスト 4.11 Registry.getInstanceのインターフェース仕様

```
public static Registry getInstance()  
    throws InstantiationException,  
           IllegalAccessException,  
           ClassNotFoundException,  
           IOException,  
           RemoteException,  
           javax.ejb.CreateException,  
           NamingException
```

レジストリのインスタンスを取得します。

戻り値:

レジストリのインスタンス

例外:

- InstantiationException
- IllegalAccessException
- ClassNotFoundException
- IOException
- RemoteException
- javax.ejb.CreateException
- NamingException

リスト 4.12 Registry.getEbxmlSenderのインターフェース仕様

```
public EbxmlSender getEbxmlSender()  
Registry から Ebxml Sender の実装 (アダプタ) を取得します。
```

戻り値:

Ebxml Sender

リスト 4.13 EbxmlSender.sendのインターフェース仕様

```
public void send(EbxmlMessage message)  
    throws SendAdapterException
```

通信サーバへメッセージを送信します。実際には、通信サーバへ message の送信処理を委譲します。

パラメータ:

message - Ebxml のメッセージとして送信する内容

例外:

SendAdapterException

4.3.3 受信インターフェース

ビジネスモジュールがメッセージを受信するためには、通信サーバが外部からメッセージを受信した際、その受信データを `EbxmlMessage` オブジェクトに設定した上で、以下の `EbxmlReceiveBean` (ステートレスセッションBean) を呼び出す必要がある。

EJBの呼び出し方法の説明 (JNDI名を含む) や、アプリケーションサーバの実装によるものであるため、ここでは割愛する。

リスト 4.14 EbxmlReceiveBean.receiveのインターフェース仕様

Bean名

`jp.go.meti.dscm.ejb.EbxmlReceiveBean`

インターフェース

```
public void receive(EbxmlMessage message)
    throws ReceiveAdapterException,
           RemoteException
```

通信サーバからの EbXML メッセージを受信します。

パラメータ:

`message` - 受信した Ebxml メッセージ 通信サーバ内で生成する必要がある。

例外:

`ReceiveAdapterException`
`RemoteException`

なお、この呼び出しは以下の EJB 設定にあるとおり、新規トランザクションとして扱われる。グローバルトランザクションには対応していない。

リスト 4.15 EbxmlReceiveBeanのejb-jar.xml

```
<session >
  <ejb-name>EbxmlReceive</ejb-name>
  <home>jp.go.meti.dscm.ejb.EbxmlReceiveHome</home>
  <remote>jp.go.meti.dscm.ejb.EbxmlReceive</remote>
  <local-home>jp.go.meti.dscm.ejb.EbxmlReceiveLocalHome</local-home>
  <local>jp.go.meti.dscm.ejb.EbxmlReceiveLocal</local>
  <ejb-class>jp.go.meti.dscm.ejb.EbxmlReceiveSession</ejb-class>
  <session-type>Stateless</session-type>
  <transaction-type>Container</transaction-type>
</session>
<container-transaction >
  <method >
    <ejb-name>EbxmlReceive</ejb-name>
```

```
<method-name>*</method-name>  
</method>  
<trans-attribute>RequiresNew</trans-attribute>  
</container-transaction>
```

4.4 コラボレーション仕様

ビジネスモジュールを利用してコラボレーション（メッセージ交換の順序）を記述するためには、以下の3種類の定義ファイルを使用する。それら3種類のファイル（コラボレーション記述、イベント記述、通信プロファイル）の仕様について、概要を説明する。

詳細については、「ビジネスモジュール環境設定書」を参照されたい。

4.4.1 コラボレーション記述ファイル

このファイルは、情報交換の流れ（プロセスフロー）と、その情報交換への参加者（パーティー）を定義するためのものである。このファイルは、これらの参加者間であらかじめ共有しなければならない。

パーティ間の通信条件の定義、パーティとロールを関連付けるアサインの定義、実行するプロセスフローのパターンの設定を行なう。

コラボレーション記述のXML論理構造は次のようになる。

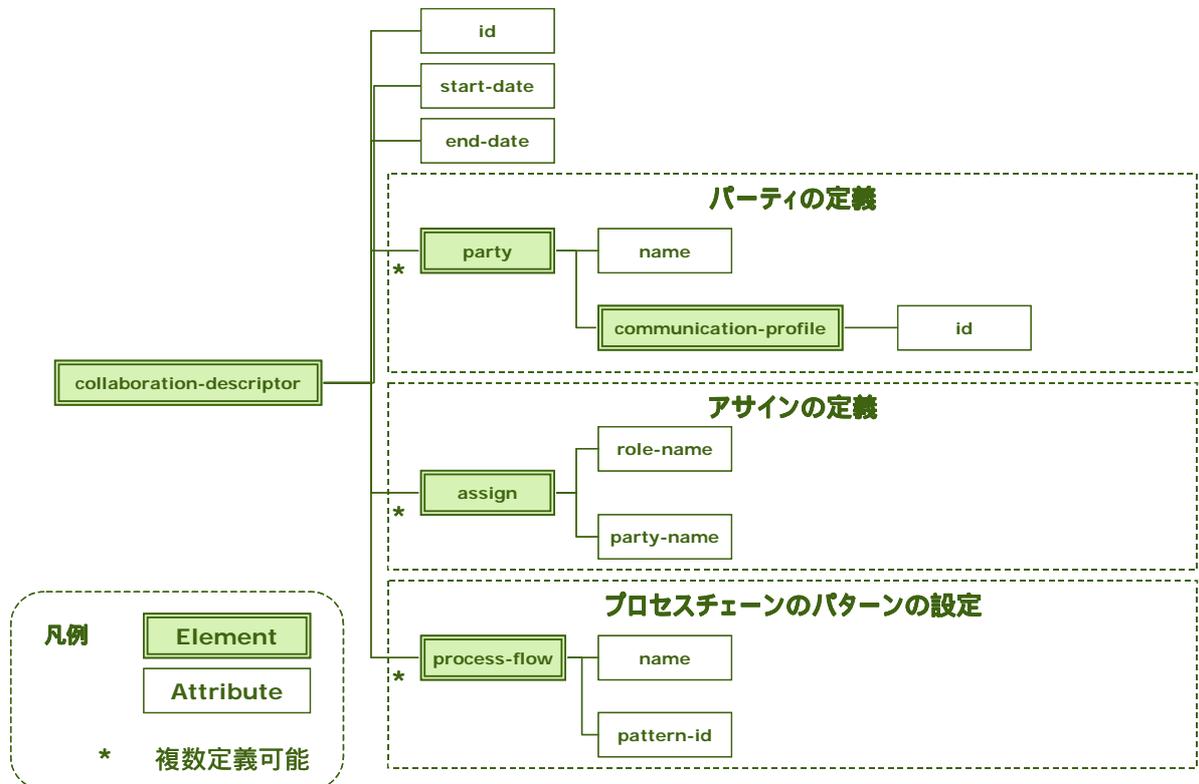


図 4.16 コラボレーション記述のXML論理構造

4.4.2 イベント記述

コラボレーション記述で定義された個々のコラボレーションについて、参加者間で異なる設定を行う部分（スケジュール時刻などの）を切り出して定義するための設定ファイルである。

イベント記述のXML論理構造は以下の通り。

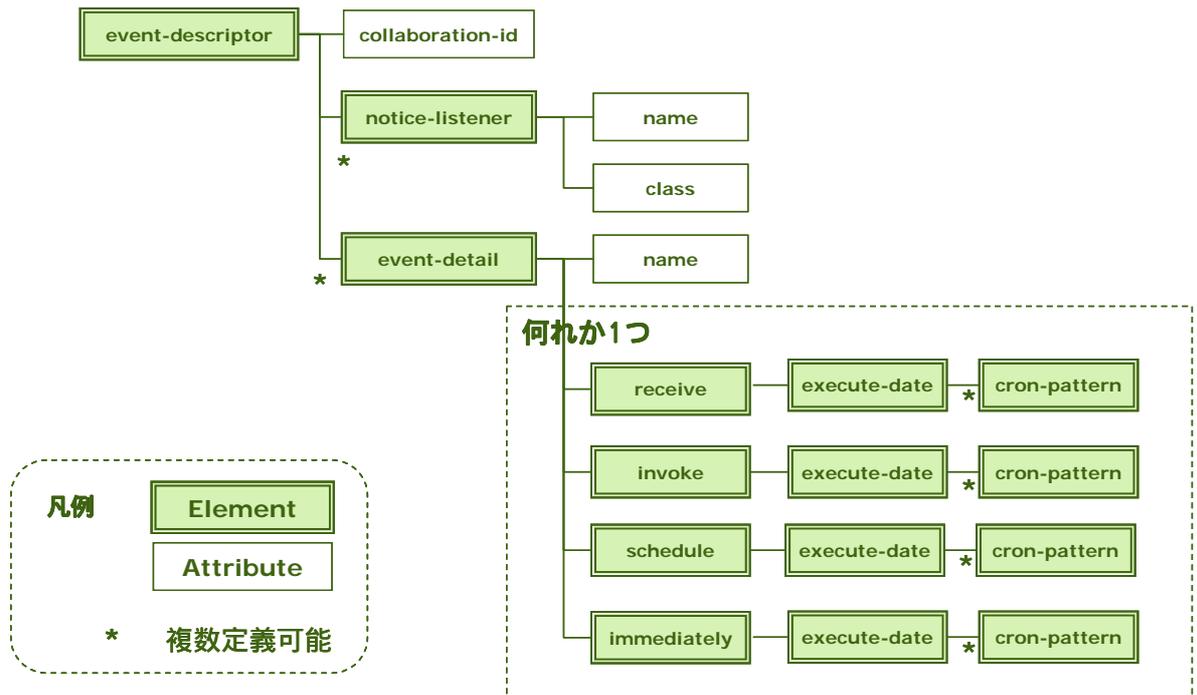


図 4.17 イベント記述のXML論理構造

4.4.3 通信プロフィール

通信サーバと連携するために必要な各種の設定値を管理するためのファイルである。主に、送信先などの情報を e b X M L の C P A にマッピングするために使用される。このファイルは、送受信を行う2つの拠点間で共有しなければならない。

通信プロフィールの X M L 論理構造を以下に示す。

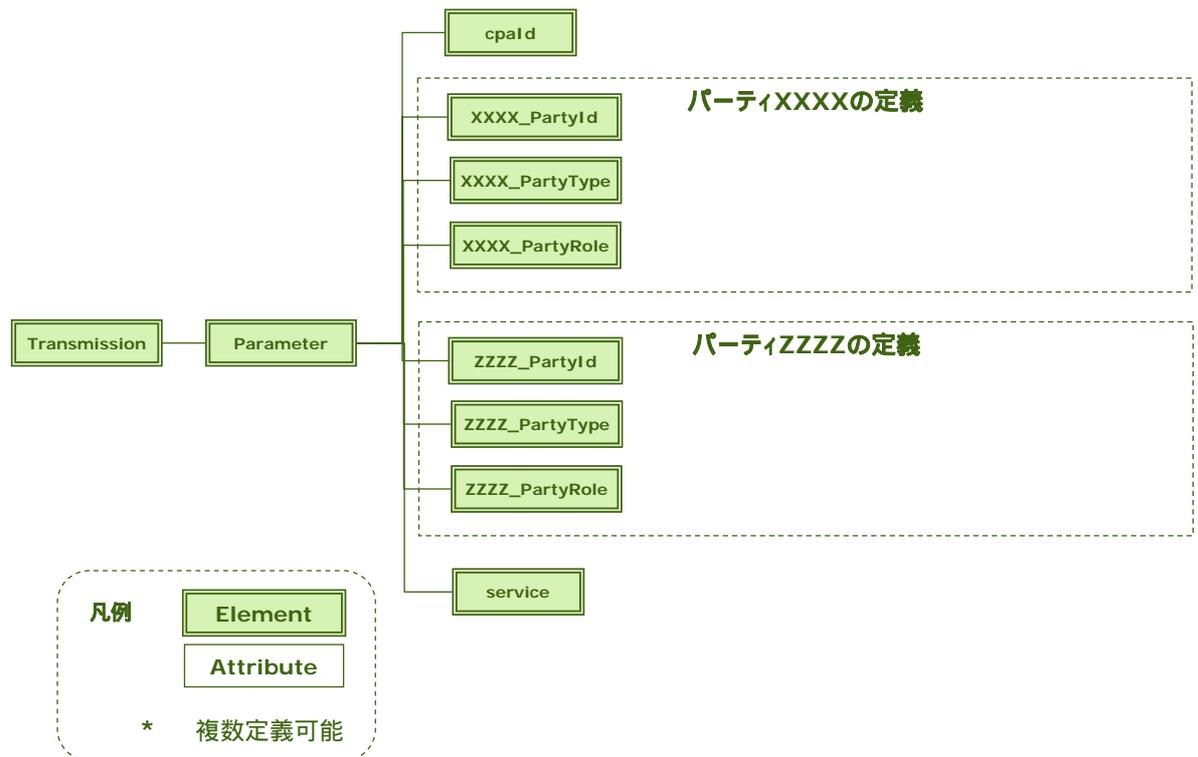


図 4.18 通信プロフィールの X M L 論理構造