

Cosminexus

ビジネスプロセス管理 / エンタープライズサービスバス V8

サービスプラットフォーム 機能解説

解説書

3020-3-U54-80

対象製品

適用 OS : Windows Server 2003 , Windows Server 2003 R2 , Windows XP , Windows Vista , Windows 7 x86 , Windows 7 x64 ¹

P-2443-7T84 uCosminexus Service Architect 08-70

適用 OS : Windows Server 2003 , Windows Server 2003 R2 , Windows Server 2003 (x64) ¹ , Windows Server 2003 R2 (x64) ¹ , Windows Server 2008 x86 , Windows Server 2008 x64 ¹ , Windows Server 2008 R2

P-2443-7S84 uCosminexus Service Platform 08-70

適用 OS : Windows Server 2003 (x64) , Windows Server 2003 R2 (x64) , Windows Server 2008 x64 , Windows Server 2008 R2

P-2943-7S84 uCosminexus Service Platform 08-70 ²

適用 OS : HP-UX 11i V2 (IPF) , HP-UX 11i V3 (IPF)

P-1J43-7S81 uCosminexus Service Platform 08-70 ²

適用 OS : AIX V7.1 , AIX V6.1 , AIX 5L V5.3

P-1M43-7S81 uCosminexus Service Platform 08-70 ²

適用 OS : Red Hat Enterprise Linux AS 4 (x86) , Red Hat Enterprise Linux ES 4 (x86) , Red Hat Enterprise Linux AS 4 (AMD64 & Intel EM64T) , Red Hat Enterprise Linux ES 4 (AMD64 & Intel EM64T) , Red Hat Enterprise Linux 5 Advanced Platform (x86) , Red Hat Enterprise Linux 5 (x86) , Red Hat Enterprise Linux 5 Advanced Platform (AMD/Intel 64) , Red Hat Enterprise Linux 5 (AMD/Intel 64) , Red Hat Enterprise Linux Server 6 (32-bit x86) , Red Hat Enterprise Linux Server 6 (64-bit x86_64)

P-9S43-7S81 uCosminexus Service Platform 08-70 ²

適用 OS : Red Hat Enterprise Linux AS 4 (IPF) , Red Hat Enterprise Linux 5 Advanced Platform (Intel Itanium) , Red Hat Enterprise Linux 5 (Intel Itanium)

P-9V43-7S81 uCosminexus Service Platform 08-70 ²

適用 OS : Windows Server 2003 , Windows Server 2003 R2 , Windows XP , Windows Vista , Windows 7 x86 , Windows 7 x64 ¹

P-2443-CL14 uCosminexus Service Adapter Architect for FTP 01-10

適用 OS : Windows Server 2003 , Windows Server 2003 (x64) ¹ , Windows Server 2003 R2 , Windows Server 2003 R2 (x64) ¹ , Windows Server 2008 x86 , Windows Server 2008 x64 ¹ , Windows Server 2008 R2 , HP-UX 11i V2 (IPF) , HP-UX 11i V3 (IPF) , AIX V7.1 , AIX V6.1 , AIX 5L V5.3 , Red Hat Enterprise Linux AS 4 (x86) , Red Hat Enterprise Linux ES 4 (x86) , Red Hat Enterprise Linux AS 4 (AMD64 & Intel EM64T) , Red Hat Enterprise Linux ES 4 (AMD64 & Intel EM64T) , Red Hat Enterprise Linux 5 Advanced Platform (x86) , Red Hat Enterprise Linux 5 (x86) , Red Hat Enterprise Linux 5 Advanced Platform (AMD/Intel 64) , Red Hat Enterprise Linux 5 (AMD/Intel 64) , Red Hat Enterprise Linux Server 6 (32-bit x86) , Red Hat Enterprise Linux Server 6 (64-bit x86_64)

P-2443-CK14 uCosminexus Service Adapter for FTP 01-10

注 1 WOW64 (Windows On Windows 64) 環境だけで使用できます。

注 2 製品のサポート時期をご確認ください。

これらのプログラムプロダクトのほかにもこのマニュアルをご利用になれる場合があります。詳細は「リリースノート」でご確認ください。

本製品では日立トレース共通ライブラリをインストールします。

輸出時の注意

本製品を輸出される場合には、外国為替および外国貿易法ならびに米国の輸出管理関連法規などの規制をご確認の上、必要な手続きをお取りください。

なお、ご不明な場合は、弊社担当営業にお問い合わせください。

商標類

AIX は、米国およびその他の国における International Business Machines Corporation の商標です。

AIX 5L は、米国およびその他の国における International Business Machines Corporation の商標です。

AMD は、Advanced Micro Devices, Inc. の商標です。

Borland のブランド名および製品名はすべて、米国 Borland Software Corporation の米国およびその他の国における商標または登録商標です。

CORBA は、Object Management Group が提唱する分散処理環境アーキテクチャの名称です。

HP-UX は、Hewlett-Packard Company のオペレーティングシステムの名称です。

IIOP は、OMG 仕様による ORB(Object Request Broker) 間通信のネットワークプロトコルの名称です。

Itanium は、アメリカ合衆国およびその他の国における Intel Corporation の商標です。

iWay および iWay Software は、Information Builders, Inc. の米国およびその他の国における登録商標です。

J2EE は、Oracle Corporation 及びその子会社、関連会社の米国 及びその他の国における登録商標または商標です。

Java は、Oracle Corporation 及びその子会社、関連会社の米国 及びその他の国における登録商標または商標です。

JDBC は、Oracle Corporation 及びその子会社、関連会社の米国 及びその他の国における登録商標または商標です。

JSP は、Oracle Corporation 及びその子会社、関連会社の米国及びその他の国における登録商標または商標です。

Linux は、Linus Torvalds 氏の日本およびその他の国における登録商標または商標です。

Microsoft は、米国 Microsoft Corporation の米国およびその他の国における登録商標または商標です。

MyEclipse は、米国 Genuitec 社の商品名称です。

Netscape は、米国およびその他の国における Netscape Communications Corporation の登録商標です。

OMG, CORBA, IIOP, UML, Unified Modeling Language, MDA, Model Driven Architecture は、Object Management Group, Inc. の米国及びその他の国における登録商標または商標です。

Oracle 及び Oracle 10g は、Oracle Corporation 及びその子会社、関連会社の米国 及びその他の国における登録商標または商標です。

Oracle 及び Oracle9i は、Oracle Corporation 及びその子会社、関連会社の米国 及びその他の国における登録商標または商標です。

Oracle 及び Oracle Database 11g は、Oracle Corporation 及びその子会社、関連会社の米国 及びその他の国における登録商標または商標です。

Red Hat は、米国およびその他の国で Red Hat, Inc. の登録商標もしくは商標です。

SAP, R/3, SAP NetWeaver, および記載されているすべてのその他の SAP 製品およびサービス名は、ドイツおよびその他世界各国における SAP AG の商標または登録商標です。

SOAP (Simple Object Access Protocol) は、分散ネットワーク環境において XML ベースの情報を交換するための通信プロトコルの名称です。

UNIX は、The Open Group の米国ならびに他の国における登録商標です。

W3C は、World Wide Web Consortium の商標（多数の国において登録された）です。

WebSphere は、米国およびその他の国における International Business Machines Corporation の商標です。

Windows は、米国 Microsoft Corporation の米国およびその他の国における登録商標または商標です。

Windows Server は、米国 Microsoft Corporation の米国およびその他の国における登録商標または商標です。

Windows Vista は、米国 Microsoft Corporation の米国およびその他の国における登録商標または商標です。

Eclipse は、開発ツールプロバイダのオープンコミュニティである Eclipse Foundation, Inc. により構築された開発ツール統合のためのオープンプラットフォームです。

This product includes software developed by the Apache Software Foundation (<http://www.apache.org/>).

マイクロソフト製品の表記について

このマニュアルでは、マイクロソフト製品の名称を次のように表記しています。

表記			製品名
Microsoft Cluster Service			Microsoft(R) Cluster Service
Windows	Windows 7	Windows 7 x86	Microsoft(R) Windows(R) 7 Enterprise(32 ビット版)
			Microsoft(R) Windows(R) 7 Professional(32 ビット版)
			Microsoft(R) Windows(R) 7 Ultimate(32 ビット版)
		Windows 7 x64	Microsoft(R) Windows(R) 7 Enterprise(64 ビット版)
			Microsoft(R) Windows(R) 7 Professional(64 ビット版)
			Microsoft(R) Windows(R) 7 Ultimate(64 ビット版)
Windows Server 2003	Windows Server 2003 Enterprise Edition	Microsoft(R) Windows Server(R) 2003, Enterprise Edition Operating System (x86)	
	Windows Server 2003 Standard Edition	Microsoft(R) Windows Server(R) 2003, Standard Edition Operating System (x86)	
Windows Server 2003 R2	Windows Server 2003 R2 Enterprise Edition	Microsoft(R) Windows Server(R) 2003 R2, Enterprise Edition Operating System (x86)	
	Windows Server 2003 R2 Standard Edition	Microsoft(R) Windows Server(R) 2003 R2, Standard Edition Operating System (x86)	
Windows Server 2003 (x64)	Windows Server 2003 Enterprise x64 Edition	Microsoft(R) Windows Server(R) 2003, Enterprise x64 Edition Operating System	
	Windows Server 2003 Standard x64 Edition	Microsoft(R) Windows Server(R) 2003, Standard x64 Edition Operating System	

表記		製品名
Windows Server 2003 R2 (x64)	Windows Server 2003 R2 Enterprise x64 Edition	Microsoft(R) Windows Server(R) 2003 R2, Enterprise x64 Edition Operating System
	Windows Server 2003 R2 Standard x64 Edition	Microsoft(R) Windows Server(R) 2003 R2, Standard x64 Edition Operating System
Windows Server 2008 x86		Microsoft(R) Windows Server(R) 2008 Enterprise 32-bit
		Microsoft(R) Windows Server(R) 2008 Standard 32-bit
Windows Server 2008 x64		Microsoft(R) Windows Server(R) 2008 Enterprise x64 Edition
		Microsoft(R) Windows Server(R) 2008 Standard x64 Edition
Windows Server 2008 R2		Microsoft(R) Windows Server(R) 2008 R2 Enterprise
		Microsoft(R) Windows Server(R) 2008 R2 Standard
Windows Vista	Windows Vista Business	Microsoft(R) Windows Vista(R) Business
	Windows Vista Enterprise	Microsoft(R) Windows Vista(R) Enterprise
	Windows Vista Ultimate	Microsoft(R) Windows Vista(R) Ultimate
Windows XP		Microsoft(R) Windows(R) XP Professional Operating System

発行

2011年8月 3020-3-U54-80

著作権

All Rights Reserved. Copyright (C) 2008, 2011, Hitachi, Ltd.

変更内容

変更内容 (3020-3-U54-80) uCosminexus Service Architect 08-70 , uCosminexus Service Platform 08-70 , uCosminexus Service Adapter Architect for FTP 01-10 , uCosminexus Service Adapter for FTP 01-10

追加・変更内容	変更箇所
Cosminexus サービスプラットフォームを構成するソフトウェア製品ごとの前提 OS に、次の OS を追加した。 <ul style="list-style-type: none">• AIX V7.1• Red Hat Enterprise Linux Server 6 (32-bit x86)• Red Hat Enterprise Linux Server 6 (64-bit x86_64)	-
Eclipse Classic 3.6.1 に対応したことに伴い、最新の画面図に差し替えた。	1.1.1 , 1.1.2 , 1.1.3 , 1.1.4 , 1.1.5 , 1.1.11
データベースを利用する環境の場合に、運用環境と実行環境を 1 対 1 で構成できるようにした。	1.5
リポジトリのコンポーネントインポート機能を追加した。	1.6
Web サービスの接続先 URL を取得できなかった場合に、動的に接続先を変更して SOAP 通信をできるようにした。	2.2.6
電文のフォーマット定義ファイルに任意形式 (any 形式) を設定できるようにしたことに伴い、説明を追加した。	4.2.6 , 5.6
Cosminexus サービスプラットフォームのシステムを JP1 で監視する場合の推奨プロセスの説明を追加した。	8.1.2
Cosminexus サービスプラットフォームの前提となるデータベースから、次のデータベースを削除した。 <ul style="list-style-type: none">• HiRDB Version 7• Oracle 9i• Oracle 10g R1	8.2

単なる誤字・脱字などはお断りなく訂正しました。

はじめに

このマニュアルは、Cosminexus サービスプラットフォームで SOA を適用したシステムを実現するための開発環境、および実行・運用環境の機能について説明しています。

Cosminexus サービスプラットフォームでは、次に示すプログラムプロダクトを使用してシステムを開発、運用します。

- uCosminexus Service Architect
- uCosminexus Service Platform
- uCosminexus Service Adapter Architect for FTP
- uCosminexus Service Adapter for FTP

また、オペレーティングシステム (OS) の種類によって、機能が異なる場合があります。OS ごとの違いがある場合の表記方法については、「適用 OS の違いによる機能相違点の表記」を参照してください。

対象読者

Cosminexus サービスプラットフォームを利用して、SOA に対応したシステムを構築するシステム開発者、およびそのシステムの運用管理者を対象としています。

対象読者の前提知識として必要な知識を次に示します。

- SOA に関する基本的な知識
- J2EE に関する基本的な知識
- リレーショナルデータベースおよびデータベース操作言語 (SQL) に関する基本的な知識
- XML に関する基本的な知識

なお、JP1 との連携機能を使用する場合は、次の内容を理解していることが必要です。

- JP1 の統合管理、ジョブ管理、ネットワーク管理、およびアベイラビリティ管理に関する基本的な知識

適用 OS の違いによる機能相違点の表記

このマニュアルは、適用 OS が Windows, AIX, HP-UX および Linux の製品に対応するため、OS の種類によって、機能が異なる場合があります。OS ごとの違いがある場合は、次に示す表記を使用して、それぞれの説明に OS 名を明記しています。

表記	意味
Windows の場合	Windows に該当する表記です。
AIX の場合	AIX に該当する表記です。
HP-UX の場合	HP-UX に該当する表記です。
Linux の場合	Linux に該当する表記です。
UNIX の場合	UNIX (AIX , HP-UX および Linux) に該当する表記です。

このマニュアルで使用している記号

! 注意事項

間違いやすい点，および間違うとトラブルが発生するような項目について説明しています。

ポイント

その説明の要点について説明しています。

参考

補足的な情報を説明しています。

目次

1	Cosminexus サービスプラットフォームの機能概要	1
1.1	開発環境の機能概要	2
1.1.1	電文フォーマットの作成機能	2
1.1.2	サービスアダプタの作成機能	3
1.1.3	DB アダプタの作成機能	4
1.1.4	ビジネスプロセスの作成機能	5
1.1.5	データ変換定義の作成機能	8
1.1.6	ユーザ定義受付の作成機能	10
1.1.7	配備定義機能	11
1.1.8	パッケージング機能	13
1.1.9	配備から開始または停止から削除の一括実行機能	13
1.1.10	サービスリクエストの作成機能	14
1.1.11	ビジネスプロセスのデバッグ機能	15
1.2	実行環境の機能概要	18
1.2.1	サービス振り分け実行機能	18
1.2.2	各種アダプタ	18
1.2.3	ビジネスプロセス実行機能	18
1.3	運用環境の機能概要	19
1.3.1	実行環境のセットアップ機能	19
1.3.2	HCSC コンポーネントの配備機能	20
1.3.3	実行環境での運用を管理する機能	21
1.3.4	実行履歴の管理機能	23
1.4	HCSC サーバのクラスタ構成	25
1.4.1	ロードバランス機能を利用した HCSC サーバの冗長構成	25
1.4.2	クラスタソフトウェアを利用した HCSC サーバの冗長構成	26
1.5	クラスタ構成における運用環境の構成パターン	30
1.6	分散開発時のリポジトリのインポート	33
1.6.1	コンポーネントインポート機能を利用した分散開発の流れ	33
1.6.2	分散開発時の注意事項	36
2	HCSC サーバのサービス部品呼び出し	37
2.1	サービス部品呼び出しの基本的な流れ	38
2.1.1	基本的な構造	38

2.1.2	サービス部品を呼び出す流れ	41
2.1.3	サービス部品側と異なる電文フォーマットでサービス部品を呼び出す場合の流れ	47
2.1.4	サービス部品呼び出し処理で与えられる識別情報	50
2.2	Web サービス (SOAP 通信) を使ったサービス部品呼び出し	53
2.2.1	Web サービスの通信基盤との関係	53
2.2.2	サービス部品呼び出し時のユーザ電文と WSDL の関係	55
2.2.3	Web サービス (SOAP 通信) の場合のトランザクション	63
2.2.4	Web サービス (SOAP 通信) の場合の Cookie 情報	63
2.2.5	Web サービス (SOAP 通信) 場合の SOAP メッセージの構成	67
2.2.6	Web サービス (SOAP 通信) の場合の接続先動の変更	69
2.3	SessionBean を使ったサービス部品呼び出し	73
2.3.1	Cosminexus (EJB コンテナ) との関係	73
2.3.2	サービス部品呼び出し時のユーザ電文とサービス部品のスタブとの関係	73
2.3.3	SessionBean 使用時のトランザクション	75
2.4	MDB (WS-R) を使ったサービス部品呼び出し	78
2.4.1	Cosminexus RM との関係	78
2.4.2	サービス部品呼び出し時のメッセージの流れ	78
2.4.3	同期のサービス部品を呼び出す場合の応答電文の扱い	85
2.4.4	MDB (WS-R) 使用時のトランザクション	87
2.4.5	デッドメッセージキュー	93
2.4.6	MDB (WS-R) で使用するキューの名称と個数	93
2.5	MDB (DB キュー) を使ったサービス部品呼び出し	94
2.5.1	Cosminexus RM との関係	94
2.5.2	サービス部品呼び出し時のメッセージの流れ	96
2.5.3	同期のサービス部品を呼び出す場合の応答電文の扱い	103
2.5.4	MDB (DB キュー) 使用時のトランザクション	105
2.5.5	デッドメッセージキュー	106
2.5.6	MDB (DB キュー) で使用するキューの名称と個数	106
3	標準プロトコル以外のシステムとの接続	107
3.1	データベースとの接続	108
3.1.1	DB アダプタを使ったデータベースへのアクセス	108
3.1.2	DB アダプタが使用する DB Connector の設定	111
3.2	OpenTP1 との接続	115
3.2.1	TP1 から同期の標準受付を使った HCSC サーバのサービス部品の呼び出し	115
3.2.2	TP1 から非同期の標準受付を使った HCSC サーバのサービス部品の呼び出し	117

3.2.3	TP1 から TP1/RPC 受付を使った HCSC サーバのサービス部品の呼び出し	117
3.2.4	HCSC サーバから TP1 アダプタを使った TP1 サービス部品の呼び出し	119
3.2.5	HCSC サーバからの MDB (DB キュー) サービスアダプタを使った TP1 サービス部品の呼び出し	121
3.3	Object Wrapper システムとの接続	122
3.4	他社 ERP パッケージとの接続	123
3.5	ファイルを扱うシステムとの接続	125
3.6	メッセージキューとの接続	127
3.7	FTP クライアントおよび FTP サーバとの接続	128
3.8	カスタム受付を使用した場合のシステムとの接続	131
3.9	カスタムアダプタを使用した場合のシステムとの接続	134

4

	ビジネスプロセスを使った運用	139
4.1	ビジネスプロセスの機能	140
4.1.1	受付・応答	140
4.1.2	サービス呼び出し	141
4.1.3	Java 呼び出し	142
4.1.4	データ変換	143
4.1.5	代入	146
4.1.6	繰り返し	147
4.1.7	分岐	148
4.1.8	並列処理	150
4.2	ビジネスプロセスの仕組み	152
4.2.1	ビジネスプロセスの処理の流れ	152
4.2.2	ビジネスプロセスの通信モデル	153
4.2.3	プロセスインスタンス	154
4.2.4	アクティビティ	157
4.2.5	変数の種類と役割	160
4.2.6	関連セットの役割	160
4.3	永続化するビジネスプロセスと永続化しないビジネスプロセス	161
4.3.1	永続化するビジネスプロセス	161
4.3.2	永続化しないビジネスプロセス	161
4.4	ビジネスプロセスのトランザクション	164
4.4.1	ビジネスプロセスの状態の互換性の指定を ON にした場合のトランザクション	165
4.4.2	ビジネスプロセスの状態の互換性の指定を OFF にした場合のトランザクション	174
4.5	ビジネスプロセスの再実行	183

4.5.1	運用環境の画面からの再実行	183
4.5.2	コマンドからの再実行	183
4.5.3	サービスリクエストからの再実行 (SOAP 通信)	183
4.5.4	サービスリクエストからの再実行 (SessionBean)	184
4.6	ビジネスプロセスを使用した待機処理	186

5

	システム動作を円滑にするための機能	189
5.1	HCSC サーバに関連するタイマの設定	190
5.1.1	HCSC サーバで設定できるタイマ	190
5.1.2	サービス部品のタイムアウト値の変更	198
5.1.3	タイマ設定時の注意事項	199
5.2	サービス部品の接続先変更	200
5.3	SOAP Fault 動作定義ファイルによるフォルトのキャッチ	201
5.4	アプリケーションからの稼働状態の確認	204
5.4.1	稼働状態の確認の概要	204
5.4.2	稼働状態の確認の流れ	204
5.4.3	稼働状態の確認時の注意事項	205
5.5	XML 電文内の属性値に指定した名前空間接頭辞の補完	207
5.6	任意形式 (any 形式) の電文フォーマットの定義	208
5.6.1	任意形式 (any 形式) を設定した場合の処理の流れ	208
5.6.2	データ変換をしない場合の変数 (メッセージ型) の組み合わせ	210

6

	性能向上のための機能	215
6.1	サービス部品呼び出し処理の多重度	216
6.1.1	HCSC サーバ内の多重度	216
6.1.2	データベースへのアクセスに関する多重度	219
6.1.3	Web サービス (SOAP 通信) に関する多重度	223
6.2	XML 電文のキャッシュ機能	227

7

	障害調査のための機能	229
7.1	障害情報の取得・障害回復・障害解析機能	230
7.1.1	障害情報の取得機能	230
7.1.2	障害回復機能	230
7.1.3	障害解析機能	230

7.2	データ検証機能	231
7.2.1	データ検証機能の概要	231
7.2.2	データ検証機能の設定	232
7.2.3	データ検証機能の有効範囲	233
7.3	ユーザ電文トレース機能	234
7.3.1	ユーザ電文トレース機能の概要	234
7.3.2	ユーザ電文トレース機能の設定	234
7.3.3	ユーザ電文トレース機能使用時の注意事項	234

8

	他製品と連携して利用する機能	237
8.1	JP1 と連携した運用管理	238
8.1.1	JP1/IM との連携による障害検知	238
8.1.2	JP1 によるプロセス監視	239
8.2	接続できるデータベース	243

付録

	付録 A このマニュアルの参考情報	245
	付録 A.1 関連マニュアル	246
	付録 A.2 このマニュアルでの表記	249
	付録 A.3 英略語	252
	付録 A.4 KB (キロバイト) などの単位表記について	253
	付録 B 用語解説	254

索引

	索引	265
--	----	-----

1

Cosminexus サービスプラットフォームの機能概要

この章では、Cosminexus サービスプラットフォームで SOA を適用したシステムを実現するための機能の概要について説明します。

-
- 1.1 開発環境の機能概要
 - 1.2 実行環境の機能概要
 - 1.3 運用環境の機能概要
 - 1.4 HCSC サーバのクラスタ構成
 - 1.5 クラスタ構成における運用環境の構成パターン
 - 1.6 分散開発時のリポジトリのインポート
-

1.1 開発環境の機能概要

Cosminexus サービスプラットフォームの開発環境は、実行環境で動かすアダプタやビジネスプロセスなどの定義をするための環境です。開発環境では、主に次の機能を提供しています。

- 電文フォーマットの作成機能
- サービスアダプタの作成機能
- DB アダプタの作成機能
- ビジネスプロセスの作成機能
- データ変換定義の作成機能
- ユーザ定義受付の作成機能
- ビジネスプロセスのデバッグ機能

これらの機能は、開発環境の画面を利用して定義します。

また、開発環境の画面を利用して作成した定義の整合性を検証する機能も備えています。HCSC コンポーネントの検証機能は、HCSC コンポーネントの作成中や作成後など、任意の段階で実行できます。

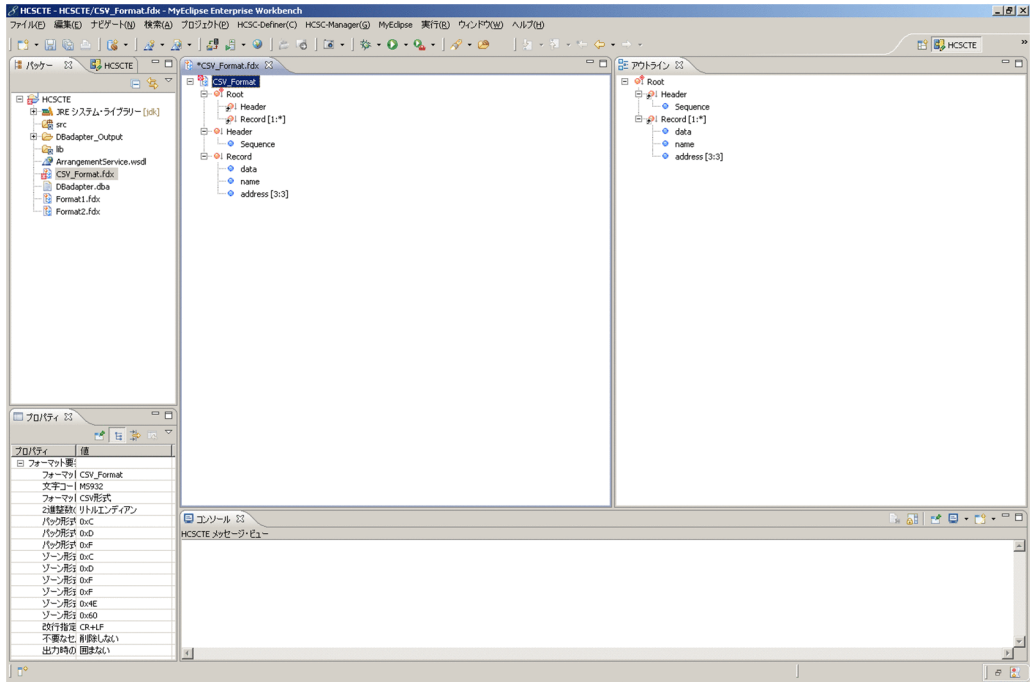
各機能の概要を次に説明します。

1.1.1 電文フォーマットの作成機能

電文フォーマットとは、サービスリクエスタ、サービスアダプタ、およびサービス部品の間でやり取りされる電文の形式です。利用するデータの形式が XML 形式の場合は XML 形式の電文フォーマットを、バイナリ形式（XML 形式以外）の場合はバイナリ形式の電文フォーマットを作成します。バイナリ形式の電文フォーマットの場合は、画面を利用して作成できます。

バイナリ形式の電文フォーマットの作成に利用する画面の例を次の図に示します。

図 1-1 バイナリ形式の電文フォーマットを作成する画面の例



バイナリ形式の電文フォーマットは、ウィザードで新規追加したあと、図 1-1 の画面で、電文フォーマットを構成する要素の詳細な内容を定義できます。

なお、すでに作成されているバイナリ形式の電文フォーマットを流用して、新たなバイナリ形式の電文フォーマットを作成する場合には、既存のバイナリ形式の電文フォーマットを複製して使用することもできます。

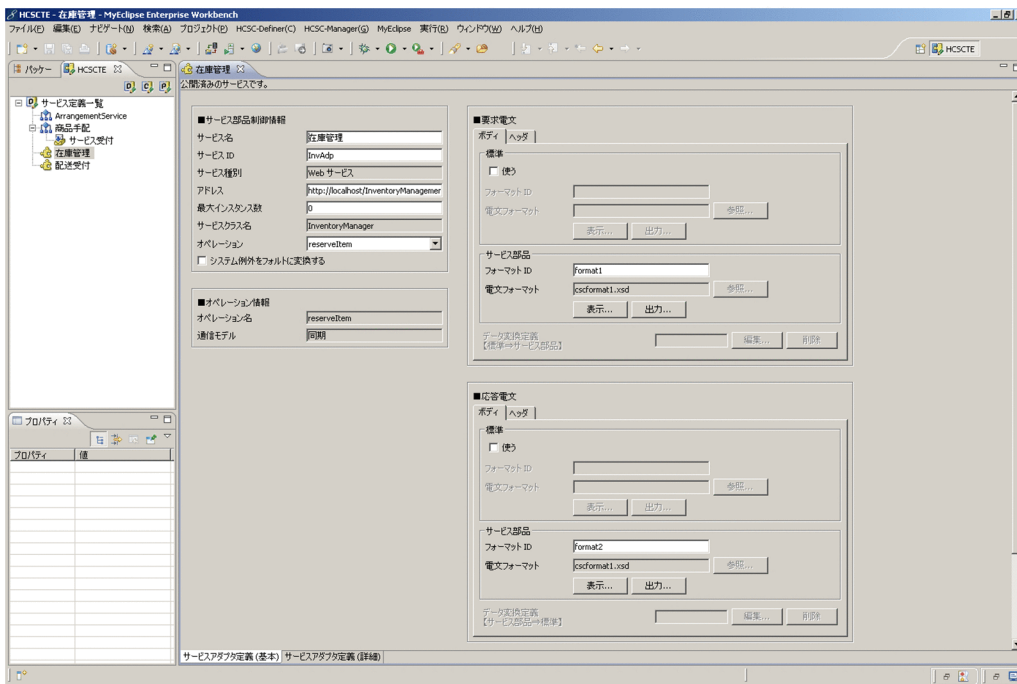
1.1.2 サービスアダプタの作成機能

サービスアダプタとは、呼び出すサービス部品を定義した HCSC コンポーネントです。呼び出すサービス部品の種類（Web サービス、SessionBean、MDB (WS-R)、MDB (DB キュー)）ごとに定義する内容は異なります。

サービスアダプタの作成に利用する画面の例を次の図に示します。

1. Cosminexus サービスプラットフォームの機能概要

図 1-2 サービスアダプタを作成する画面の例



サービスアダプタは、ウィザードで新規追加したあと、図 1-2 の画面で詳細な内容を定義します。

なお、すでに作成されているサービスアダプタを流用して、新たなサービスアダプタを作成する場合には、既存のサービスアダプタを複製して使用することもできます。

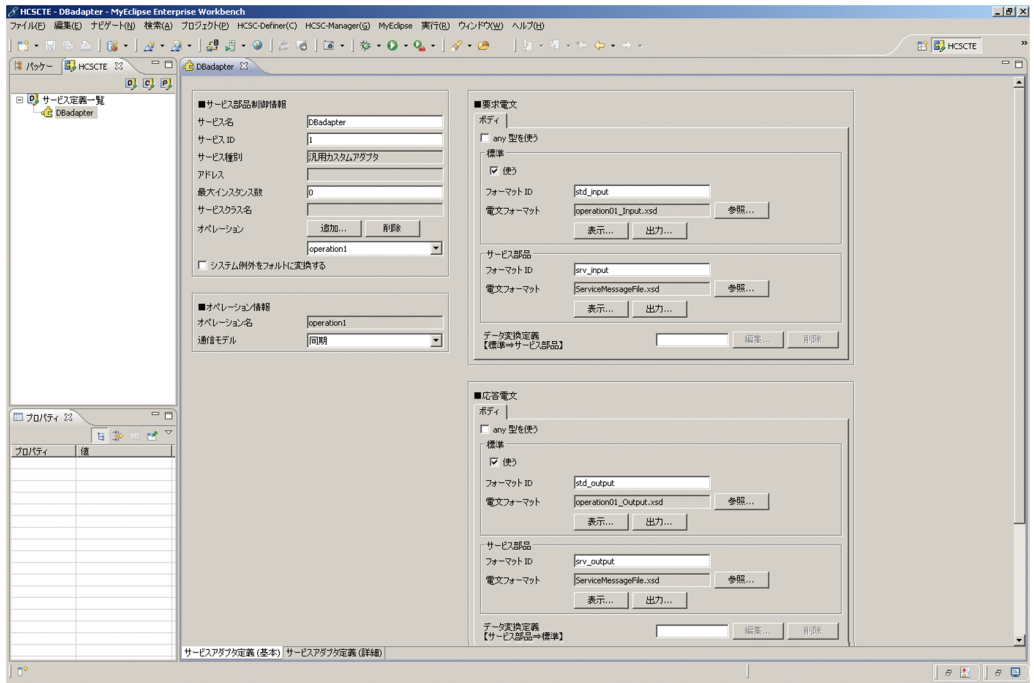
1.1.3 DB アダプタの作成機能

DB アダプタとは、データベースに対して SQL を実行することをサービス部品の一つとして利用する場合に、アクセスするデータベースや実行する SQL を定義した HCSC コンポーネントです。

実行する SQL 命令などを定義ファイルに定義してから、画面を利用して作成します。

DB アダプタの作成に利用する画面の例を次の図に示します。

図 1-3 DB アダプタを作成する画面の例



DB アダプタは、ウィザードで新規追加したあと、図 1-3 の画面で詳細な内容を定義します。

なお、すでに作成されている DB アダプタを流用して、新たな DB アダプタを作成する場合には、既存の DB アダプタを複製して使用することもできます。

1.1.4 ビジネスプロセスの作成機能

ビジネスプロセスとは、複数のサービスの処理の順番や条件などを定義して、一連の業務の流れとして定義したものです。

画面で定義したビジネスプロセスは BPEL として記述されます。

ビジネスプロセスの作成に利用する画面の例を次の図に示します。

(1) ビジネスプロセスの実行状態の管理

ビジネスプロセスを新規追加する時には、実行環境でのビジネスプロセスの実行状態を管理するかどうかを設定する機能があります。

ビジネスプロセスの実行状態の管理を設定すると、ビジネスプロセスの実行途中に障害が発生した場合に、ビジネスプロセスのどの段階で障害が発生し、どこまでが実行されたのか把握できます。停止したビジネスプロセスの状況を把握することで、ビジネスプロセス内の特定の処理から再度実行できます。

ビジネスプロセスの実行状態の管理については、「1.3.4 実行履歴の管理機能」を参照してください。

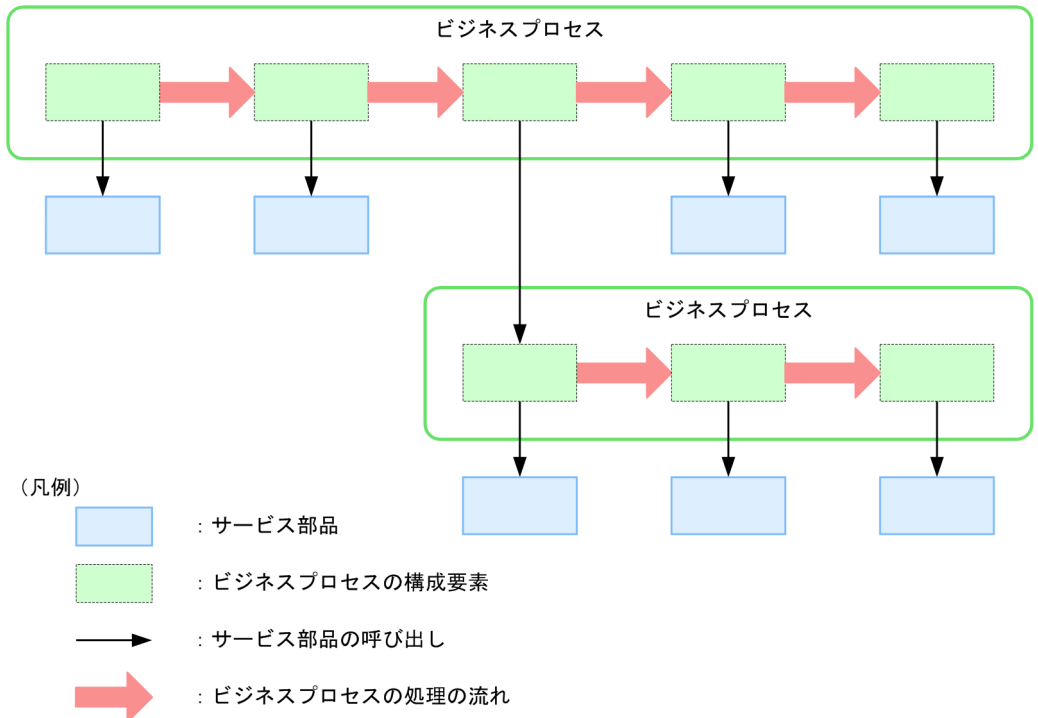
なお、ビジネスプロセスの実行状態を管理するかどうかを選択できるので、処理の速度などの性能を重視するビジネスプロセスでは、ビジネスプロセスの実行状態を管理しない設定もできます。

(2) ビジネスプロセスの階層化

あるビジネスプロセスからほかのビジネスプロセスを呼び出すように定義して、ビジネスプロセスを階層化できます。

ビジネスプロセスの階層化の概要を次の図に示します。

図 1-5 ビジネスプロセスの階層化



1. Cosminexus サービスプラットフォームの機能概要

図 1-5 のように、ビジネスプロセスから各サービス呼び出す一連の流れの中に、ビジネスプロセスを呼び出す処理を定義できます。

ビジネスプロセスで呼び出す処理としてビジネスプロセスを部品化できるため、より複雑な業務処理もビジネスプロセス化できます。

(3) 上流工程で作成したビジネスプロセス定義のインポート

ビジネスプロセスの設計時に、設計ツール（BPMN を用いた上流設計ツール）を使用して作成した BPEL 定義を、サービスプラットフォームの開発環境（uCosminexus Service Architect）へインポートできます。インポートした BPEL 定義は、ビジネスプロセス定義として変換され、画面上にアクティビティとして表示されます。メッセージやプロセスの詳細化などの編集ができます。

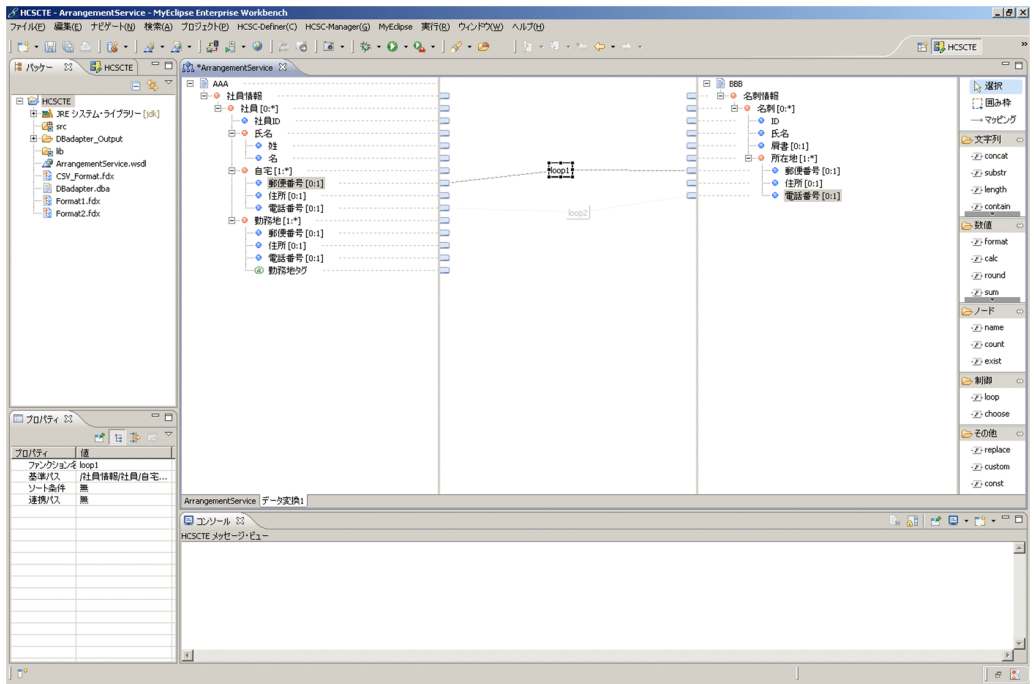
1.1.5 データ変換定義の作成機能

データ変換定義とは、実行環境でサービスリクエストから送信される電文データの変換方法や、サービス部品間で受け渡すデータのフォーマットの変換方法について定義したものです。

変換元 / 変換先のスキーマファイルを作成したあと、作成したスキーマファイルを基に、変換元の要素や属性を変換先のどの要素や属性に変換（マッピング）するかなど、変換時の処理について定義します。

データ変換定義の作成に利用する画面の例を次の図に示します。

図 1-6 データ変換定義を作成する画面の例



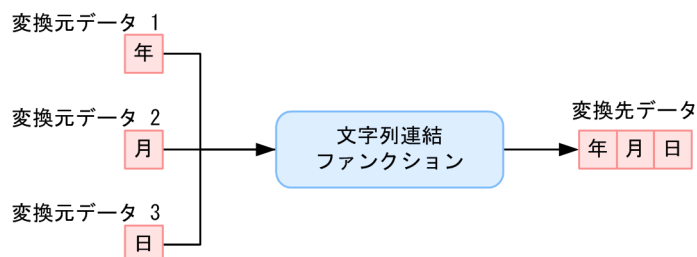
データ変換を定義する画面では、変換元と変換先を接続（マッピング）し、変換元から変換先への値の代入をビジュアルに定義できます。

また、変換元のデータを加工してから変換先に代入したい場合は、ファンクションを設定できます。ファンクションには、文字列の連結や数値演算などの機能があります。

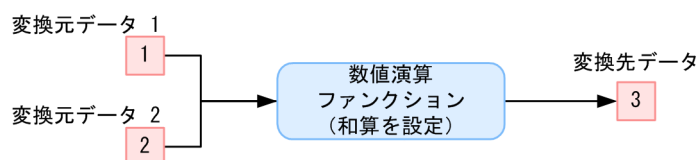
ファンクションを利用したデータの加工の例を次の図に示します。

図 1-7 ファンクションを利用したデータの加工の例

- ・複数の文字列データを連結する場合



- ・複数の数値データを和算する場合



また、変換元データが複数ある場合のくり返し処理や、一定の条件に合致する場合にだけデータを代入するなどの設定もできます。

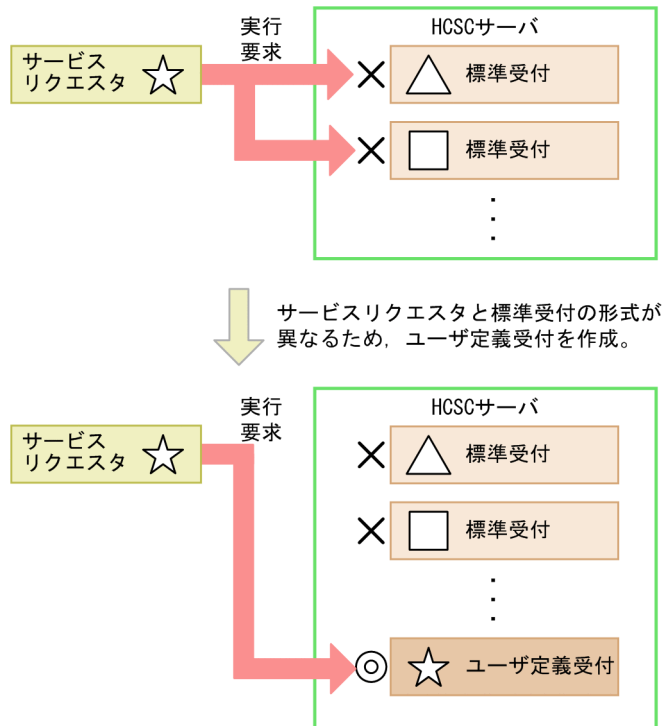
1.1.6 ユーザ定義受付の作成機能

ユーザ定義受付とは、サービスリクエストからの実行要求を受け付けるためのインターフェースです。HCSC サーバの機能として提供している標準受付とは異なり、任意の形式を定義できるインターフェースです。開発環境で定義したあと、HCSC サーバに配備して利用します。

Cosminexus サービスプラットフォームで提供している標準受付は、サービスリクエストからのサービス部品の実行要求を受け付ける固定のインターフェースです。標準受付を利用するには、それぞれの標準受付の形式に合わせてサービスリクエストを作成する必要があります。そのため、既存のサービスリクエストを再利用する場合、標準受付のインターフェースの形式と合わないときには、サービスリクエストを作り直したり、大きく改造したりする必要がありました。

このような場合、既存のサービスリクエストからの実行要求を受け付けられる形式のユーザ定義受付を作成できます。ユーザ定義受付の作成について、次の図に示します。

図 1-8 ユーザ定義受付の作成



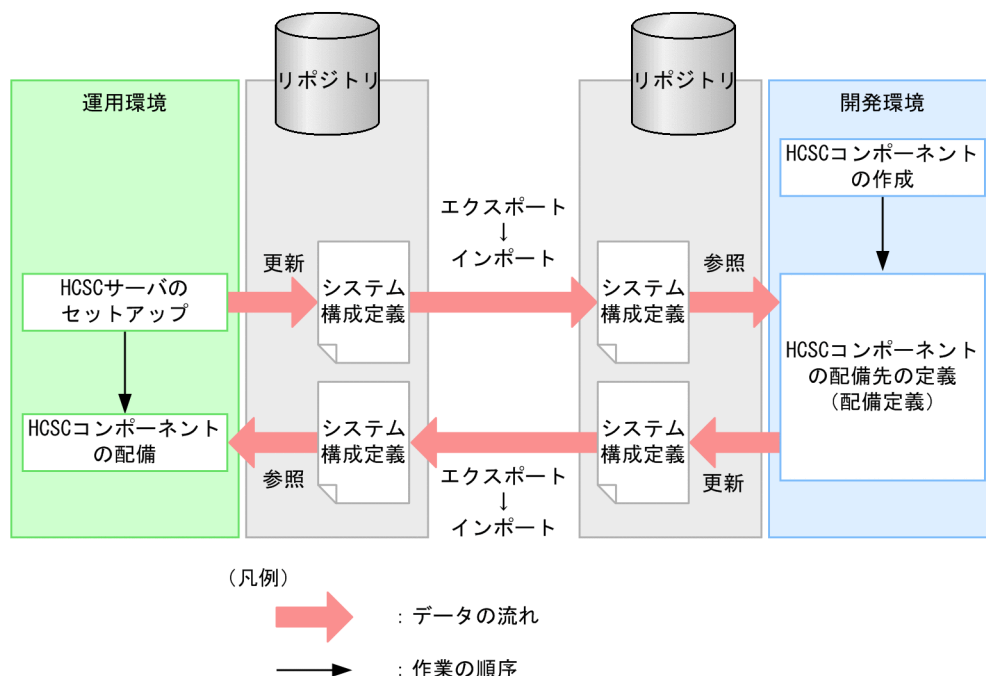
ユーザ定義受付は、サービスリクエスタが呼び出すサービス部品から WSDL を作成し、その WSDL を基に作成します。Cosminexus サービスプラットフォームには、ユーザ定義受付を効率的に作成するためのウィザードが用意されています。

1.1.7 配備定義機能

HCSC コンポーネントの配備は、開発環境と運用環境でリポジトリを介して相互に情報をやり取りしながら実施します。

HCSC コンポーネントの配備の概要について、次の図に示します。

図 1-9 HCSC コンポーネントの配備



運用環境では、HCSC サーバおよびクラスタのセットアップ時に、リポジトリにある実行環境の構成情報（システム構成定義）が更新されます。更新されたシステム構成定義は、リポジトリの管理機能でエクスポートします。

開発環境では、運用環境でエクスポートした HCSC サーバおよびクラスタのセットアップ情報を含むシステム構成定義をリポジトリにインポートし、システム構成定義に HCSC コンポーネントをどの HCSC サーバに配備するかを定義（配備定義）して更新し、更新したシステム構成定義をエクスポートします。

ポイント

開発環境で配備定義するときは、サービス部品をクラスタ、または単一の HCSC サーバ単位で配置します。

そのあと、運用環境では、開発環境で更新したシステム構成定義をリポジトリにインポートします。システム構成定義の内容に従って、実行環境へ HCSC コンポーネントを配備します。

リポジトリのエクスポートについては、マニュアル「Cosminexus サービスプラットフォーム システム構築・運用ガイド」の「4.2 リポジトリのエクスポート」を参照してください。リポジトリのインポートについては、マニュアル「Cosminexus サービスプラットフォーム システム構築・運用ガイド」の「4.3 リポジトリのインポート」を参照

してください。

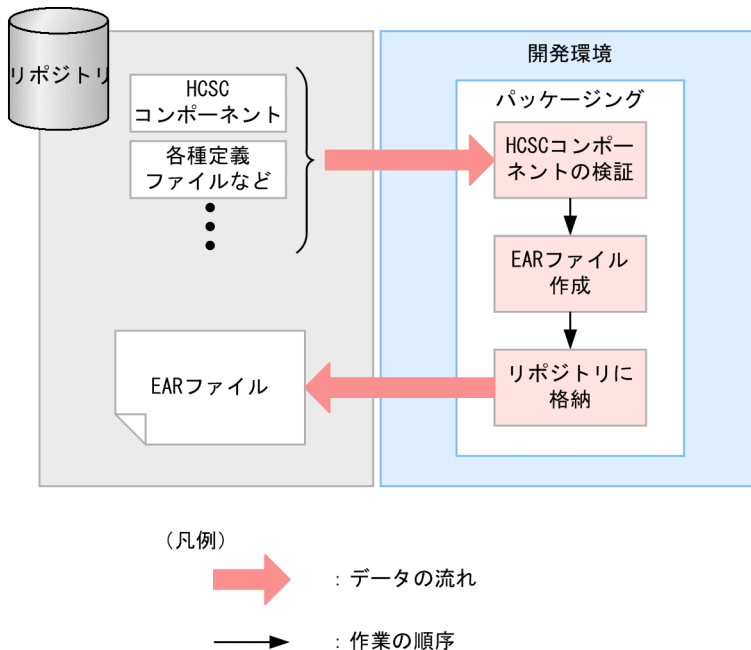
1.1.8 パッケージング機能

パッケージングとは、開発環境で作成した HCSC コンポーネントと必要な情報を EAR ファイル形式で組み立てて、リポジトリに格納することをいいます。

開発環境では、画面を利用してパッケージングを実行できます。

パッケージングの概要を次の図に示します。

図 1-10 パッケージング機能



開発環境からパッケージング機能を実行すると、パッケージングに必要な情報がリポジトリから収集され、HCSC コンポーネントが検証されます。

検証の結果、HCSC コンポーネントに必要な定義がすべて定義されている場合、EAR ファイルが作成されます。

HCSC コンポーネントに不備がある場合、パッケージングは中止されます。

作成された EAR ファイルは、リポジトリに格納されます。運用環境では、リポジトリに格納された EAR ファイル形式の HCSC コンポーネントを実行環境に配備します。

1.1.9 配備から開始または停止から削除の一括実行機能

開発環境では、次の一連の処理を一括で実行できます。

1. Cosminexus サービスプラットフォームの機能概要

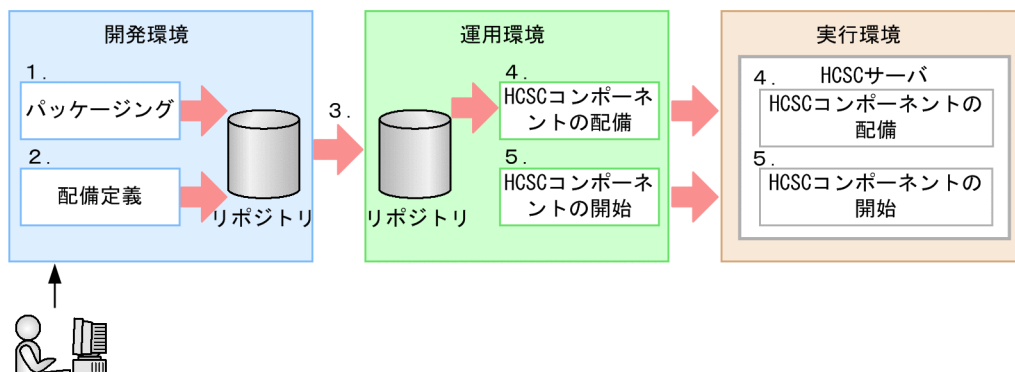
- HCSC コンポーネントを HCSC サーバに配備して開始する
- HCSC コンポーネントを停止して HCSC サーバから削除する

開発環境または運用環境で一つ一つ実施していた作業を開発環境で一括実行するため、ユーザの操作負担を軽減できます。なお、一括実行は、システム開発時、または単体テストから結合テスト時を対象としています。

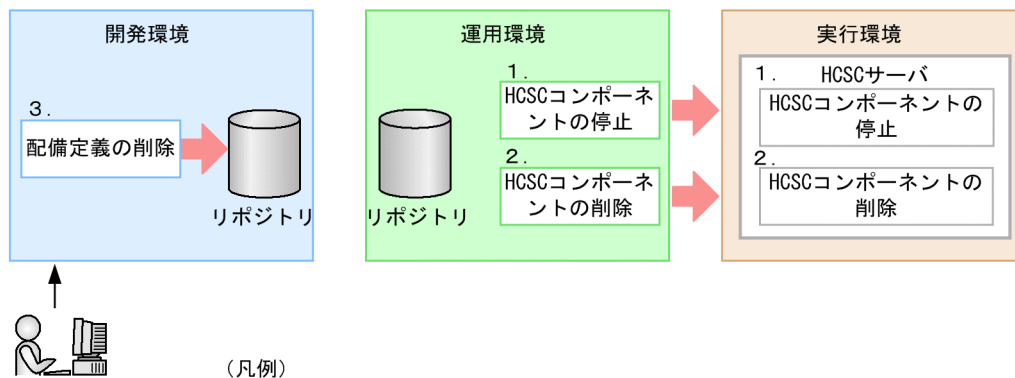
一括実行の処理の流れを次の図に示します。図中の番号は、開発環境でユーザが操作したあとに、自動的に実行される処理の順番を示しています。

図 1-11 一括実行の処理の流れ

●HCSCコンポーネントをHCSCサーバに配備して開始する場合



●HCSCコンポーネントを停止してHCSCサーバから削除する場合



(凡例)

→ : 処理の流れ

→ : ユーザの操作

1.1.10 サービスリクエストの作成機能

サービスリクエストとは、業務担当者からのサービス部品の実行要求を受け付け、各 HCSC コンポーネントにサービス部品の実行要求を送信するアプリケーションです。

サービスリクエストは、開発環境で作成し、実行環境に組み込んで利用します。

Cosminexus サービスプラットフォームの開発環境の画面には、サービスリクエストの作成に必要な WSDL やスタブを取得するなど、サービスリクエストの作成を支援する機能があります。

1.1.11 ビジネスプロセスのデバッグ機能

ビジネスプロセスのデバッグ機能とは、ビジネスプロセスに着目したテストおよびデバッグを実行できる機能です。

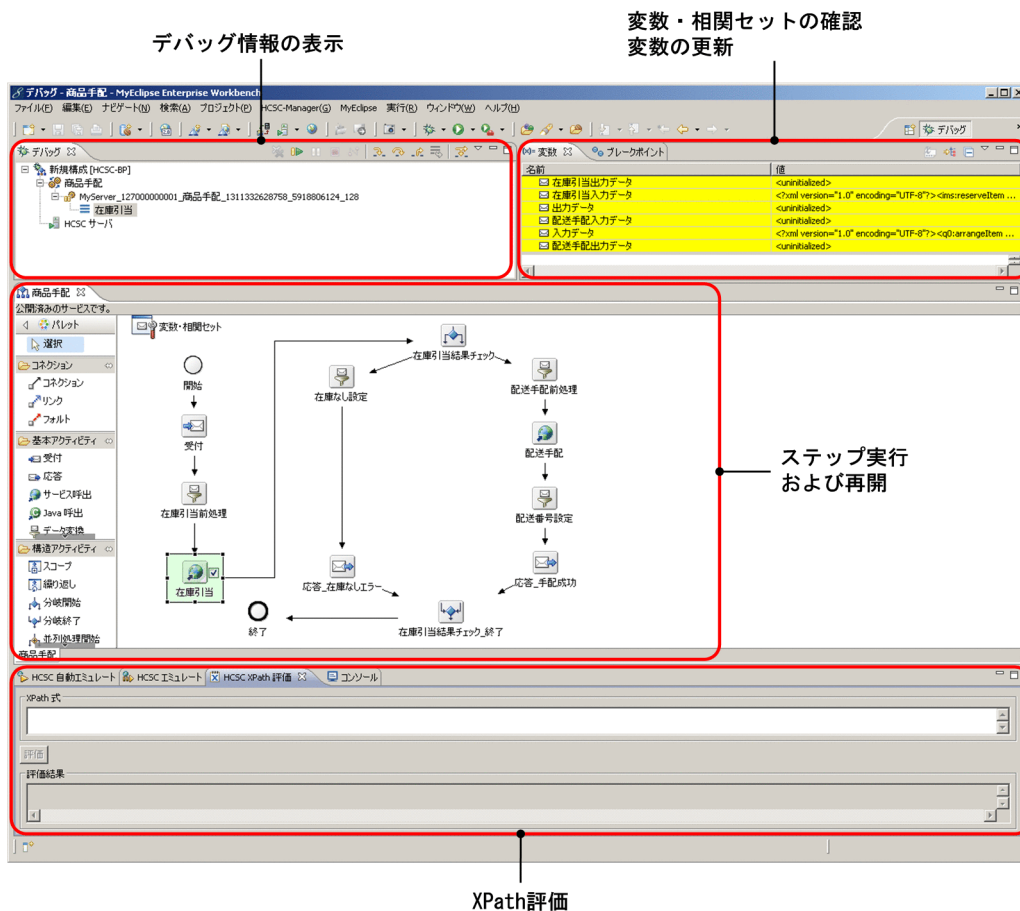
ビジネスプロセスのデバッグ機能を利用することで、HCSC コンポーネントを実行環境に配備しなくても、開発環境上でビジネスプロセスをデバッグできます。ビジネスプロセスの定義およびデバッグが同じ環境上で行えるため、開発効率を向上できます。

ビジネスプロセスのデバッグでは、ビジネスプロセスの処理がアクティビティ間をどのように進んでいるかを確認したり、アクティビティで使用している変数の中身を確認・更新したりできます。

ビジネスプロセスのデバッグで利用する画面の例と利用できる機能を次の図に示します。

1. Cosminexus サービスプラットフォームの機能概要

図 1-12 ビジネスプロセスのデバッグで利用する画面の例と利用できる機能



ビジネスプロセスのデバッグで利用できる機能を次に示します。

- ステップ実行および再開
アクティビティにブレークポイントを設定することで、ビジネスプロセスの処理を任意のタイミングで中断できます。中断しているビジネスプロセスでは、アクティビティ単位で中断しながら、逐次処理を進めることができます。
- デバッグ情報の表示
デバッグ中のビジネスプロセスの名称、およびアクティビティの名称を表示します。
- 変数・相関セットの確認
ビジネスプロセスで使用している変数および相関セットの値を確認します。
- 変数の更新
ビジネスプロセスで使用している変数の値を変更し、ビジネスプロセスに反映します。
- XPath 評価
分岐アクティビティや代入アクティビティに指定した条件式の妥当性を評価します。

ビジネスプロセスのデバッグ機能では、サービスおよびサービスリクエストをエミュ

レートして利用することもできます。

1.2 実行環境の機能概要

Cosminexus サービスプラットフォームの実行環境は、SOA を実現するための環境です。実行環境では、主に次の機能を提供しています。

- サービス振り分け実行機能
- 各種アダプタ
- ビジネスプロセス実行機能

各機能の概要を次に説明します。

1.2.1 サービス振り分け実行機能

業務・機能を“サービス”として管理し、要求に対してサービス振り分けを実行します。

サービス振り分け実行機能の詳細については、「2. HCSC サーバのサービス部品呼び出し」を参照してください。

1.2.2 各種アダプタ

さまざまな種類のサービスを呼び出すためのアダプタを提供します。

各種アダプタの詳細については、「3. 標準プロトコル以外のシステムとの接続」を参照してください。

1.2.3 ビジネスプロセス実行機能

サービス呼び出しの流れを制御するためのビジネスプロセスの開発、および実行を行います。

ビジネスプロセスの詳細については、「4. ビジネスプロセスを使った運用」を参照してください。

1.3 運用環境の機能概要

Cosminexus サービスプラットフォームの運用環境は、実行環境の構築、必要な情報の配置、運用管理、および実行履歴の管理をする環境です。運用環境では、主に次の機能を提供しています。

- 実行環境のセットアップ機能
- HCSC コンポーネントの配備機能
- 実行環境での運用を管理する機能
- 実行履歴の管理機能

これらの機能は、コマンドまたは運用環境の画面を利用して実行します。

各機能の概要を次に説明します。

1.3.1 実行環境のセットアップ機能

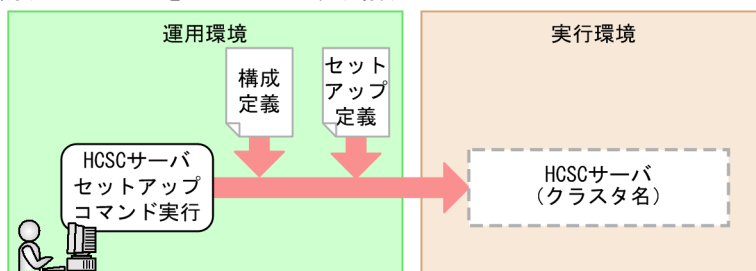
運用環境では、コマンドを利用して、実行環境の HCSC サーバをセットアップできます。また、HCSC サーバをセットアップするときに、あわせてクラスタもセットアップできます。

(1) HCSC サーバおよびクラスタのセットアップ機能

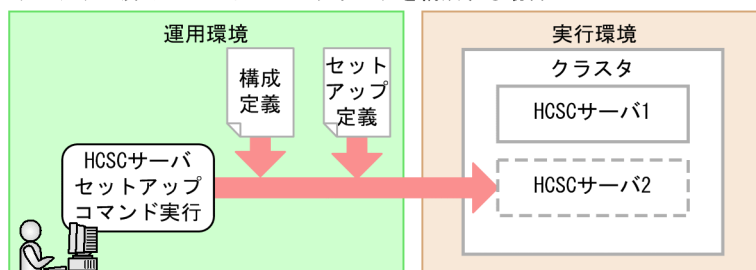
HCSC サーバのセットアップの概要について、次の図に示します。

図 1-13 HCSC サーバのセットアップの概要

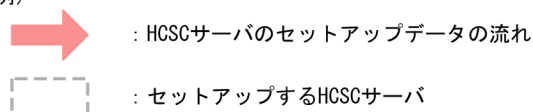
- ・ 単独の HCSC サーバをセットアップする場合



- ・ セットアップ済みの HCSC サーバとクラスタを構成する場合



(凡例)



運用環境からコマンドを実行すると、あらかじめ作成していた HCSC サーバの構成定義ファイルとセットアップ定義ファイルからセットアップに必要な情報を取得して HCSC サーバがセットアップされます。

単独の HCSC サーバをセットアップする場合

コマンド実行時にクラスタ名を指定しますが、この段階では、単独の HCSC サーバであるため、クラスタの構成にはなりません。

ほかの HCSC サーバをセットアップするときに、クラスタ名を指定してセットアップすることで、クラスタを構成できます。

セットアップ済みの HCSC サーバとクラスタを構成する場合

コマンド実行時に、ほかの HCSC サーバをセットアップしたときに指定したクラスタ名を指定すると、すでにセットアップされている HCSC サーバとクラスタを構成する HCSC サーバがセットアップできます。

1.3.2 HCSC コンポーネントの配備機能

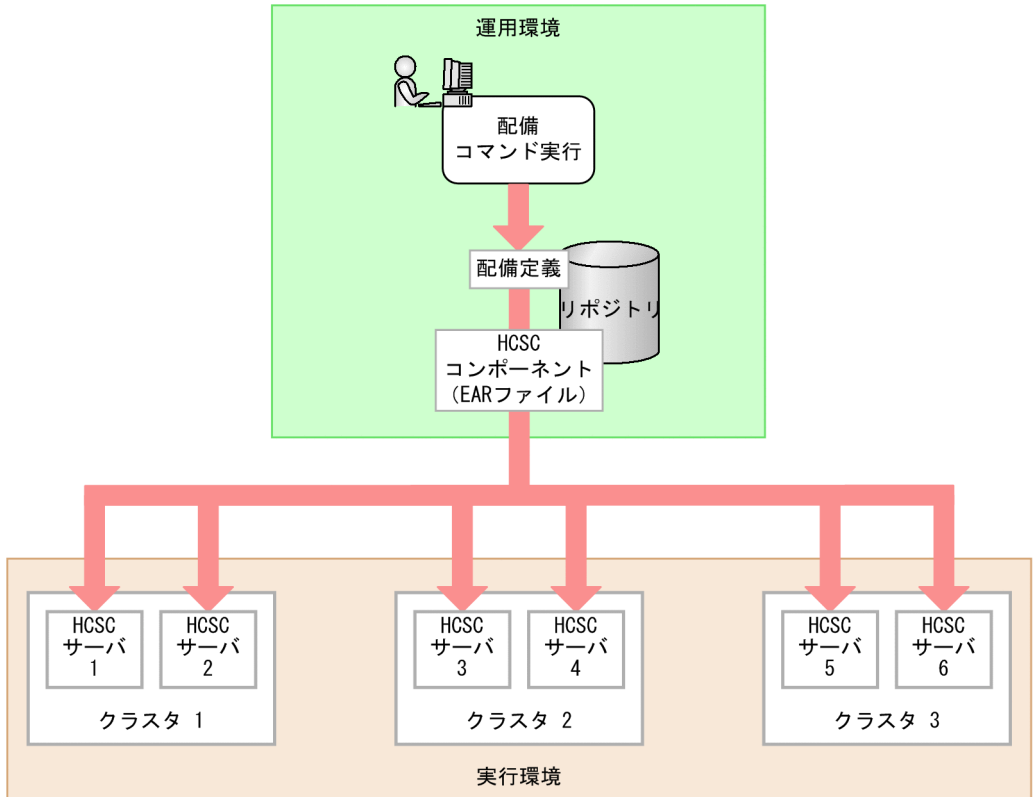
配備とは、HCSC コンポーネントを実行環境で利用できるよう、実行環境に必要な情報

を配置することです。


HCSC コンポーネントは、運用環境でコマンドを実行して実行環境に配備できます。

配備の概要について、次の図に示します。

図 1-14 配備の概要



(凡例)

 : 配備するデータの流れ

HCSC コンポーネントを配備する場合、開発環境で作成してリポジトリに格納されている配備定義の情報に従って、HCSC コンポーネント (EAR ファイル) を実行環境 (HCSC サーバ) に配備します。

1.3.3 実行環境での運用を管理する機能

運用環境では、実行環境での業務の運用を管理できます。

実行環境での業務を開始する場合や、実行環境の構成を変更するために業務を停止する場合、また障害の発生によって業務を停止する場合など、必要に応じて運用環境で実行

1. Cosminexus サービスプラットフォームの機能概要

環境の業務の運用状態を確認し、操作できます。

運用環境から実施できる実行環境の操作について、次の表に示します。

表 1-1 運用環境から実施できる実行環境の操作

操作内容	実行インターフェース	
	画面	コマンド
HCSC サーバのセットアップ	-	
HCSC サーバのアンセットアップ	-	
HCSC サーバの起動		
HCSC サーバの起動（クラスタ内一括起動）	-	
HCSC サーバの停止		
HCSC サーバの停止（クラスタ内一括停止）	-	
HCSC サーバの情報の参照		
HCSC サーバのセットアップ情報の参照	-	
HCSC サーバの定義	-	
HCSC サーバの定義内容の確認		
HCSC サーバの定義の変更	-	
HCSC サーバのバージョンアップ	-	
標準受付の開始		
標準受付の停止		
サービスアダプタの配備	-	
配備済みサービスアダプタの削除	-	
サービスアダプタの開始		
サービスアダプタの停止		
サービスアダプタの情報の参照		
ビジネスプロセスの配備	-	
配備済みビジネスプロセスの削除	-	
ビジネスプロセスの開始		
ビジネスプロセスの停止		
ビジネスプロセスの情報の参照		
ユーザ定義受付の配備	-	
配備済みユーザ定義受付の削除	-	
ユーザ定義受付の開始		
ユーザ定義受付の停止		
ユーザ定義受付の情報の参照		

操作内容	実行インターフェース	
	画面	コマンド
ユーザ定義受付の定義情報の確認		
ユーザ定義受付の定義情報の変更	-	
HCSC サーバ、標準受付、および HCSC コンポーネントの状態の確認		
HCSC サーバのリソースの稼働状態の表示		
アプリケーションの操作	-	
HCSC-Manager 環境のバックアップ	-	
HCSC-Manager 環境のリストア	-	
サービス情報の変更	-	
サービス情報の確認		

(凡例)

- : 利用できます。
- : 利用できません。

注 画面とコマンドで参照できる情報が異なります。

1.3.4 実行履歴の管理機能

運用環境では、ビジネスプロセスのプロセスインスタンスの実行履歴やサービス部品を呼び出すための電文の実行履歴を管理できます。

(1) プロセスインスタンスの実行履歴の管理

プロセスインスタンスの実行履歴は、ビジネスプロセスのプロセスインスタンスの実行状況を履歴としてデータベースに記録（永続化）して管理します。

(a) プロセスインスタンスの実行履歴の検索

条件を指定して、プロセスインスタンスの実行履歴を検索できます。関連セット、開始日時または状態を検索キーとして利用できます。

(b) プロセスインスタンスの実行履歴の詳細情報の参照

プロセスインスタンスの実行履歴の詳細情報を参照できます。次に示す情報を参照できます。

- 関連セット定義名および関連セット値
- 各プロセスインスタンスが保持する変数の変数定義名、および変数値
- ビジネスプロセス内に定義された各アクティビティの状態

(c) プロセスインスタンスの実行履歴の削除

ビジネスプロセスの確認のために使用するデータベースから不要なプロセスインスタンス

1. Cosminexus サービスプラットフォームの機能概要

スの実行履歴を削除できます。一件ずつ削除することも、状態を指定して一括して削除することもできます。

(d) 実行中のプロセスインスタンスの再実行

フォルトが発生するなどの理由で処理できなかったビジネスプロセスのプロセスインスタンスを確認して、実行が中断したプロセスインスタンスを再実行できます。プロセスインスタンスの再実行は、一件ずつ再実行することも一括して再実行することもできます。

(2) 電文の実行履歴の管理

電文の実行履歴は、サービス部品を呼び出すための電文の実行状況を履歴としてデータベースに記録（永続化）して管理します。

(a) 電文の実行履歴の検索と詳細情報の参照

条件を指定して、電文の実行履歴を検索できます。メッセージ共通 ID、サービスリクエスト ID、開始日時 / 終了日時、サービスリクエストで受け付けた入力サービス名、呼び出し先のサービス名、JMS メッセージ ID、および JMS 関連識別子を検索キーとして利用できます。検索した結果に表示される情報の範囲も、基本的な情報から、詳細情報まで選択できます。表示された情報から、サービスを特定したり、電文の配送状況を追跡したりできます。

(b) 電文の実行履歴の削除

データベースにある実行履歴情報テーブルから、不要になった電文の実行履歴のレコードを削除できます。

1.4 HCSC サーバのクラスタ構成

複数の HCSC サーバを組み合わせることで、より冗長なクラスタ構成の環境を構築できます。

1.4.1 ロードバランス機能を利用した HCSC サーバの冗長構成

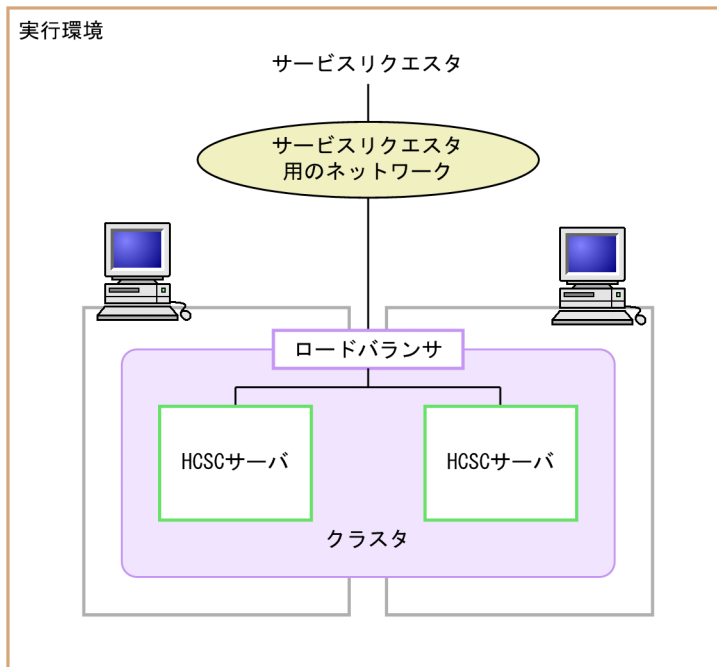
ロードバランス機能とは、複数台の HCSC サーバを並列に並べることで、サービスリクエストからのサービスの実行要求の一点集中による J2EE サーバの負荷を軽減するための機能です。Cosminexus サービスプラットフォームでは、次に示すロードバランサを使用して負荷を分散できます。

- 汎用ロードバランサ
同期受付（Web サービス）の場合に使用します。
- CTM
同期受付（SessionBean）の場合に使用します。

Cosminexus サービスプラットフォームの実行環境では、ロードバランサを使用して、複数の HCSC サーバを組み合わせることで、より冗長なクラスタ構成の環境を構築できます。

HCSC サーバのロードバランスクラスタ構成について、次の図に示します。

図 1-15 HCSC サーバのロードバランスクラスタ構成



1. Cosminexus サービスプラットフォームの機能概要

同期受付（Web サービス、SessionBean）の場合だけ、ロードバランサを使用してサービスリクエストからのサービス部品の実行要求を、複数の HCSC サーバに分散して送信できます。

使用するロードバランサによっては、運用を停止しないで、ロードバランスクラスタを構成する HCSC サーバを追加したり、削除したりできます。また、すでに HCSC サーバに配備済みのサービスアダプタやビジネスプロセスを停止しないで、サービスアダプタやビジネスプロセスを新規追加することもできます。

1.4.2 クラスタソフトウェアを利用した HCSC サーバの冗長構成

クラスタソフトウェアとは、システムの信頼性および稼働率の向上を目的とした、サーバプログラムを含めたシステムの切り替えを実現するプログラムです。Cosminexus サービスプラットフォームでは、次に示すクラスタソフトウェアと連携できます。

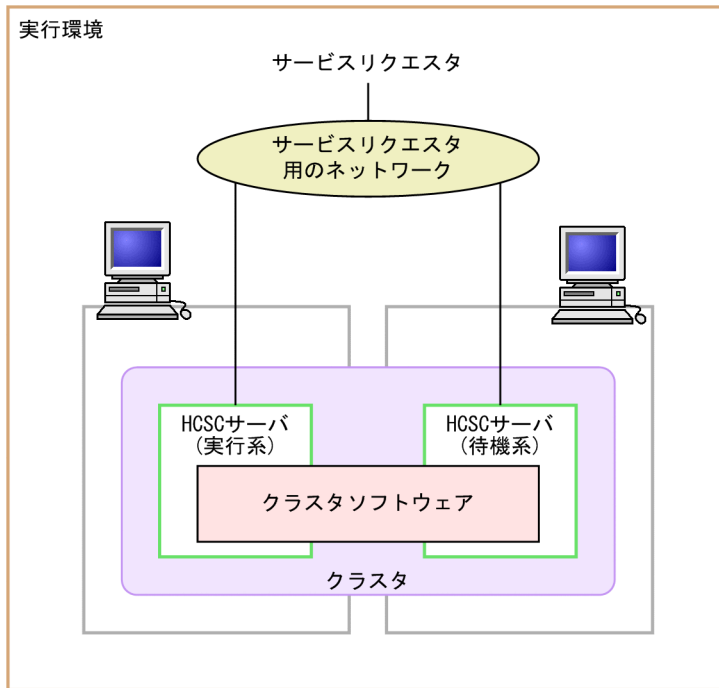
- Microsoft Cluster Service（Windows の場合）
- HA モニタ（UNIX の場合）

Cosminexus サービスプラットフォームの実行環境では、クラスタソフトウェアと連携して、同じ内容の HCSC サーバを二つ組み合わせ、より冗長なクラスタ構成の環境を構築できます。

クラスタを構成する二つの HCSC サーバのセットアップ情報や、配備される HCSC コンポーネントは、同じ内容である必要があります。

HCSC サーバのクラスタ構成について、次の図に示します。

図 1-16 HCSC サーバのクラスタ構成

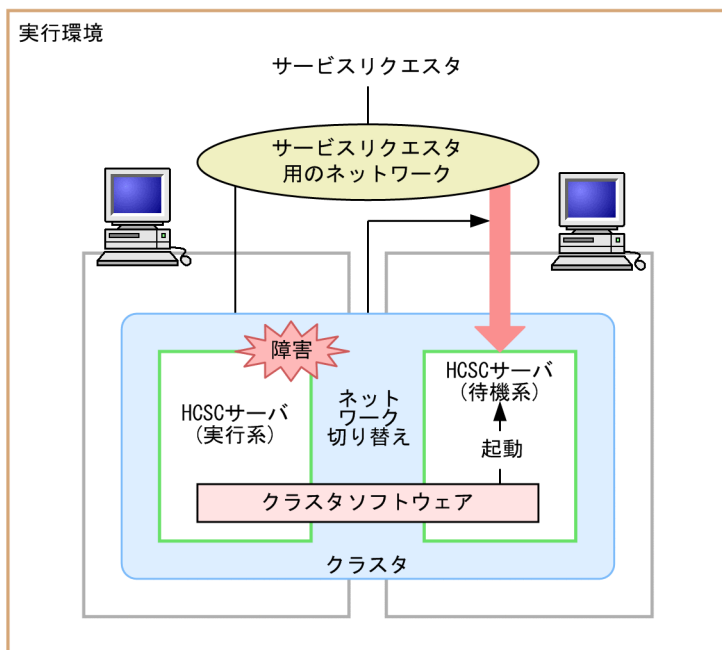
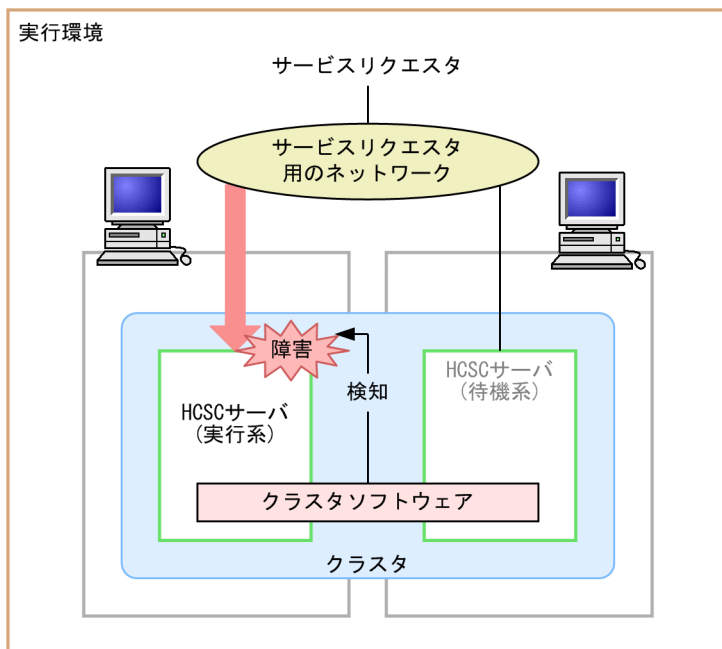




Cosminexus サービスプラットフォームの実行環境では、クラスタソフトウェアと連携して、実行系と待機系を1対1で運用するシステム（1:1系切り替えシステム）を構築できます。

1:1系切り替えシステムでは、実行系に何らかの障害が発生すると、クラスタソフトウェアがこれを検知して、自動的に待機している系に切り替えて業務を続行します。障害発生時の系の切り替えを次の図に示します。

1. Cosminexus サービスプラットフォームの機能概要

図 1-17 障害発生時の系の切り替え



(凡例)  : HCSCサーバへのアクセス  : クラスタソフトウェアによる制御

障害の発生から系の切り替えまでの流れは次のとおりです。

1. 実行系 HCSC サーバで障害が発生します。
2. クラスタソフトウェアが実行系 HCSC サーバの障害を検知します。
3. 実行系 HCSC サーバのあるマシンへのネットワークが切断されます。
4. クラスタソフトウェアがネットワークを待機系 HCSC サーバのあるマシンに切り替えます。
5. クラスタソフトウェアが待機系 HCSC サーバを起動します。

なお、クラスタソフトウェアには、障害を検知した場合に待機系 HCSC サーバを自動的に起動するよう設定しておきます。

1.5 クラスタ構成における運用環境の構成パターン

クラスタ構成にする場合に実現できる運用環境と実行環境の構成パターンには、実行環境の数に関係なく運用環境を 1 台とする構成のほかに、運用環境と実行環境を 1 対 1 に配置する構成があります。

運用環境と実行環境を 1 対 1 に配置する構成のメリットとデメリットを次に示します。

メリット

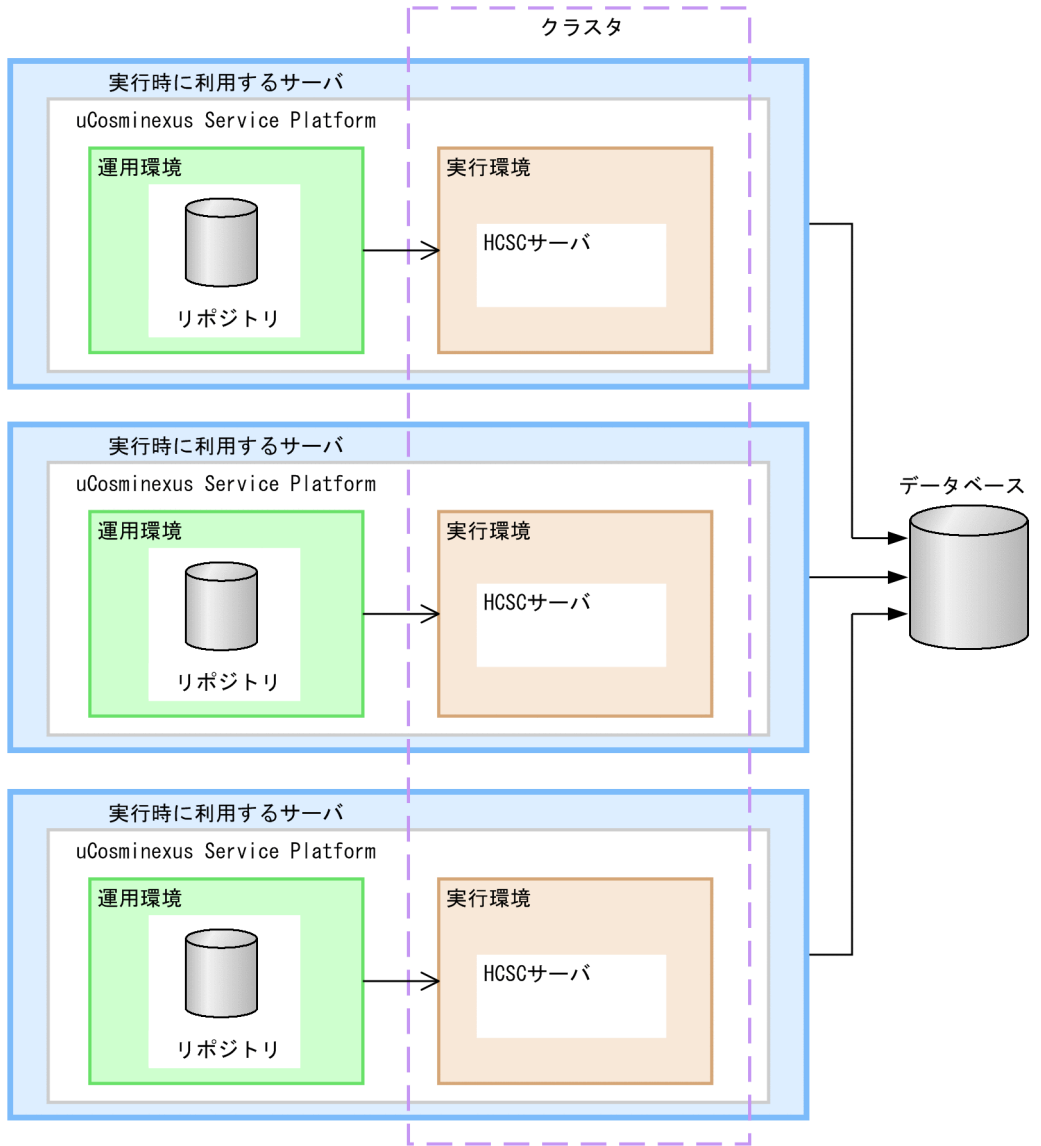
- 実行環境と運用環境を 1 台のマシンにまとめられます。
- 運用環境が 1 台ダウンしても、エラーになったリクエストをほかの運用環境から再実行できます。
- 運用環境用のマシンが不要になるため、運用監視ソフトのライセンス数を低減できます。
- 予備の運用環境が不要になります。

デメリット

- UNIX 上に実行環境を構築する場合、運用環境の GUI を利用できません。
- クラスタ単位の一括操作ができません。

運用・実行 1 対 1 構成の構成例を次の図に示します。

図 1-18 運用・実行 1 対 1 構成の構成例



(凡例)

→ : 起動・停止命令の流れ → : データの流れ

! 注意事項

運用・実行 1 対 1 構成に移行する場合の注意事項

- 移行前の環境が 08-70 以前にセットアップした環境の場合、クラスタに含まれる HCSC サーバをいったんすべてアンセットアップする必要があります。このため、データベースに格納されているプロセスインスタンスや実行履歴など、クラスタで共有する情報は移行できません。
 - データベースを利用していない運用・実行 1 対 1 構成からデータベースを利用した運用・実行 1 対 1 構成には移行できません。
-

1.6 分散開発時のリポジトリのインポート

複数の開発環境で分散開発する場合、開発したリポジトリを運用環境で使用するために、リポジトリを集約する必要があります。リポジトリの集約方法には次の2通りの方法があります。

- 開発環境でサービスインポートを実行する。
- 運用環境でコンポーネントインポートを実行する。

開発環境でリポジトリを集約する場合は、インポート対象の取捨選択やサービス ID の衝突回避など、運用環境で集約するよりも詳細に条件を設定できます。

運用環境でリポジトリを集約する場合は、リポジトリを集約するための開発環境が不要になるため、マシンの台数を削減できます。

サービスインポートの詳細についてはマニュアル「Cosminexus サービスプラットフォーム 開発ガイド」の「3.2.3 リポジトリをインポートする」を、コンポーネントインポートの詳細についてはマニュアル「Cosminexus サービスプラットフォーム システム構築・運用ガイド」の「4.6 コンポーネント単位でのリポジトリの追加、更新、および削除」を参照してください。

1.6.1 コンポーネントインポート機能を利用した分散開発の流れ

コンポーネントインポート機能を利用すると、開発環境を用意しないで、運用コマンドだけでリポジトリを集約できます。このため、開発環境（マスタ）とテスト環境を1台のマシンに集約でき、マシンの台数を削減できます。

開発環境でリポジトリを集約する場合の流れを図 1-19 に、運用環境でリポジトリを集約する場合の流れを図 1-20 にそれぞれ示します。

図 1-19 開発環境でリポジトリを集約する場合の流れ

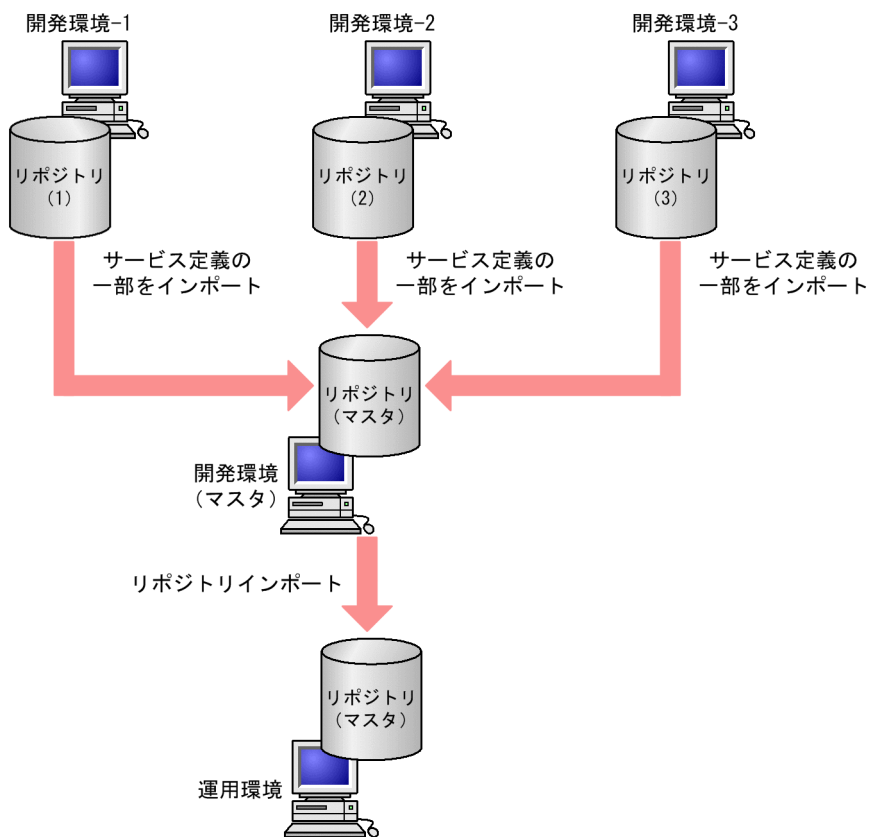
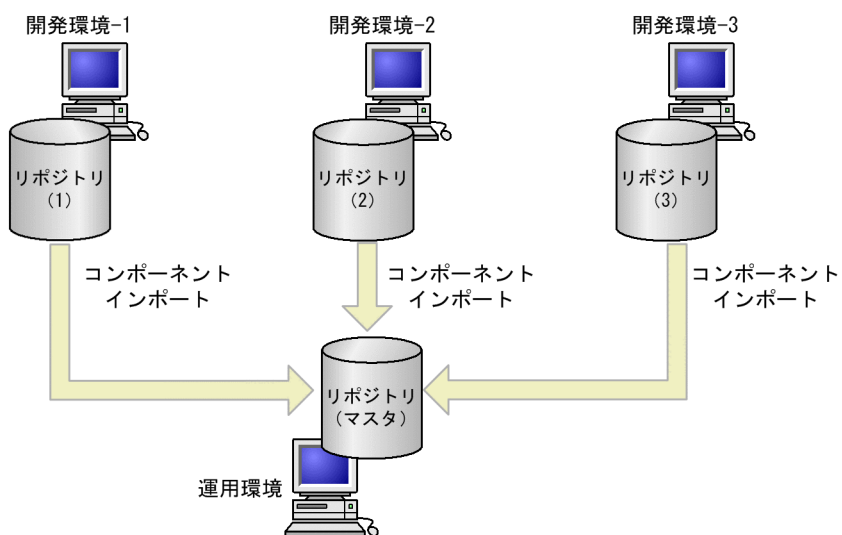


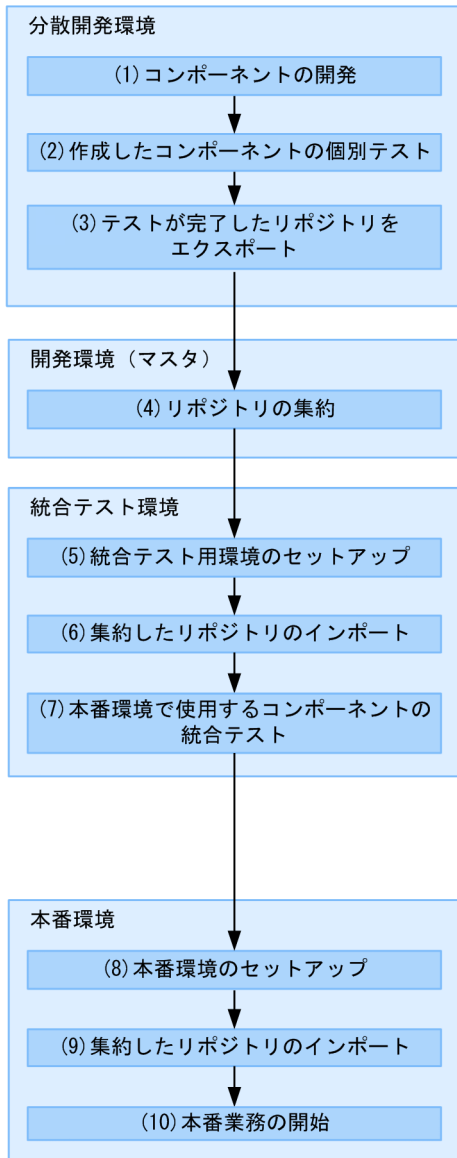
図 1-20 運用環境でリポジトリを集約する場合の流れ



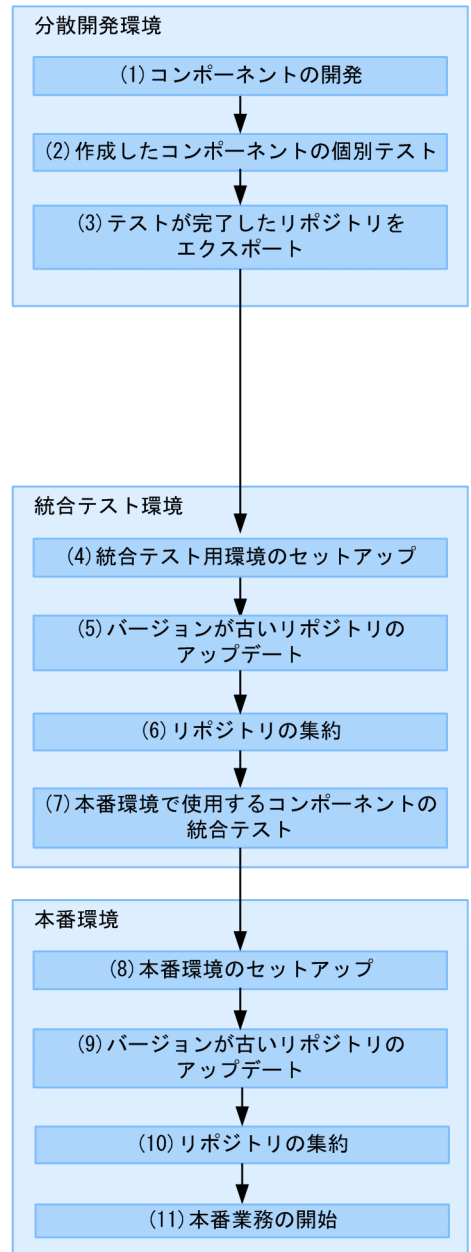
また、開発環境でリポジトリを集約する場合と、運用環境でリポジトリを集約する場合の環境の移行手順を次の図に示します。

図 1-21 環境の移行手順

■開発環境でリポジトリを集約する場合



■運用環境でリポジトリを集約する場合



1.6.2 分散開発時の注意事項

分散開発時の注意事項を次に示します。

- サービス ID, サービス名, コンテキストルートおよび UOC ファイルは, リポジトリの集約時に重複しないよう, 分散開発環境全体で一意となるように命名してください。
- UOC ファイルの共通パッケージは, 同期を取ってください。

2

HCSCサーバのサービス部品呼び出し

この章では、サービス部品を呼び出す基本的な流れや、Cosminexus サービスプラットフォームの実行環境の仕組みについて説明します。また、Web サービス、SessionBean、MDB (WS-R)、およびMDB (DB キュー) を使用した場合の、サービス部品の呼び出しについても説明します。なお、開発環境の定義は、運用環境を通じて実行環境で動作しますが、ここでは運用環境の説明は省略します。

2.1 サービス部品呼び出しの基本的な流れ

2.2 Web サービス (SOAP 通信) を使ったサービス部品呼び出し

2.3 SessionBean を使ったサービス部品呼び出し

2.4 MDB (WS-R) を使ったサービス部品呼び出し

2.5 MDB (DB キュー) を使ったサービス部品呼び出し

2.1 サービス部品呼び出しの基本的な流れ

Cosminexus サービスプラットフォームを使ってシステム構築や運用をする場合、サービスリクエストから HCSC サーバを経由してサービス部品を呼び出す流れを理解することで、より良いシステムが実現でき、SOA を適用することによる効果が得られます。

また、SOA を適用したシステム開発をする場合も、どのように運用されるのか理解した上で、サービス部品呼び出し処理やエラー時の処理を設計することが大切です。

サービス部品を呼び出す流れには、プロトコルで共通な部分と、プロトコルごとに異なる部分があります。ここでは、プロトコルで共通するサービス部品を呼び出す基本的な流れについて説明します。

2.1.1 基本的な構造

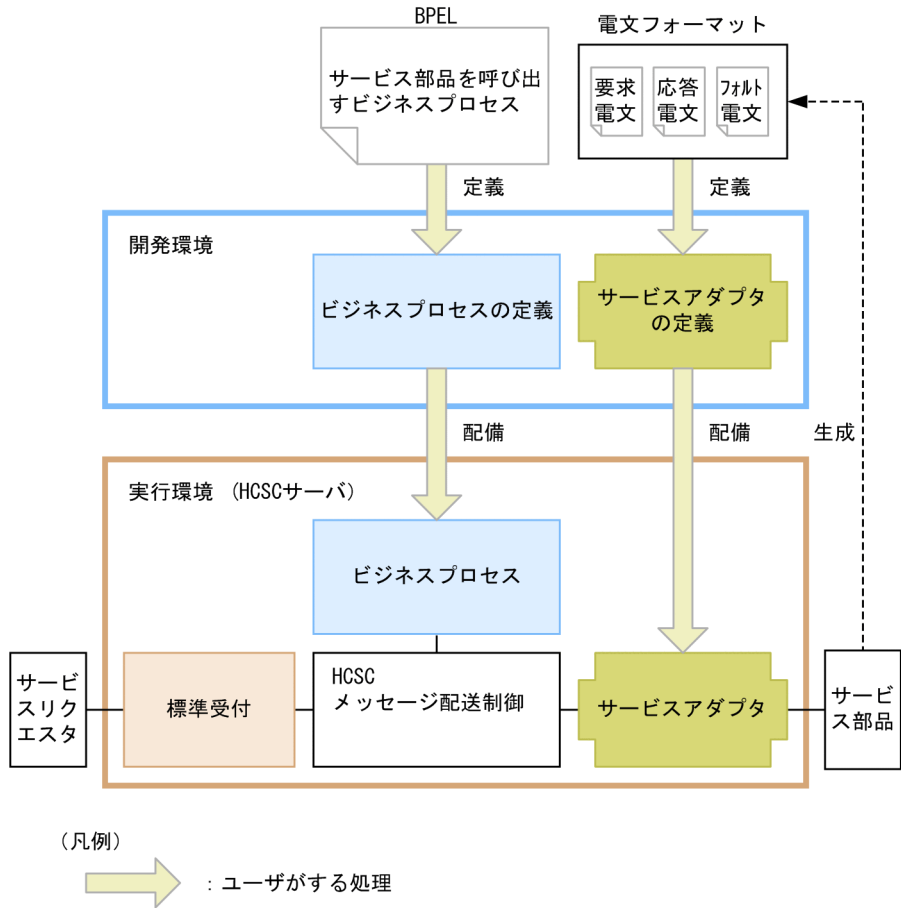
サービス部品を呼び出すためには、基本的にサービスリクエスト、受付、HCSC メッセージ配送制御、サービスアダプタ、およびサービス部品が必要です。受付は、サービス部品を呼び出すための要求を受け付けるところで、標準受付とユーザ定義受付があります。

標準受付とは、HCSC サーバが標準提供している、サービスリクエストからの要求電文を受け付けるための機能（インターフェース）です。ユーザ定義受付とは、開発環境でユーザが定義したものを HCSC サーバに配備してから、サービスリクエストからの実行要求を受け付けるための機能（インターフェース）で、ユーザが定義した任意のインターフェースで受け付けることができます。

(1) 標準受付の場合

標準受付を使用する場合の構造と開発の流れとの関係を次の図に示します。

図 2-1 標準受付を使用する場合の構造と開発の流れとの関係



サービス部品を呼び出すサービスリクエストと HCSC サーバ、およびサービス部品の関係は、次のようになっています。開発の流れに沿って、それぞれの関係を説明します。

- サービス部品側の開発の流れ

最初に呼び出すサービス部品を作成します。次に、呼び出すサービス部品に合わせた形式の電文フォーマットやビジネスプロセス定義 (BPEL) を作成し、開発環境で電文フォーマットからサービスアダプタを定義したり、ビジネスプロセスを定義したりします (ビジネスプロセスにも電文フォーマットを定義します)。呼び出すサービス部品に合わせた形式の電文フォーマットを作成する方法は、使用するプロトコル (SOAP, RMI-IIOP, WS-R, および JMS) によって異なります。定義したサービスアダプタやビジネスプロセスは実行環境 (HCSC サーバ) へ配備します。電文フォーマットについては、マニュアル「Cosminexus サービスプラットフォーム 開発ガイド」の「4. 電文フォーマットの作成」を参照してください。ビジネスプロセス定義 (BPEL) については、マニュアル「Cosminexus サービスプラットフォーム 開発ガイド」の「6. ビジネスプロセスの定義」を参照してください。

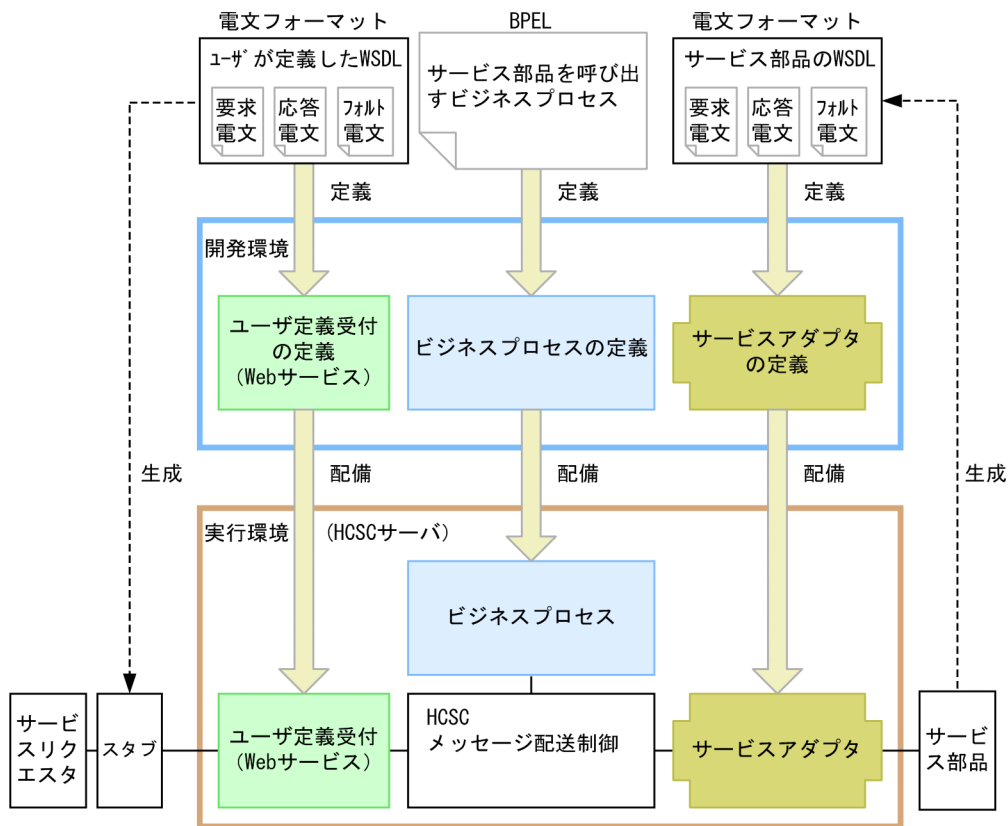
2. HCSC サーバのサービス部品呼び出し

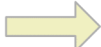
- サービスリクエスタ側の開発の流れ
サービス部品の作成後、定義したサービスアダプタやビジネスプロセスを呼び出すサービスリクエスタを作成します。サービスリクエスタを作成する方法については、使用する標準受付の protocols によって異なります。サービスリクエスタの作成については、マニュアル「Cosminexus サービスプラットフォーム 開発ガイド」の「10. サービスリクエスタの作成」を参照してください。

(2) ユーザ定義受付の場合

ユーザ定義受付を使用する場合の構造と開発の流れとの関係を次の図に示します。

図 2-2 ユーザ定義受付を使用する場合の構造と開発の流れとの関係



(凡例)
 : ユーザがする処理

SOAP 通信を使用する場合は、ユーザ定義受付を使用できます。サービス部品を呼び出すサービスリクエスタと HCSC サーバ、およびサービス部品の関係は、次のようになっています。開発の流れに沿って、それぞれの関係を説明します。

- サービス部品側の開発の流れ

最初に呼び出すサービス部品を作成します。次に、呼び出すサービス部品に合わせた形式の電文フォーマットやビジネスプロセス定義（BPEL）を作成し、開発環境で電文フォーマットからサービスアダプタを定義したり、ビジネスプロセスを定義したりします（ビジネスプロセスにも電文フォーマットを定義します）。定義したサービスアダプタ、ビジネスプロセス、およびユーザ定義受付は実行環境（HCSC サーバ）へ配備します。電文フォーマットについては、マニュアル「Cosminexus サービスプラットフォーム 開発ガイド」の「4. 電文フォーマットの作成」を参照してください。ビジネスプロセス定義（BPEL）については、マニュアル「Cosminexus サービスプラットフォーム 開発ガイド」の「6. ビジネスプロセスの定義」を参照してください。

- 受付側の開発の流れ

サービス部品の作成後、呼び出すビジネスプロセスの電文フォーマットに合わせた WSDL を作成し、開発環境でユーザ定義受付を定義します。ユーザ定義受付については、マニュアル「Cosminexus サービスプラットフォーム 開発ガイド」の「8. ユーザ定義受付の定義」を参照してください。

- サービスリクエスト側の開発の流れ

ユーザ定義受付の作成後、定義したユーザ定義受付を呼び出すサービスリクエストを作成します。ユーザ定義受付を定義するときに使用した WSDL からスタブを生成し、スタブを呼び出すサービスリクエストを作成します。サービスリクエストの作成については、マニュアル「Cosminexus サービスプラットフォーム 開発ガイド」の「10. サービスリクエストの作成」を参照してください。

2.1.2 サービス部品を呼び出す流れ

サービスリクエストからサービス部品を呼び出す流れは、受付の種類と呼び出す HCSC コンポーネントの組み合わせによって動作が異なります。受付の種類と呼び出す HCSC コンポーネントの組み合わせを次に示します。

- 標準受付からサービスアダプタを直接呼び出してサービス部品を実行する場合
- 標準受付からビジネスプロセスを呼び出してサービス部品を実行する場合
- ユーザ定義受付からビジネスプロセスを呼び出してサービス部品を実行する場合

これらの場合について、次に説明します。エラー時の処理の流れについては、マニュアル「Cosminexus サービスプラットフォーム システム構築・運用ガイド」の「7. 障害対策」を参照してください。

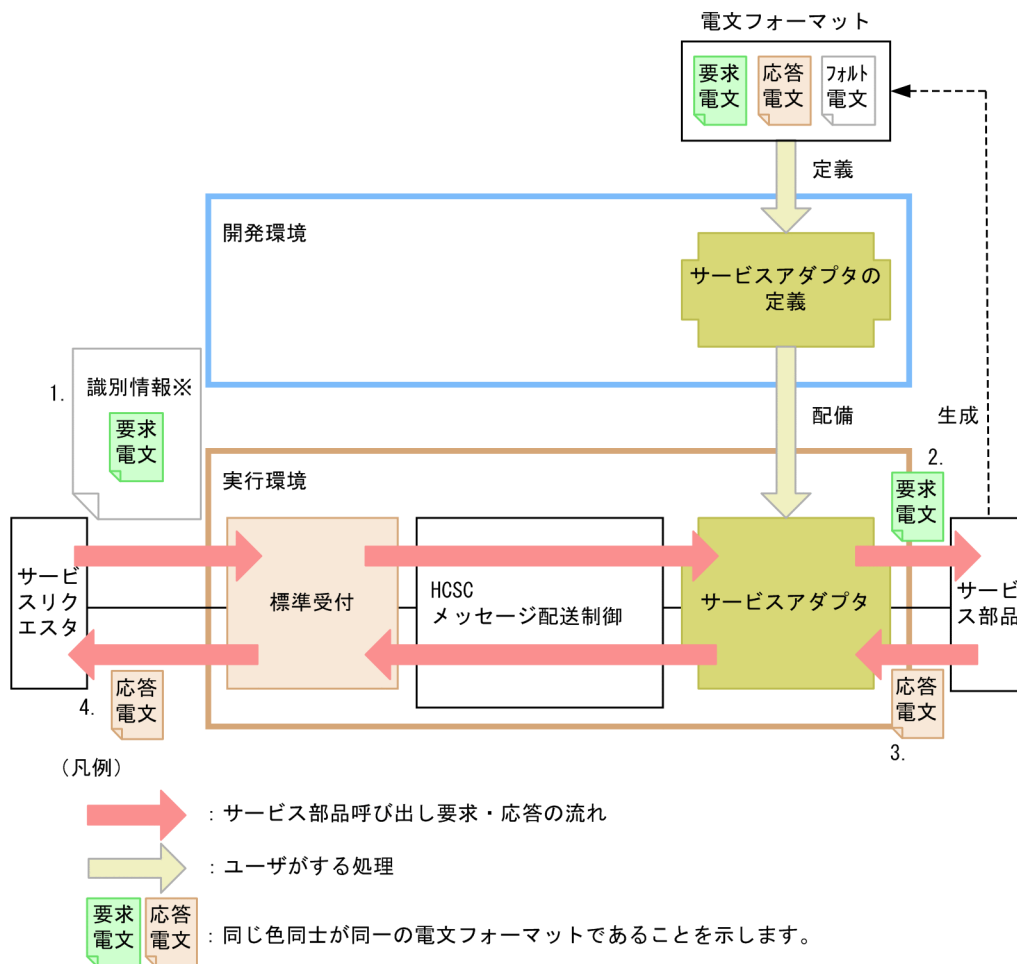
(1) 標準受付からサービスアダプタを直接呼び出してサービス部品を実行する場合

標準受付からサービスアダプタを直接呼び出してサービス部品を実行する場合、サービスアダプタで定義したサービス名をサービスリクエストで指定して実行します。実行すると、サービスアダプタで定義したサービス部品を呼び出します。

サービスアダプタを直接呼び出す場合の電文の流れを次の図に示します。

2. HCSC サーバのサービス部品呼び出し

図 2-3 サービスアダプタを直接呼び出す場合の電文の流れ



注※ 識別情報とは、サービス名、クライアント相関IDなどです。

1. サービスリクエストから標準受付に送信する要求電文は、開発環境でサービスアダプタを定義したときに設定した（サービス部品の形式に合わせた）電文フォーマットを使用します。そのため、サービスリクエスト内で、サービスアダプタで定義した電文フォーマットに合わせた要求電文を作成し、作成した電文を標準受付のパラメタ（ユーザ電文のパラメタ）に設定して実行します。
2. サービスアダプタからサービス部品を呼び出します。呼び出すときは、サービスアダプタで定義した要求電文を使用します。このように、サービスリクエストから送信する要求電文とサービス部品が受け取る要求電文は同一になります。なお、サービス部品の電文フォーマット以外の電文フォーマットでも、サービスアダプタでデータ変換定義をすれば、サービスリクエストから要求できます。
3. サービス部品からサービスアダプタに応答電文を送信します。応答電文は、サービスアダプタで定義した応答電文と同じ電文フォーマットを使用します。
4. サービスリクエストへ応答します。応答には、サービスアダプタで定義した応答電文

の電文フォーマットを使用します。このように、サービス部品から返す応答電文とサービスリクエストで受け取る応答電文は同一になります。

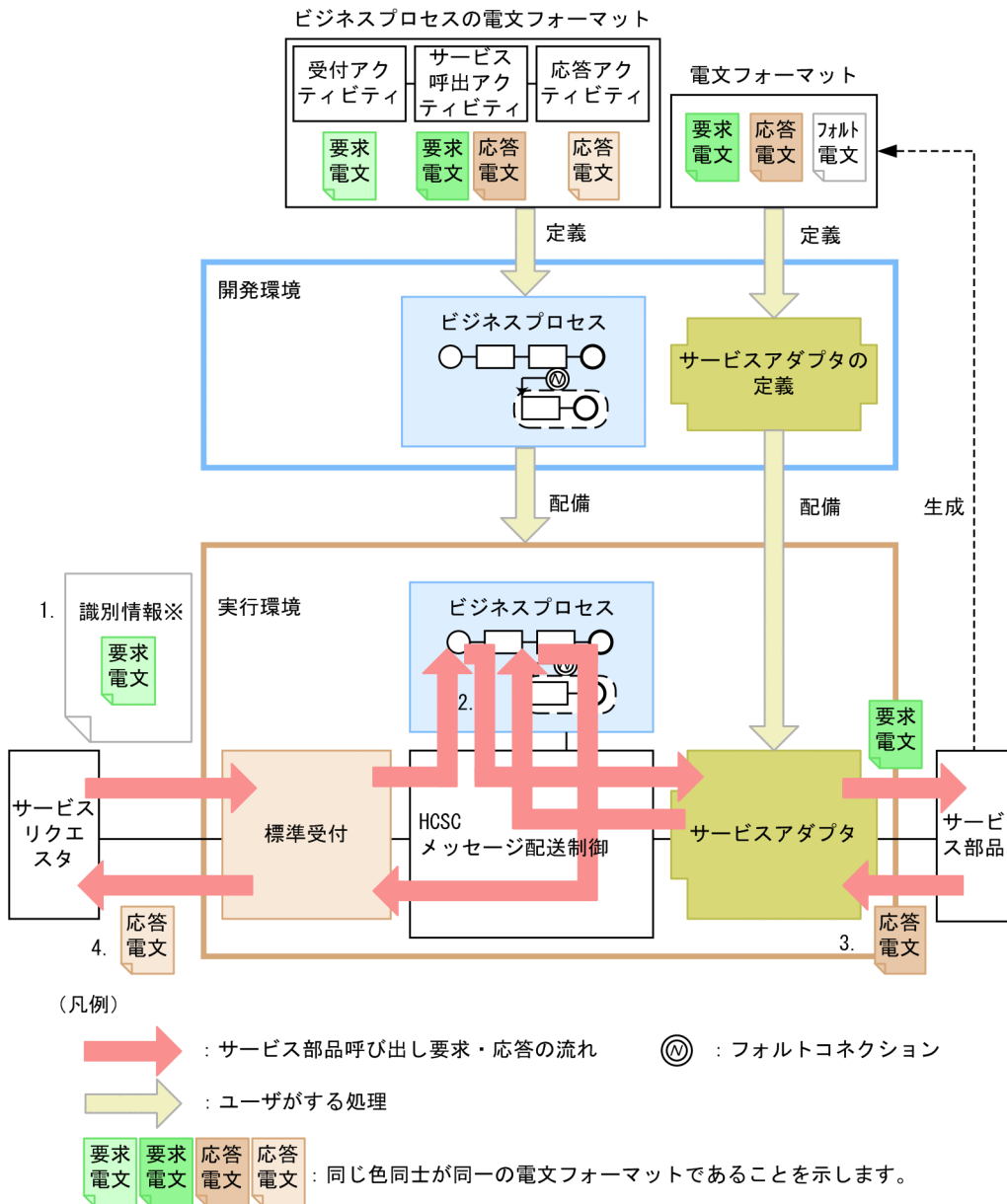
データ変換定義については、マニュアル「Cosminexus サービスプラットフォーム 開発ガイド」の「7.3 データ変換定義」を参照してください。

(2) 標準受付からビジネスプロセスを呼び出してサービス部品を実行する場合

標準受付からビジネスプロセスを呼び出してサービス部品を実行する場合、ビジネスプロセスで定義したサービス名（ビジネスプロセス名）をサービスリクエストで指定して実行します。そして、ビジネスプロセスに定義したプロセスに従って、処理を実行します。プロセスはアクティビティから構成されています。その中の、サービス部品呼出アクティビティでは、定義されたサービスアダプタを呼び出して、サービスアダプタに定義されたサービス部品を実行します。ビジネスプロセスを呼び出す場合の電文の流れを次の図に示します。

2. HCSC サーバのサービス部品呼び出し

図 2-4 ビジネスプロセスを呼び出す場合の電文の流れ



注※ 識別情報とは、サービス名、クライアント相関IDなどです。

1. サービスリクエスト内で、ビジネスプロセスの受付アクティビティで定義した要求電文の電文フォーマットに合わせた要求電文を作成し、作成した電文を標準受付のパラメタ（ユーザ電文のパラメタ）に設定して実行します。サービスリクエストで指定する要求電文は、ビジネスプロセスの受付アクティビティに定義した電文フォーマットを使用します。

2. ビジネスプロセスからサービス部品を呼び出します。呼び出すときは、サービス呼出アクティビティで定義した要求電文を使用します。この要求電文の電文フォーマットは、呼び出すサービス部品のサービスアダプタで定義した要求電文の形式と同じである必要があります。なお、サービス部品の電文フォーマット以外の電文フォーマットでも、サービスアダプタでデータ変換定義をすれば、サービス呼出アクティビティから要求できます。
3. サービス部品からサービスアダプタに応答電文を送信します。応答電文は、ビジネスプロセスのサービス呼出アクティビティの応答電文と同じ電文フォーマットを定義します。
4. サービスリクエストへ応答します。応答には、ビジネスプロセスの応答アクティビティで定義した応答電文の電文フォーマットを使用します。

データ変換定義については、マニュアル「Cosminexus サービスプラットフォーム 開発ガイド」の「7.3 データ変換定義」を参照してください。

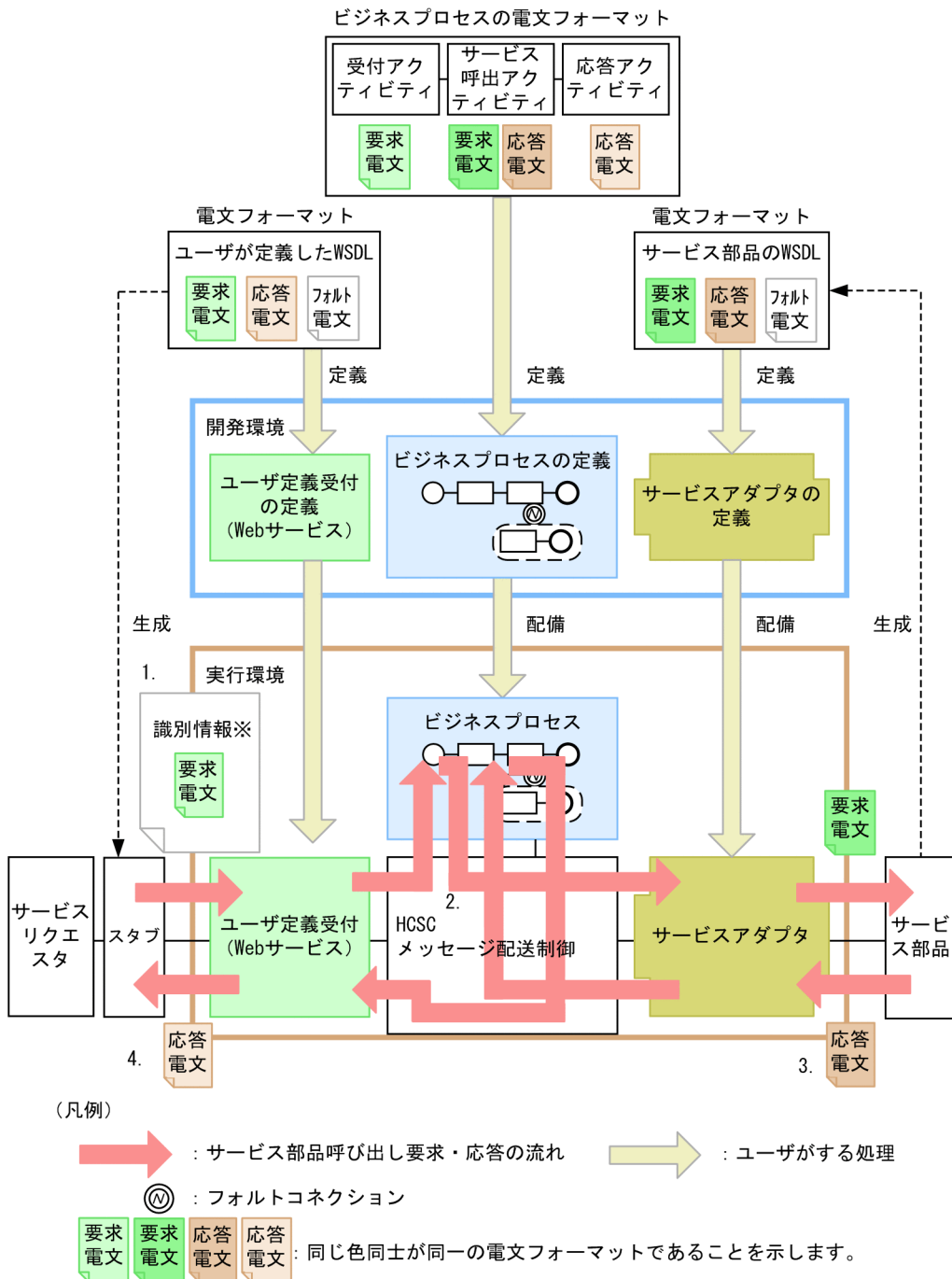
(3) ユーザ定義受付からビジネスプロセスを呼び出してサービス部品を実行する場合

ユーザ定義受付は、ビジネスプロセスの呼び出しをする場合にだけ使用できます。サービスリクエストからは、HCSC サーバで定義したサービス名（ビジネスプロセス名）などは指定しないで、ユーザ定義受付に定義した WSDL を使用して要求します。ユーザ定義受付に定義した WSDL は、ビジネスプロセスの受付アクティビティに定義した電文フォーマットに形式を合わせます。

開発環境での定義に従ってユーザ定義受付からビジネスプロセスを呼び出し、ビジネスプロセスに定義したプロセスに従って、処理を実行します。サービス呼出アクティビティでは、定義されたサービスアダプタを呼び出し、サービスアダプタに定義されたサービス部品を実行します。ユーザ定義受付を使う場合の電文の流れを次の図に示します。

2. HCSC サーバのサービス部品呼び出し

図 2-5 ユーザ定義受付を使う場合の電文の流れ



注※ 識別情報とは、サービス名、クライアント関連IDなどです。

1. サービスリクエスタからの要求電文は、ビジネスプロセスの受付アクティビティに定

義した電文フォーマットと同一のものになります。そのため、ビジネスプロセスの受付アクティビティに定義した電文フォーマットに合わせた形式の WSDL を開発環境のユーザ定義受付定義で設定する必要があります。サービスリクエストからユーザ定義受付を呼び出すときは、開発環境でユーザ定義受付に定義した WSDL を使用します。WSDL から WSDL2Java コマンドを使用してスタブを生成し、スタブを呼び出すようにサービスリクエストを実装します。

2. ビジネスプロセスからサービス部品を呼び出すときは、サービス呼出アクティビティで定義した要求電文を使用します。この要求電文の電文フォーマットは、呼び出すサービス部品のサービスアダプタで定義した要求電文の形式と同じである必要があります。
3. サービス部品からの応答電文もビジネスプロセスのサービス呼出アクティビティの応答電文で同じ電文フォーマットを定義します。なお、サービス部品の電文フォーマット以外の電文フォーマットでも、サービスアダプタでデータ変換定義をすれば、サービス呼出アクティビティから要求できます。
4. サービスリクエストへの応答は、ビジネスプロセスの応答アクティビティで定義した応答電文の電文フォーマットを使用します。この電文フォーマットは開発環境のユーザ定義受付定義で設定する WSDL の応答電文と同じ形式である必要があります。

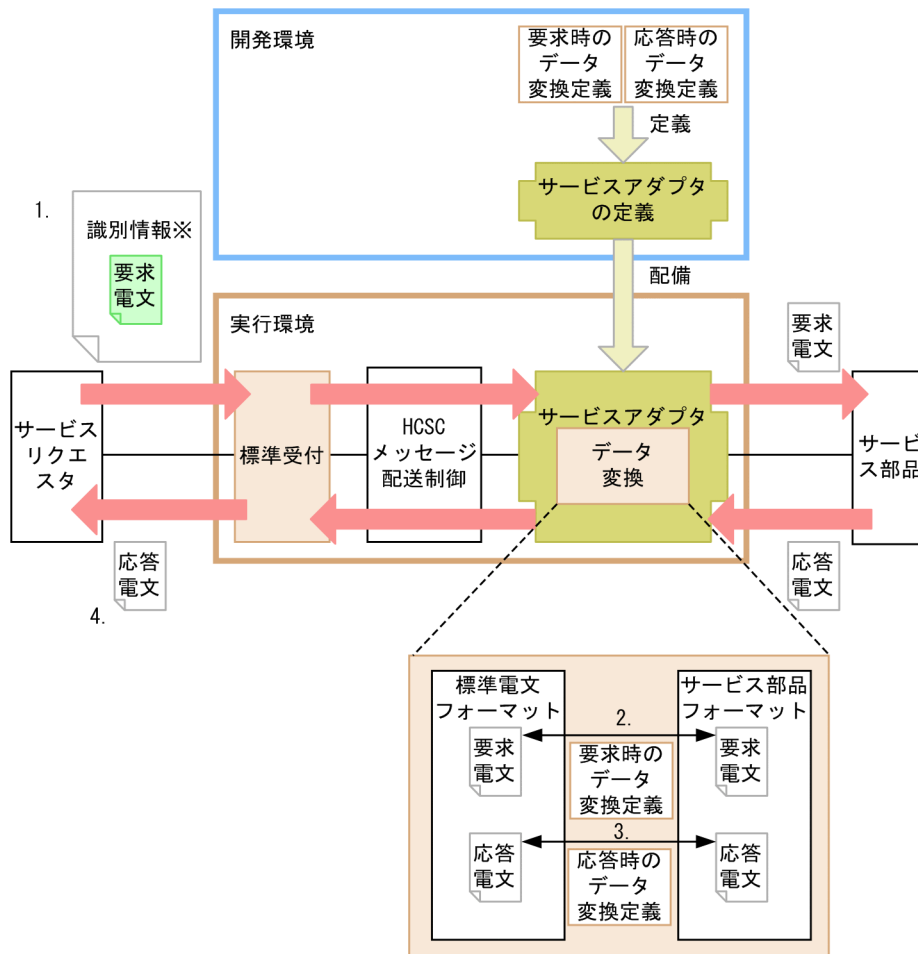
データ変換定義については、マニュアル「Cosminexus サービスプラットフォーム 開発ガイド」の「7.3 データ変換定義」を参照してください。

2.1.3 サービス部品側と異なる電文フォーマットでサービス部品を呼び出す場合の流れ




サービス部品側の電文フォーマットと異なる電文でサービス部品を呼び出す場合、またはサービス部品側の電文フォーマットと異なる電文を応答として受信する場合は、開発環境で標準電文フォーマットを定義します。また、標準電文フォーマットとサービス部品電文フォーマットの変換ルールを定義したデータ変換定義は、開発環境で定義します。標準電文フォーマットの電文をサービス部品電文フォーマットの電文に変換することでサービス部品を呼び出せます。標準電文フォーマットとデータ変換定義の関係を次の図に示します。

2. HCSC サーバのサービス部品呼び出し

図 2-6 標準電文フォーマットとデータ変換定義の関係



(凡例)

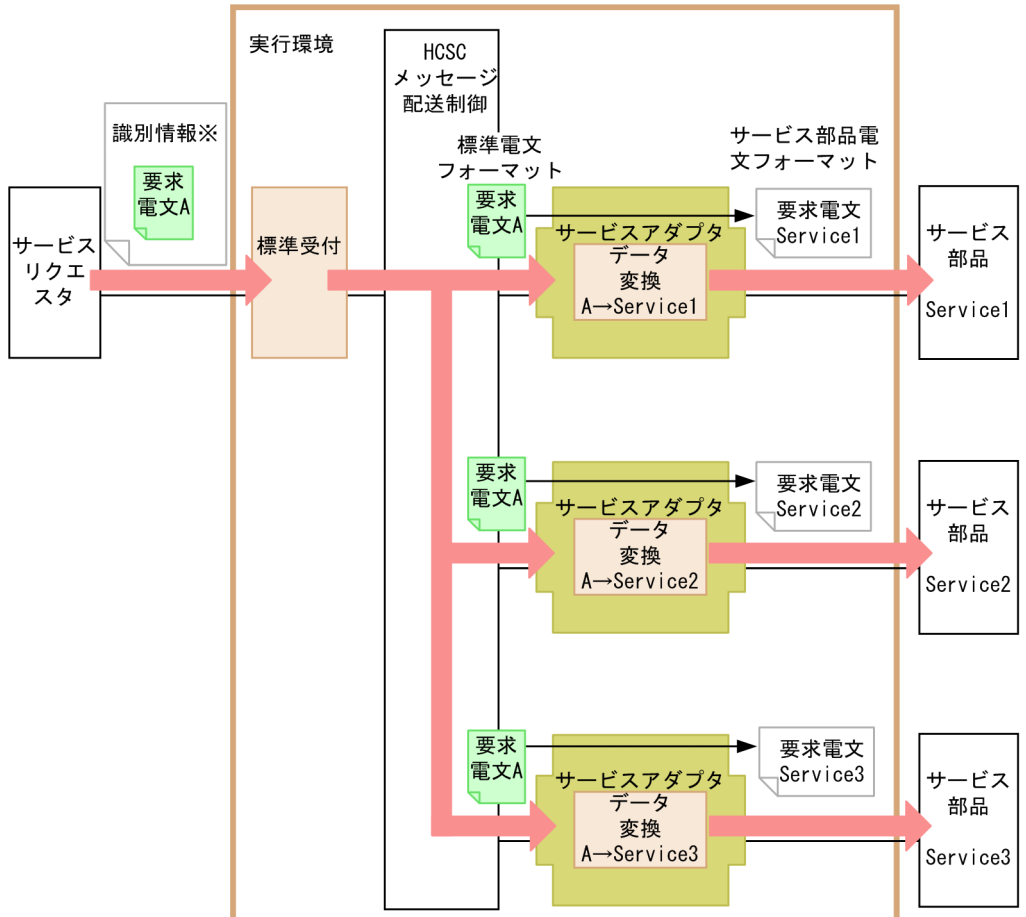
-  : サービス部品呼び出し要求・応答の流れ
-  : ユーザがする処理
-  : データ変換処理

注※ 識別情報とは、サービス名、クライアント相関IDなどです。


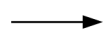

1. サービスリクエスタからは、標準電文フォーマットの形式に合わせた電文フォーマットを要求電文に指定して、サービス部品の呼び出しを要求します。
2. サービスアダプタでは標準電文フォーマットからサービス部品電文フォーマットにデータを変換し、変換後の電文フォーマットでサービス部品を呼び出します。
3. サービス部品からの応答についても、サービスアダプタで、サービス部品電文フォーマットから標準電文フォーマットにデータを変換します。
4. サービスリクエスタに標準電文フォーマットを応答します。

複数のサービスアダプタがある場合も、図 2-7 に示すように Service1 用のサービスアダプタ、Service2 用のサービスアダプタ、Service3 用のサービスアダプタで、それぞれに同一の標準電文フォーマット A と、それぞれのデータ変換定義を定義すると、サービスリクエスト側から要求するときに、要求時のサービス名 (Service1 など) を変更すれば、同じ要求電文で要求できます。

図 2-7 複数のサービスアダプタで同一の標準電文フォーマットを定義した場合の流れ



(凡例)

-  : サービス部品呼び出し要求・応答の流れ
-  : データ変換処理
-  : 同じ色同士が同一の電文フォーマットであることを示します。

注※ 識別情報とは、サービス名、クライアント関連IDなどです。

標準電文フォーマットおよびサービス部品電文フォーマットについては、マニュアル

2. HCSC サーバのサービス部品呼び出し

「Cosminexus サービスプラットフォーム 開発ガイド」の「4. 電文フォーマットの作成」を参照してください。データ変換定義については、マニュアル「Cosminexus サービスプラットフォーム 開発ガイド」の「7.3 データ変換定義」を参照してください。

2.1.4 サービス部品呼び出し処理で与えられる識別情報

HCSC サーバ内のサービス部品呼び出し要求とその流れを区別するために、各要求電文にメッセージ共通 ID とサービスリクエスト ID が割り当てられます。これらの ID は電文の実行履歴、メッセージログ、リクエストトレース、および性能解析トレースで出力されます。

メッセージ共通 ID とサービスリクエスト ID をたどることで、サービス部品呼び出しの処理の流れを追跡できます。電文の追跡については、マニュアル「Cosminexus サービスプラットフォーム システム構築・運用ガイド」の「7.7 サービス部品呼び出し要求時の障害対策」を参照してください。

なお、HCSC サーバに割り当てられる識別情報のほかに、ユーザが任意の値を設定できるクライアント関連 ID があります。

(1) 識別情報の種類

- メッセージ共通 ID

サービスリクエストからサービス部品呼び出し要求を受け取ったタイミングで自動的に割り当てられます。応答まで同じメッセージ共通 ID を使用します。

サービスリクエストからビジネスプロセスを呼び出す場合でも、ビジネスプロセス内の個々のサービス部品呼び出し処理を同一の処理として識別できます。メッセージ共通 ID の形式を次に示します。

```
CSC_[サーバ名]_[受付時刻(ミリ秒)]_[通番]
```

注 末尾に与えられる通番は、「1」から順番に割り当てられる番号です。

(例) CSC_CSCServerName_2008-04-20_11:32:18.360_1

- サービスリクエスト ID

HCSC サーバ内の個別のサービス部品呼び出し処理ごとに割り当てられる識別情報です。

サービスリクエストからのサービス部品呼び出し要求を受け取ったタイミングだけでなく、ビジネスプロセスからのサービス部品呼び出し処理を実施するタイミングでも割り当てられます。サービスリクエスト ID の形式を次に示します。

```
MSG_[サーバ名]_[受付種別]_[受付時刻(ミリ秒)]_[通番]
```


注 末尾に与えられる通番は、「1」から順番に割り当てられる番号です。

(例) MSG_CSCServerName_MDBWSR_2008-04-20_11:32:18.360_1

- ビジネスプロセスインスタンス ID

ビジネスプロセスの個々のプロセスインスタンスを特定するための識別情報（プロセスインスタンスの識別子）です。サービスリクエストからビジネスプロセスの再実行を要求する場合や、プロセスインスタンスの実行履歴を削除するときに使用します。

(2) 識別情報が割り当てられるタイミング

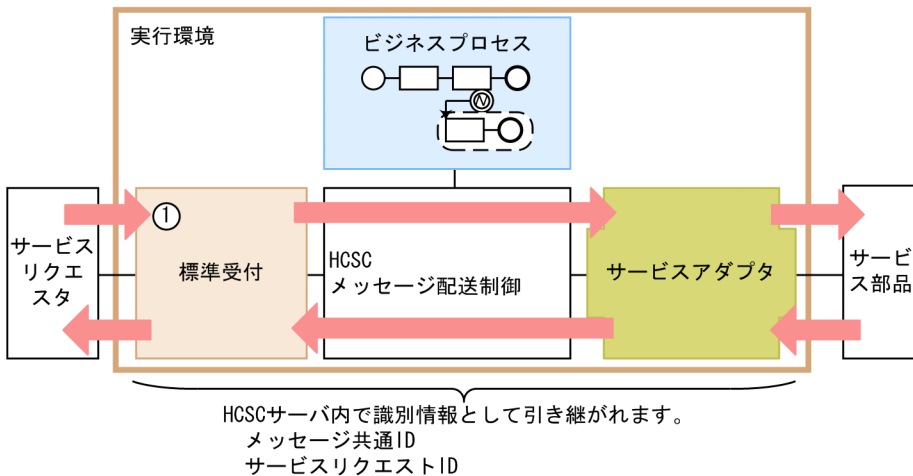
標準受付から直接サービスアダプタを呼び出す場合は、メッセージ共通 ID およびサービスリクエスト ID は、HCSC サーバ内で識別情報として引き継がれます。標準受付やユーザ定義受付からビジネスプロセスを経由してサービスアダプタを呼び出す場合は、メッセージ共通 ID は、HCSC サーバ内で識別情報として引き継がれます。しかし、サービスリクエスト ID は、サービス部品呼び出し処理ごとの識別情報として引き継がれます。また、ビジネスプロセスインスタンス ID は、ビジネスプロセスが最初に要求を受け付けるときに割り当てます。

メッセージ共通 ID とサービスリクエスト ID が割り当てられるタイミングを次の図に示します。

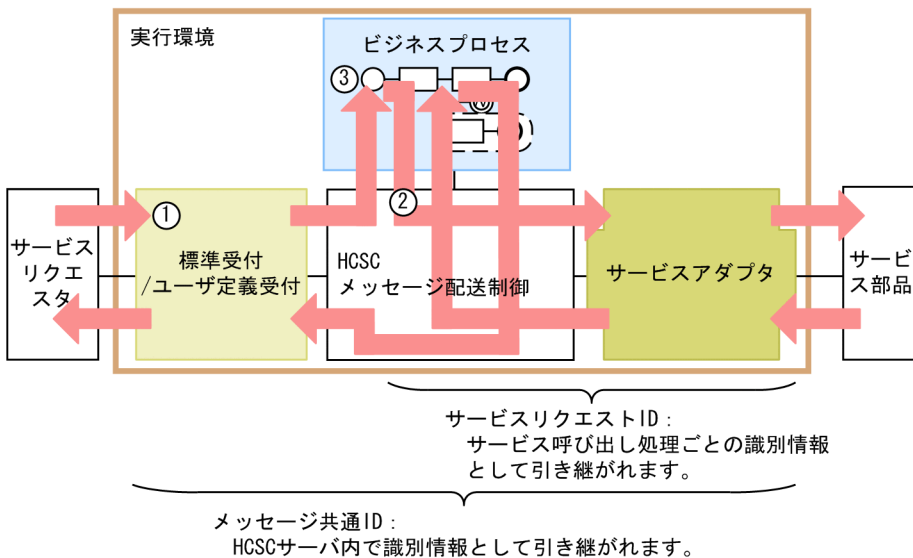
2. HCSC サーバのサービス部品呼び出し

図 2-8 識別情報割り当てのタイミング

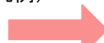
●標準受付から直接サービスアダプタを呼び出す場合



●ビジネスプロセスを呼び出す場合



(凡例)

-  : サービス部品呼び出し要求・応答の流れ
- ① : メッセージ共通ID, およびサービスリクエストIDの割り当て
- ② : サービスリクエストIDの割り当て
- ③ : ビジネスプロセスインスタンスIDの割り当て
- ④ : フォルトコネクション

2.2 Web サービス (SOAP 通信) を使ったサービス部品呼び出し

2.2.1 Web サービスの通信基盤との関係

実行環境の HCSC サーバは、Web サービスの通信基盤および Web サーバ (HTTP サーバ) 上で動作します。そのため、Web サービス (SOAP 通信) を使用する場合は、Web サービスを実行するための通信基盤や Web サーバの設定が必要です。

サービスリクエストと HCSC サーバのリクエスト受付との関係は、クライアントとサーバの関係であり、HCSC サーバのサービスアダプタとサービス部品の関係もクライアントとサーバの関係になります。

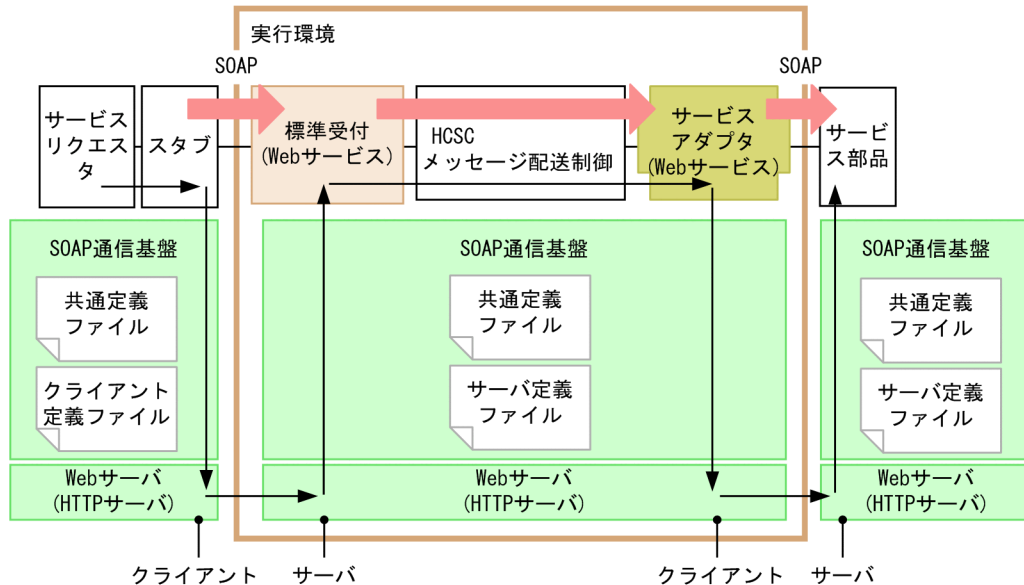
サービスリクエスト側およびサービス部品側の通信基盤は、SOAP 通信基盤を使用する場合と JAX-WS エンジンを使用する場合があります。SOAP1.1 に対応したサービスリクエストおよびサービス部品を使用する場合は、SOAP 通信基盤または JAX-WS エンジンを使用します。また、SOAP1.2 に対応したサービスリクエストおよびサービス部品を使用する場合は、JAX-WS エンジンを使用します。

サービスリクエストおよびサービス部品と、SOAP 通信基盤の関係を次の図に示します。なお、サービスリクエスト側とサービス部品側で、片方が SOAP 通信基盤、もう片方が JAX-WS エンジンを使用する構成も可能です。

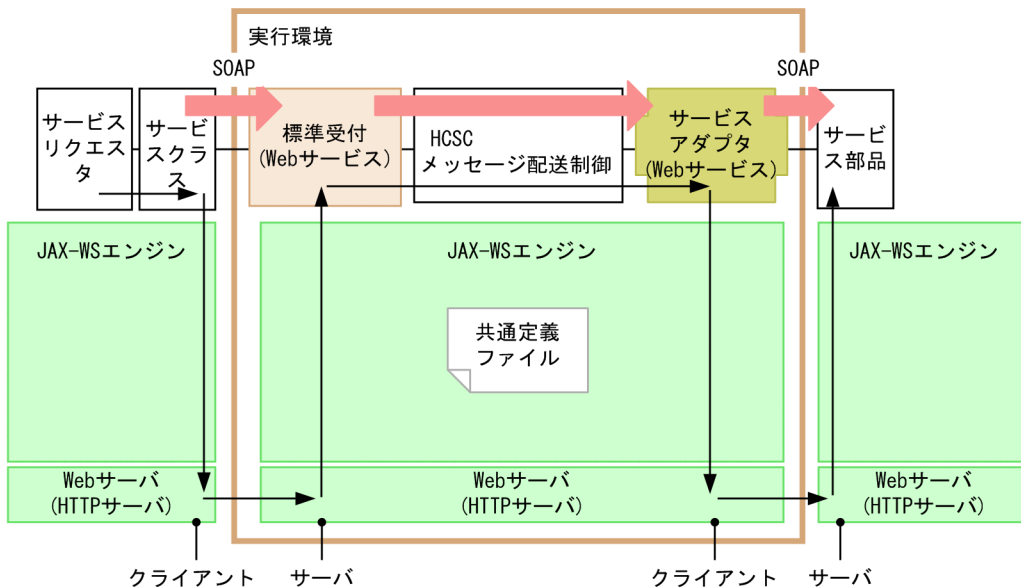
2. HCSC サーバのサービス部品呼び出し

図 2-9 SOAP 通信基盤との関係

●SOAP通信基盤を使用する場合



●JAX-WSエンジンを使用する場合



(凡例)

- : サービス部品呼び出し要求の流れ
- : 電文の流れ

2.2.2 サービス部品呼び出し時のユーザ電文と WSDL の関係

サービス部品呼び出し時のユーザ電文と WSDL の関係は、標準受付の場合とユーザ定義受付の場合とで異なります。

(1) 標準受付の場合

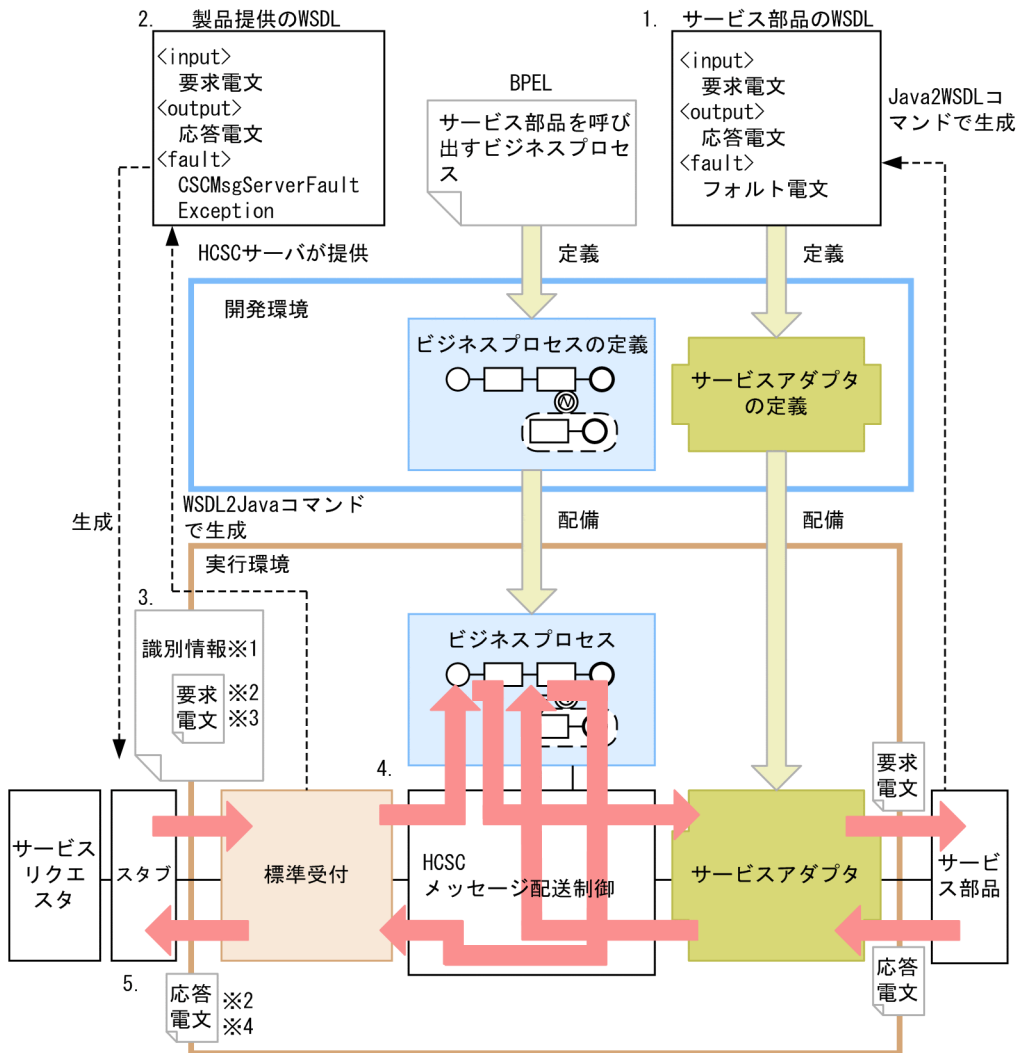
SOAP 通信時のユーザ電文と WSDL の関係（標準受付）を次に説明します。

(a) SOAP 通信基盤を使用するとき

通信基盤に SOAP 通信基盤を使用するときの、SOAP 通信時のユーザ電文と WSDL の関係（標準受付）を次の図に示します。なお、サービスリクエスト側またはサービス部品側で JAX-WS エンジンを使用する場合、JAX-WS エンジンを使用する側については、「(b) JAX-WS エンジンを使用するとき」を参照してください。

2. HCSC サーバのサービス部品呼び出し

図 2-10 SOAP 通信基盤を使用するときの SOAP 通信時のユーザ電文と WSDL の関係 (標準受付)



(凡例)

➡ : サービス部品呼び出し要求・応答の流れ

➡ : ユーザがする処理

Ⓜ : フォルトコネクション

注※1 識別情報とは、サービス名、クライアント相関IDなどです。

注※2 サービス部品のWSDLに定義されている要求電文または応答電文です。

注※3 ビジネスプロセスを呼び出す場合は、ビジネスプロセスの受付アクティビティに定義した要求電文です。

注※4 ビジネスプロセスを呼び出す場合は、ビジネスプロセスの応答アクティビティに定義した応答電文です。

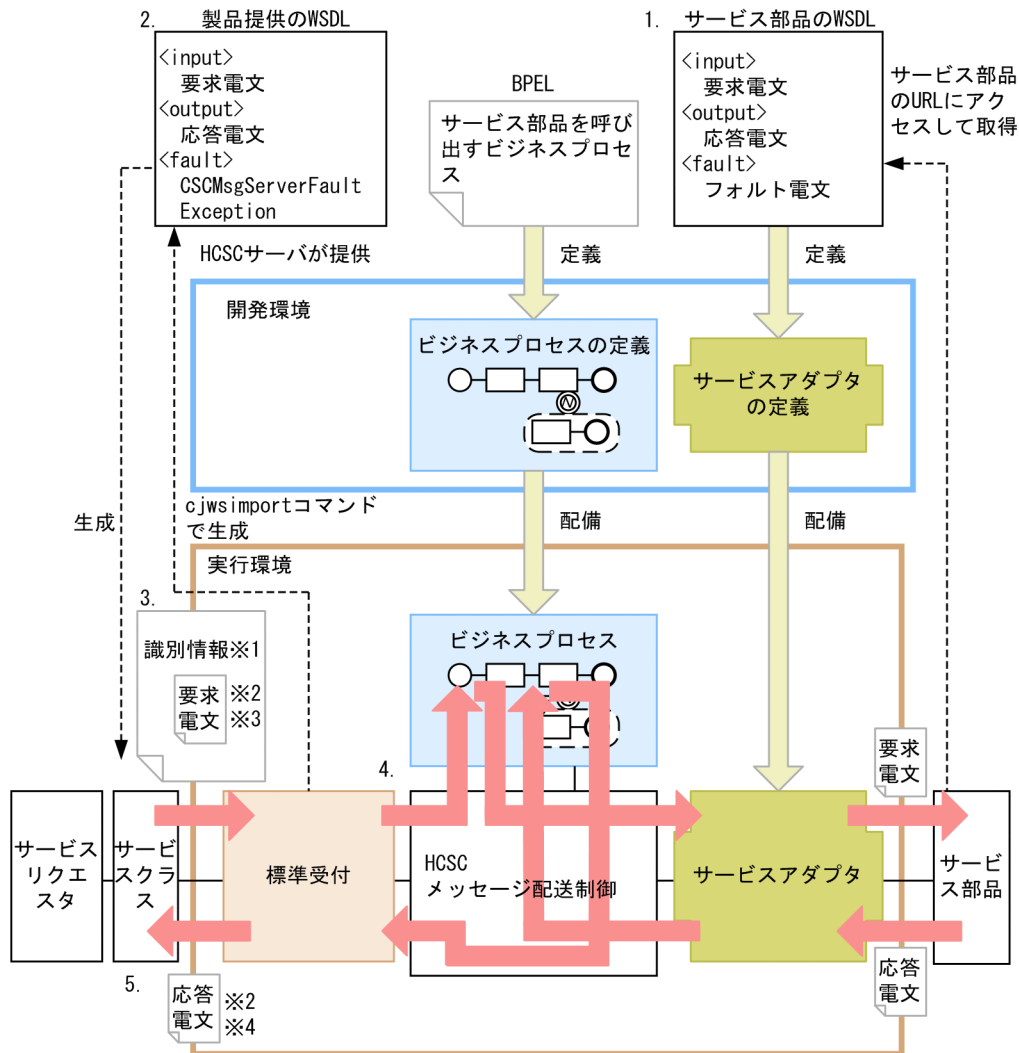
1. サービスアダプタを定義するときは、サービス部品側の電文フォーマットとしてサービス部品の WSDL を定義します（サービス部品の WSDL には、要求電文、応答電文およびフォルト電文の形式が定義されています）。WSDL を作成するには、SOAP 通信基盤が提供する Java2WSDL コマンドを使用します。
2. サービスリクエスタから標準受付を呼び出してサービス部品を呼び出すときは、HCSC サーバが提供する WSDL を使用します。HCSC サーバが提供する WSDL を基に WSDL2Java コマンドを使用してスタブを作成できます。
3. サービスリクエスタから標準受付を呼び出すときは、サービス部品側の電文フォーマットに合わせた要求電文を事前に作成します。そして、HCSC サーバが提供する WSDL に定義されている標準受付のパラメタ（ユーザ電文のパラメタ）に設定して実行します。
4. サービスリクエスタからビジネスプロセスを呼び出すときは、要求電文はビジネスプロセスの受付アクティビティに定義した電文フォーマットになります。サービスリクエスタ内でビジネスプロセスの受付アクティビティで定義されている要求電文の電文フォーマットに合わせた電文を作成し、作成した電文を標準受付のパラメタ（ユーザ電文のパラメタ）に設定して実行します。
5. サービスリクエスタへの応答は、サービスアダプタで定義した応答電文の電文フォーマットを使用します。

(b) JAX-WS エンジンを使用するとき

通信基盤に JAX-WS エンジンを使用するときの、SOAP 通信時のユーザ電文と WSDL の関係（標準受付）を次の図に示します。なお、サービスリクエスタ側またはサービス部品側で SOAP 通信基盤を使用する場合、SOAP 通信基盤を使用する側については、「(a) SOAP 通信基盤を使用するとき」を参照してください。

2. HCSC サーバのサービス部品呼び出し

図 2-11 JAX-WS エンジンを使用するときの SOAP 通信時のユーザ電文と WSDL の関係 (標準受付)



(凡例)

➡ : サービス部品呼び出し要求・応答の流れ

➡ : ユーザがする処理

Ⓝ : フォルトコネクション

注※1 識別情報とは、サービス名、クライアント相関IDなどです。

注※2 サービス部品のWSDLに定義されている要求電文または応答電文です。

注※3 ビジネスプロセスを呼び出す場合は、ビジネスプロセスの受付アクティビティに定義した要求電文です。

注※4 ビジネスプロセスを呼び出す場合は、ビジネスプロセスの応答アクティビティに定義した応答電文です。

1. サービスアダプタを定義するときは、サービス部品側の電文フォーマットとしてサービス部品の WSDL を定義します（サービス部品の WSDL には、要求電文、応答電文およびフォルト電文の形式が定義されています）。サービス部品の WSDL は、サービス部品側が用意している URL にアクセスして WSDL を取得します。
2. サービスリクエスタから標準受付を呼び出してサービス部品を呼び出すときは、HCSC サーバが提供する WSDL を使用します。HCSC サーバが提供する WSDL を基に `ejwsimport` コマンドを使用してサービスクラスを作成できます。
3. サービスリクエスタから標準受付を呼び出すときは、サービス部品側の電文フォーマットに合わせた要求電文を事前に作成します。そして、HCSC サーバが提供する WSDL に定義されている標準受付のパラメタ（ユーザ電文のパラメタ）に設定して実行します。
4. サービスリクエスタからビジネスプロセスを呼び出すときは、要求電文はビジネスプロセスの受付アクティビティに定義した電文フォーマットになります。サービスリクエスタ内でビジネスプロセスの受付アクティビティで定義されている要求電文の電文フォーマットに合わせた電文を作成し、作成した電文を標準受付のパラメタ（ユーザ電文のパラメタ）に設定して実行します。
5. サービスリクエスタへの応答は、サービスアダプタで定義した応答電文の電文フォーマットを使用します。

（2）ユーザ定義受付の場合

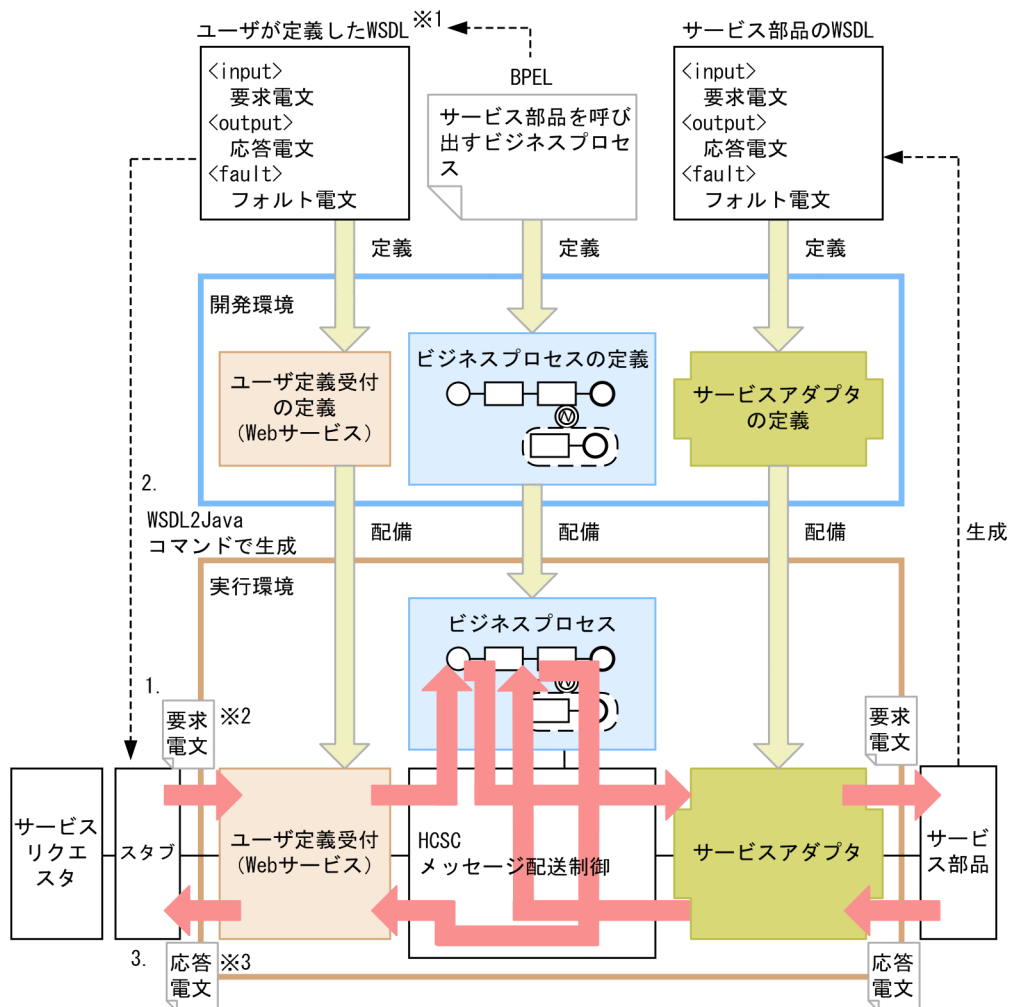
SOAP 通信時のユーザ電文と WSDL の関係（ユーザ定義受付）を次に説明します。

（a）SOAP 通信基盤を使用するとき




通信基盤に SOAP 通信基盤を使用するときの、SOAP 通信時のユーザ電文と WSDL の関係（ユーザ定義受付）を次の図に示します。

2. HCSC サーバのサービス部品呼び出し

図 2-12 SOAP 通信基盤を使用するときの SOAP 通信時のユーザ電文と WSDL の関係
(ユーザ定義受付)



(凡例)

-  : サービス部品呼び出し要求・応答の流れ
-  : ユーザがする処理
-  : フォルトコネクション

注※1 ビジネスプロセスの受付アクティビティや応答アクティビティに定義した形式に合わせた WSDL です。

注※2 ユーザが定義した WSDL に定義されている要求電文です。

注※3 ユーザが定義した WSDL に定義されている応答電文です。

1. ユーザ定義受付の場合、サービスリクエスタで指定する要求電文は、ビジネスプロセスの受付アクティビティに定義した任意の電文フォーマットの形式で要求します (受付アクティビティに定義した電文フォーマットに合わせた WSDL を作成し、ユーザ

定義受付に定義します)。

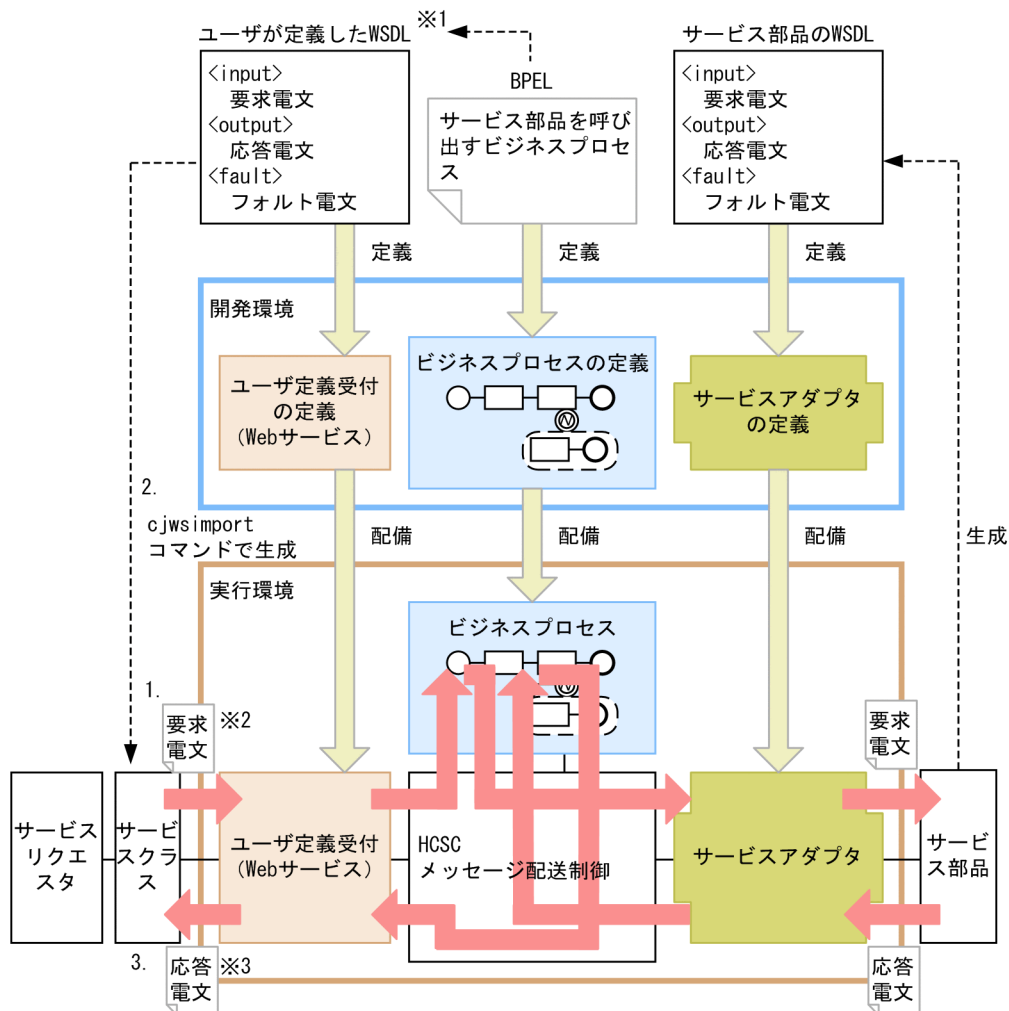
2. スタブは、ユーザが定義した WSDL を基に、SOAP アプリケーション開発支援機能が提供する WSDL2Java コマンドを使用して作成できます。標準受付の場合と異なり、サービスリクエスタ側のプログラムで要求電文の XML を作成する必要はありません。サービスリクエスタではスタブに要求を出すときに、それぞれのパラメタに値をセットして実行します (スタブの中で SOAP メッセージを作成します)。
3. サービスリクエスタへの応答は、ビジネスプロセスの応答アクティビティで定義した応答電文の電文フォーマットで応答します (応答アクティビティに定義した電文フォーマットに合わせた WSDL を作成し、ユーザ定義受付に定義します)。

(b) JAX-WS エンジンを使用するとき

通信基盤に JAX-WS エンジンを使用するときの、SOAP 通信時のユーザ電文と WSDL の関係 (標準受付) を次の図に示します。

2. HCSC サーバのサービス部品呼び出し

図 2-13 JAX-WS エンジンを使用するときの SOAP 通信時のユーザ電文と WSDL の関係 (ユーザ定義受付)



(凡例)

➡ : サービス部品呼び出し要求・応答の流れ

➡ : ユーザがする処理

Ⓜ : フォルトコネクション

注※1 ビジネスプロセスの受付アクティビティや応答アクティビティに定義した形式に合わせた WSDL です。

注※2 ユーザが定義した WSDL に定義されている要求電文です。

注※3 ユーザが定義した WSDL に定義されている応答電文です。

1. ユーザ定義受付の場合、サービスリクエストで指定する要求電文は、ビジネスプロセスの受付アクティビティに定義した任意の電文フォーマットの形式で要求します (受

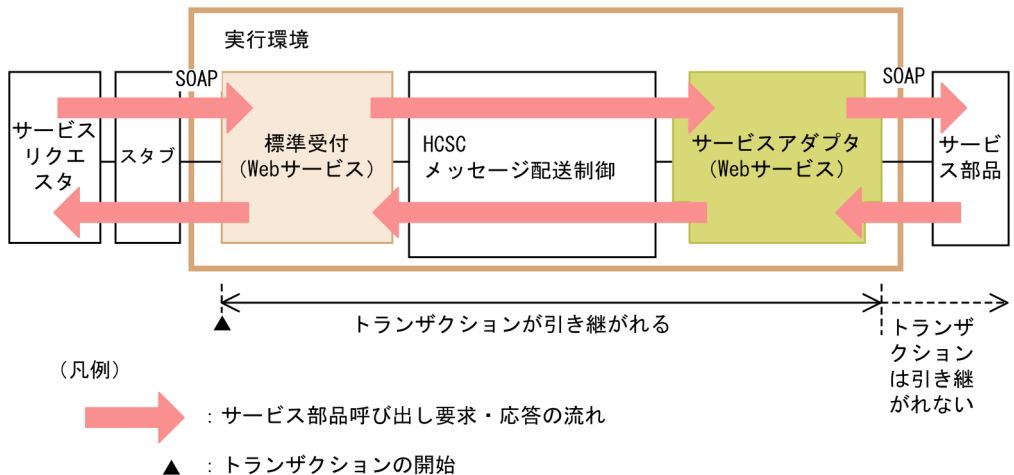
付アクティビティに定義した電文フォーマットに合わせた WSDL を作成し、ユーザ定義受付に定義します。

2. サービスクラスは、ユーザが定義した WSDL を基に、JAX-WS 機能が提供する `ejwsimport` コマンドを使用して作成できます。標準受付の場合と異なり、サービスリクエスタ側のプログラムで要求電文の XML を作成する必要はありません。サービスリクエスタではサービスクラスに要求を出すときに、それぞれのパラメータに値をセットして実行します（サービスクラスの中で SOAP メッセージを作成します）。
3. サービスリクエスタへの応答は、ビジネスプロセスの応答アクティビティで定義した応答電文の電文フォーマットで応答します（応答アクティビティに定義した電文フォーマットに合わせた WSDL を作成し、ユーザ定義受付に定義します）。

2.2.3 Web サービス（SOAP 通信）の場合のトランザクション

Web サービス（SOAP 通信）の場合、HCSC サーバ内で管理しているトランザクションは、サービス側には引き継がれません。Web サービス（SOAP 通信）の場合のトランザクションを次の図に示します。

図 2-14 Web サービス（SOAP 通信）の場合のトランザクション



2.2.4 Web サービス（SOAP 通信）の場合の Cookie 情報

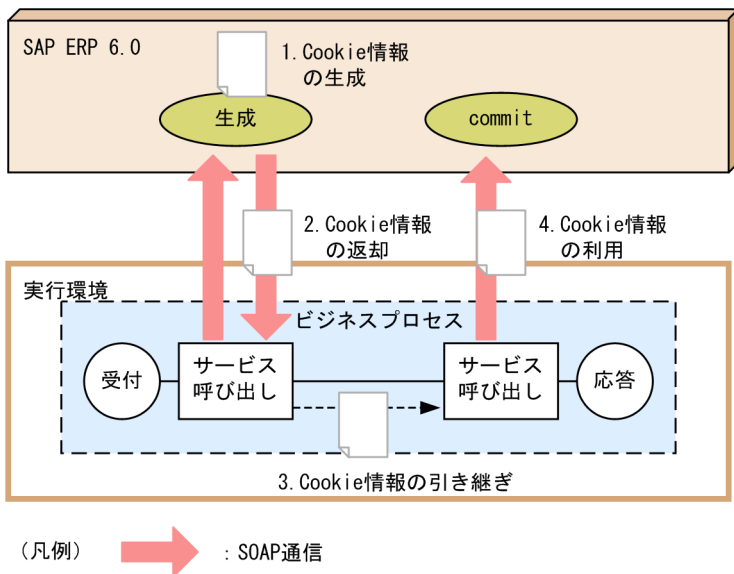
ビジネスプロセスから SAP ERP 6.0 上のサービスを呼び出す場合、更新系の処理によって返却された Cookie 情報を引き継ぐ必要があります。

SAP ERP 6.0 上のサービスを呼び出す場合で、Cookie 情報を引き継ぐには、HCSC ランタイム定義ファイルの `cookie-parsing` プロパティの指定によって、Cookie 情報の引き継ぎを有効にします。

2. HCSC サーバのサービス部品呼び出し

Cookie 情報の引き継ぎの流れを次の図に示します。

図 2-15 SAP ERP 6.0 のサービス呼び出しでの Cookie 情報の引き継ぎ



Cookie 情報としてサービス部品が発行する Set-Cookie ヘッダの情報が引き継がれます。

次に示す Cookie 情報の引き継ぎ条件をすべて満たす場合は、Set-Cookie ヘッダの属性が Cookie ヘッダに設定されます。

- Set-Cookie ヘッダの Cookie 名とその値 (NAME=VALUE) だけを設定する。
- 複数の Cookie 情報がある場合、「;」で区切って設定する。

条件を満たさない場合は、Cookie ヘッダに何も設定されません。

Cookie 情報の引き継ぎ範囲、引き継ぎ条件、およびログ出力について説明します。また、Cookie 情報の引き継ぎに関する注意事項を示します。

(1) Cookie 情報の引き継ぎ範囲

Cookie 情報を引き継ぐビジネスプロセスの範囲を次に示します。

1. 初回のサービス呼び出しから、受付アクティビティおよび応答アクティビティを実行するまで
 2. 初回のサービス呼び出しから、ビジネスプロセスインスタンスが終了するまで
1. の範囲では、次に示す場合でも Cookie 情報が引き継がれます。
- 階層的にビジネスプロセスを呼び出す場合
呼び出し先のビジネスプロセスに受付アクティビティおよび応答アクティビティがあっても Cookie 情報は引き継がれます。

- 受付アクティビティが非同期のビジネスプロセスを呼び出す場合
通信モデルが非同期の場合、受付アクティビティで定義したオペレーション名に対する応答アクティビティは配備できません。この場合はサービス呼出アクティビティ間に受付アクティビティがないときだけ Cookie 情報が引き継がれます。
- 応答アクティビティのあとにアクティビティが定義されているビジネスプロセスを呼び出す場合
この場合は、サービス呼出アクティビティ間に受付アクティビティがないときだけ Cookie 情報が引き継がれます。
- 待機アクティビティが定義されているビジネスプロセスを呼び出す場合
応答アクティビティが実行されたあとに待機アクティビティがある場合、その待機アクティビティの前後にサービス呼出アクティビティがあっても、Cookie 情報は引き継がれます。

ただし、次の場合は引き継ぎ範囲に関係なく、Cookie 情報は引き継がれません。

- 実行環境のプロセスがダウンした場合
- 複数の実行環境のプロセスでリクエストを処理する場合
- プロセスインスタンスを再実行した場合

(2) Cookie 情報の引き継ぎ条件

サービス部品を呼び出すときの情報と呼び出し先の情報に関して、次に示す条件をすべて満たす場合に Cookie 情報が引き継がれます。

- HCSC ランタイム定義ファイルの `cookie-parsing` プロパティに ON を指定している
- 「サービス部品のホスト名:ポート番号」の文字列が完全一致する
ホスト名の指定形式が異なっていると、それぞれ別の URL として扱われます。例えば、同じホストを指定する場合に、ホスト名で指定しているものと IP アドレスで指定しているものが混在しているときは、それぞれ別の URL として扱われます。このとき、Cookie 情報は引き継がれません。また、ポート番号が設定されていない場合は、「:ポート番号」のない文字列で比較されます。
- path 属性が前方一致する
前方一致と見なされる場合および見なされない場合の例を示します。
(例) 前方一致と見なされる場合
`/sap` と `/sap` , `/sap` と `/sap/aaa` , `/sap` と `/sapaaa/bbb`
(例) 前方一致と見なされない場合
`/sap` と `/aaa` , `/sap` と `/aaasap/bbb` , `/sap` と `/aaa/sap`

注意事項

- Cookie 情報の引き継ぎは、SAP ERP 6.0 の Set-Cookie ヘッダ仕様の「path 属性」によって判断されます。ただし、SOAP 通信基盤 (SOAP1.1 モード) を使用する場合は、「path 属性」は考慮されません。SOAP 通信基盤では、ホスト名が完全一致している場合に Cookie 情報が引き継がれます。
- Set-Cookie ヘッダに path 属性が含まれていない場合は、デフォルト値として

2. HCSC サーバのサービス部品呼び出し

「path=/」が設定されます。

- 次の例のように、Set-Cookie ヘッダに属性名が「path」の属性が複数設定されている場合は、path 属性に「path=/」が設定されたものとして動作します。

(例)

```
Set-Cookie: NAME=value; path=/sap/aaa; path=/sap/bbb
```

- 複数の異なる path 属性値を持つ Cookie 情報を HTTP ヘッダに設定する場合、Set-Cookie の読み込み順序に依存しないで、path 属性値が詳細な順に HTTP ヘッダに設定されます。

例えば、次の Set-Cookie ヘッダを受け取った場合、HTTP ヘッダには

「cookie:NAME3=value3; NAME2=value2; NAME1=value1」が設定されます。

(例)

```
Set-Cookie: NAME1=value1; path=/  
Set-Cookie: NAME2=value2; path=/sap  
Set-Cookie: NAME3=value3; path=/sap/aaa
```

(3) ログ出力

HCSC サーバランタイム定義ファイルの methodtrace-level プロパティに「2」以上の値を設定した場合に、メソッドトレースにログが出力されます。

(4) 注意事項

次に示す仕様および属性は、サポートしていません。

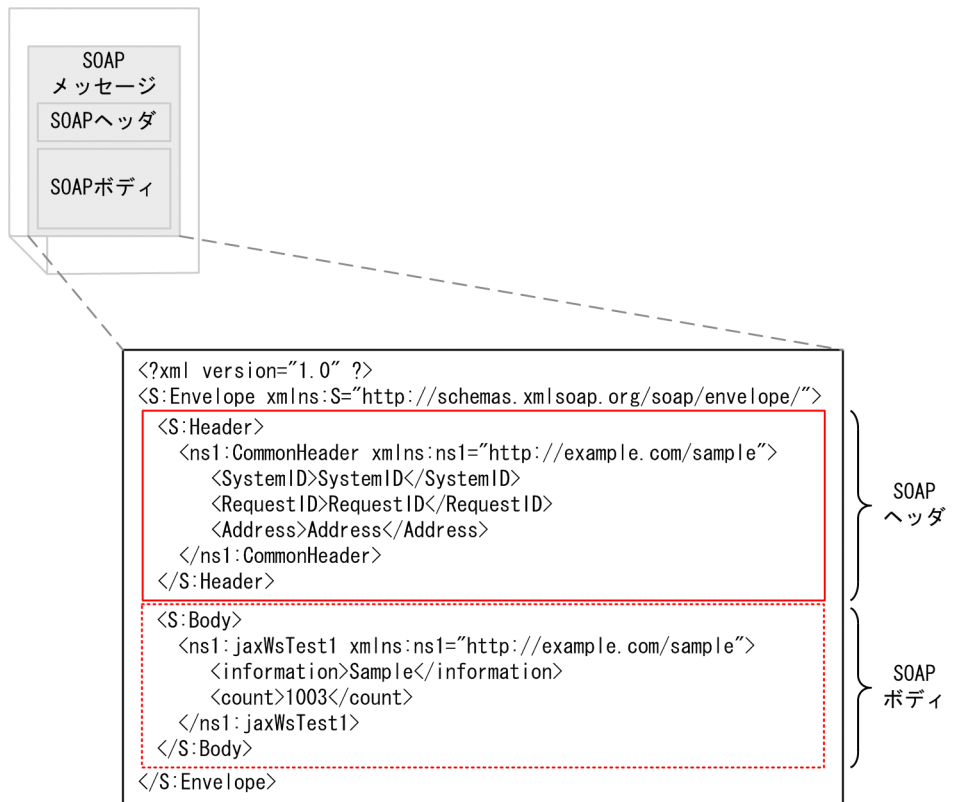
- Netscape Communications 社の Set-Cookie ヘッダの仕様である「domain 属性」、
「expires 属性」、および「secure 属性」
- RFC2109 の Set-Cookie ヘッダの仕様である「Comment 属性」、「Domain 属性」、
「Max-Age 属性」、および「Secure 属性」
- RFC2109 でサポートされている Cookie 仕様
Set-Cookie ヘッダの「Comment 属性」、「Domain 属性」、「Max-Age 属性」、および
「Secure 属性」はサポートしていません。また、Set-Cookie ヘッダの「Version 属性」
を Cookie ヘッダの「cookie-version 属性」に設定する必要があります。SAP ERP 6.0
が発行する Set-Cookie ヘッダには「Version 属性」は含まれません。
なお、Set-Cookie ヘッダの「Version 属性」に複数の Cookie 情報で異なる値が設定さ
れた場合は、最初に読み込んだ Cookie 情報が持つ「Version 属性」の値が採用されま
す。
- 「JSESSIONID」
SAP ERP 6.0 では、sap-contextid 属性でセッション管理を行っています。
- RFC2965 でサポートされている Set-Cookie2 の仕様

2.2.5 Web サービス (SOAP 通信) 場合の SOAP メッセージの構成

Web サービス (SOAP 通信) の場合、要求電文および応答電文としてやり取りしている SOAP メッセージは、ヘッダ (SOAP ヘッダ) とボディ (SOAP ボディ) で構成されています。SOAP ヘッダにはメッセージの付加的な情報 (識別情報など) が含まれ、SOAP ボディにはメッセージ本体が含まれています。

SOAP メッセージの構成例を次の図に示します。

図 2-16 SOAP メッセージの構成例



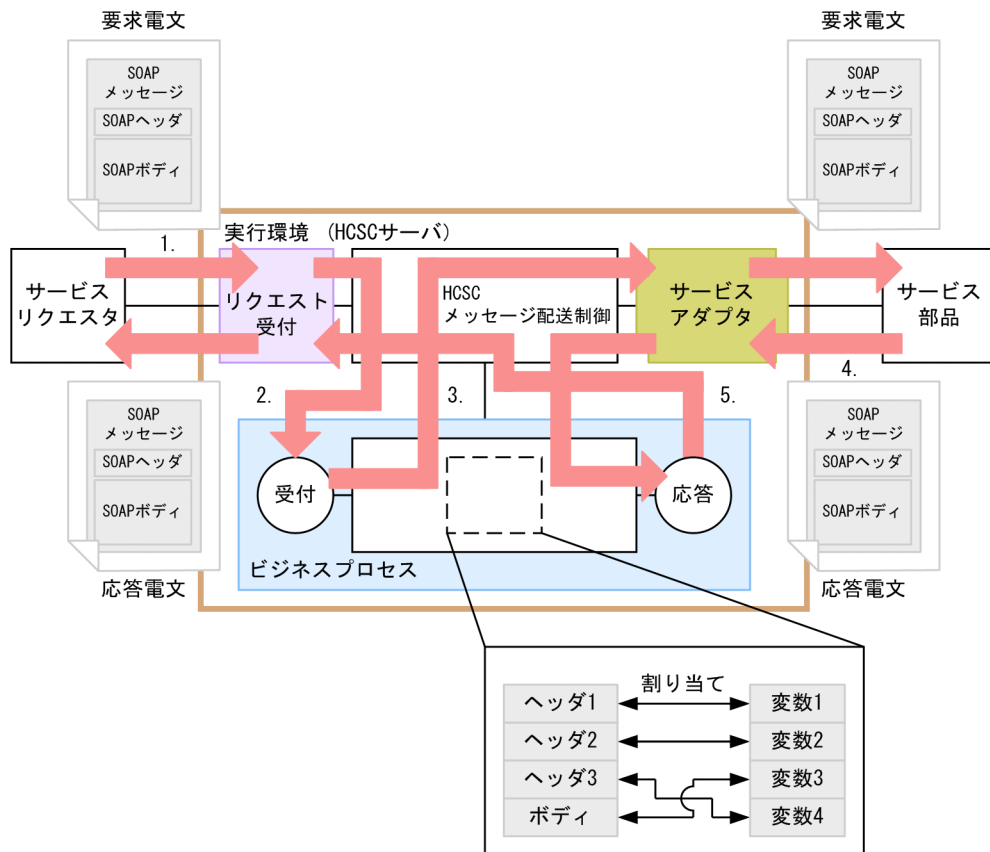
(1) SOAP メッセージとビジネスプロセスの関係

SOAP メッセージの場合、要求電文のヘッダ情報とボディ情報は、業務処理で利用できるようにビジネスプロセスの変数に割り当てられます。また、応答電文として返されるときは、変数から再びヘッダ情報とボディ情報を生成します。

SOAP メッセージとビジネスプロセスの関係を次の図に示します。

2. HCSC サーバのサービス部品呼び出し

図 2-17 SOAP メッセージとビジネスプロセスの関係



図中の 1. ~ 5. について説明します。

説明

1. サービスリクエスタから受付に対して、業務処理に必要なパラメタを設定した要求電文（SOAP メッセージ）が渡されます。
2. 受付からビジネスプロセスが起動されます。このとき、受付が受信した SOAP メッセージに設定されているヘッダ情報およびボディ情報を、業務処理で利用できるようにビジネスプロセスの変数に割り当てます。
3. ビジネスプロセスが業務処理を実行します。サービス呼び出すときは、変数からサービスの実行に必要なパラメタを設定した SOAP メッセージを生成し、サービスアダプタ経由でサービス部品に送信します。
4. サービス部品は処理が終わると、その結果を応答電文（SOAP メッセージ）として返します。ビジネスプロセスはサービスアダプタ経由でこの SOAP メッセージを受け取り、ヘッダやボディの情報を再び変数に割り当てます。
5. ビジネスプロセスの処理が終わると、変数から SOAP メッセージを生成し、受付を経由してサービスリクエスタに結果を返します。

(2) ユーザ電文トレースに出力される SOAP メッセージの情報

SOAP メッセージの場合、ユーザ電文トレースデータにはユーザ電文に含まれる次の情報が出力されます。

- soap:Header 要素を含む複数の子要素
- soap:Body 要素の一つの子要素

なお、soap:Header 要素が存在しない場合は、soap:Body 要素の子要素だけが情報として出力されます。

soap:Header 要素および soap:Body 要素については、マニュアル「Cosminexus サービスプラットフォーム 開発ガイド」の「2.6.1 Web サービスを利用したサービス部品の適用範囲」を参照してください。

2.2.6 Web サービス (SOAP 通信) の場合の接続先動的変更

サービスアダプタ (Web サービス) の通信先は、ビジネスプロセスで設定した接続先 URL を使って動的に変更できます。ビジネスプロセスで接続先 URL を設定しておくことで、サービスアダプタを実行したときに通信先が変更されます。これにより、作成するサービスアダプタの数を削減できます。

なお、ビジネスプロセスで接続先 URL を設定しなかった場合は、開発環境で設定した Web サービスの接続先 URL で SOAP 通信を行います。

(1) 接続先動的変更の処理の流れ

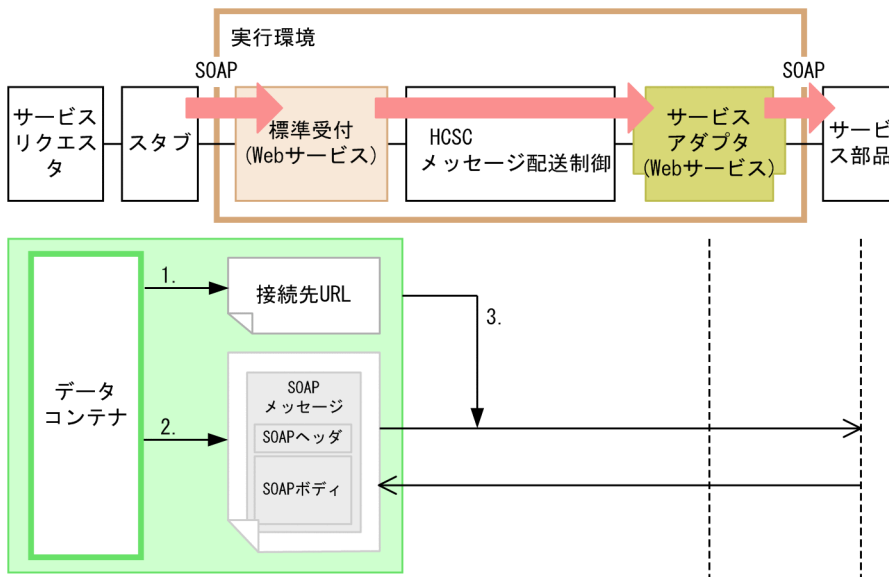
通信基盤ごとの、SOAP 通信時の接続先動的変更の処理の流れを次に説明します。

(a) SOAP 通信基盤を使用するとき

通信基盤に SOAP 通信基盤を使用するときの接続先動的変更の処理の流れを次の図に示します。

2. HCSC サーバのサービス部品呼び出し

図 2-18 通信基盤に SOAP 通信基盤を使用するときの接続先動的変更の処理の流れ



(凡例)

- ➡ : サービス部品呼び出し要求の流れ ➡ : 電文の流れ
 ➡ : データの流れ

説明

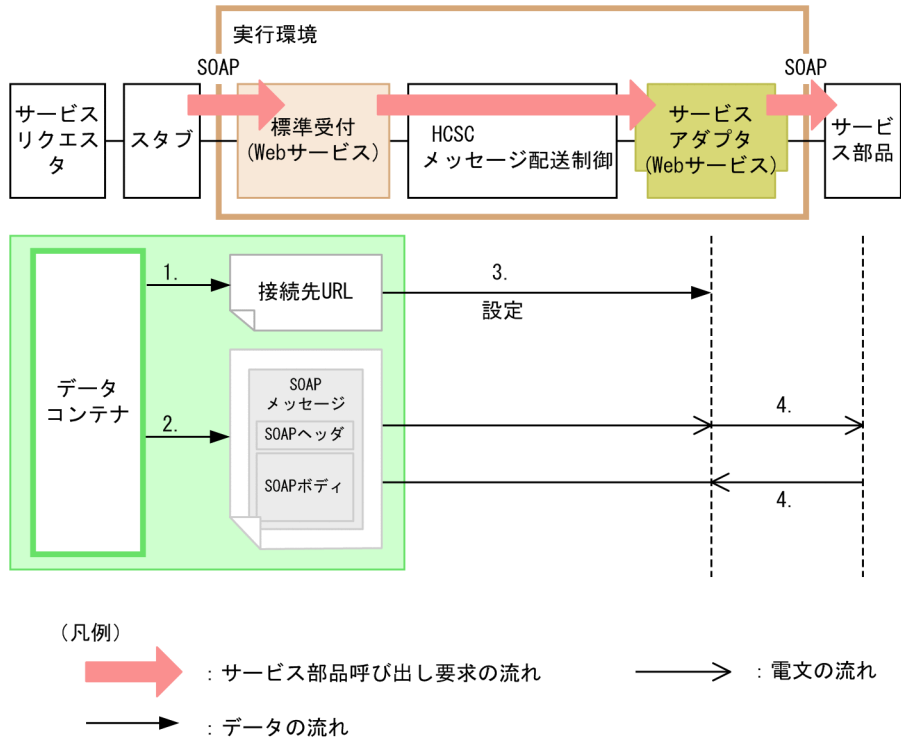
図中の番号は、次の説明の番号と対応しています。

1. データコンテナ内のヘッダ用メッセージコンテナから接続先 URL を取得します。
2. データコンテナの情報を元に SOAP メッセージを作成します。
3. 取得した接続先 URL でサービスと通信を行います。

(b) JAX-WS エンジンを使用するとき

通信基盤に JAX-WS エンジンを使用するときの接続先動的変更の処理の流れを次の図に示します。

図 2-19 通信基盤に JAX-WS エンジンを使用するときの接続先動的変更の処理の流れ



説明

図中の番号は、次の説明の番号と対応しています。

1. データコンテナ内のヘッダ用メッセージコンテナから接続先 URL を取得します。
2. データコンテナの情報をもとに SOAP メッセージを作成します。
3. 接続先 URL を設定します。
4. 設定した接続先 URL のサービスと通信を行います。

(2) 接続先 URL の取得

接続先を動的に変更するために、開発環境で接続先 URL をサービス呼び出しアクティビティのヘッダ割当変数に設定します。接続先 URL の取得の詳細については、マニュアル「Cosminexus サービスプラットフォーム 開発ガイド」の「8.3.2 サービスアダプタの接続先情報を動的に変更する例」を参照してください。

(3) ベーシック認証との関係

ベーシック認証の利用可否、およびベーシック認証で利用するユーザ ID とパスワードについては、接続先 URL を使って動的に変更できません。

ベーシック認証に関する情報は、開発環境の画面から設定した情報が有効となります。

2. HCSC サーバのサービス部品呼び出し

(4) 注意事項

次の場合、エラーメッセージが出力されて異常終了します。

- 開発環境で複数の接続先情報がヘッダ割当変数に設定された場合
XML データのシステム固有の名前空間、要素名が複数存在すると、接続先が特定できないためエラーになります。
- 設定された接続先情報が不正な場合
接続先 URL に誤りがあると、サービス呼び出し時にエラーになります。
- 接続先情報を示す XML データの endpointURL 要素が存在しない場合
endpointURL 要素がない場合、接続先が特定できないためエラーになります。

2.3 SessionBean を使ったサービス部品呼び出し

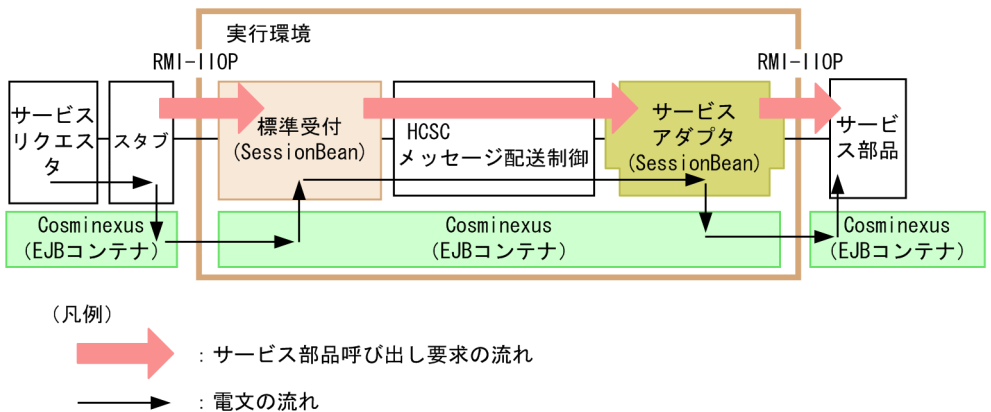
2.3.1 Cosminexus (EJB コンテナ) との関係

実行環境の HCSC サーバは、Cosminexus の EJB コンテナ上で動作します。

HCSC サーバのリクエスト受付は、EJB (Stateless Session Bean) のリモートインターフェースとして提供され、RMI-IIOP で接続します。

HCSC サーバのサービスアダプタからは、EJB (Stateless Session Bean または Statefull Session Bean) で作成されたサービス部品が実行されます。Cosminexus (EJB コンテナ) との関係を次の図に示します。

図 2-20 Cosminexus (EJB コンテナ) との関係

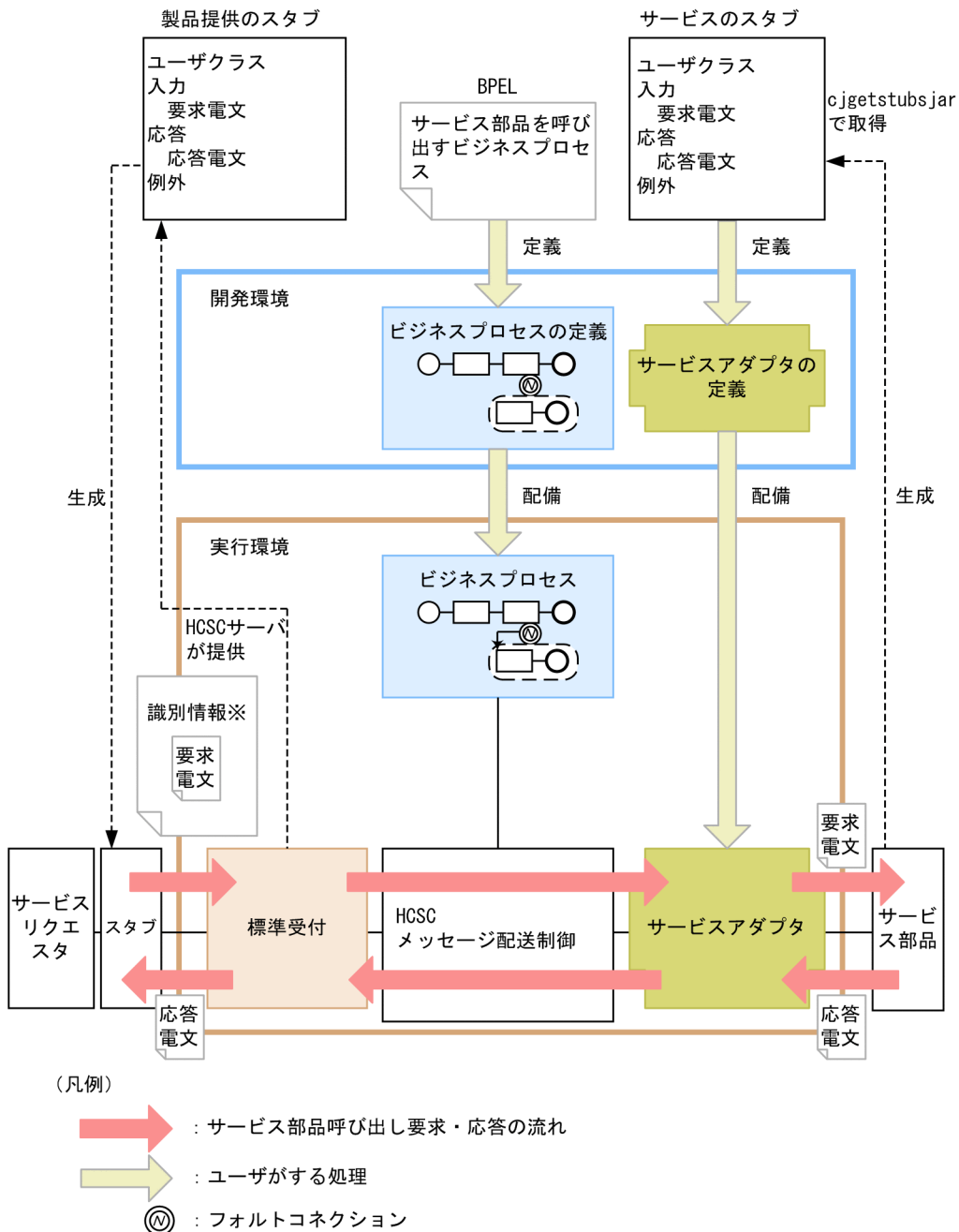


2.3.2 サービス部品呼び出し時のユーザ電文とサービス部品のスタブとの関係

SessionBean 使用時のユーザ電文とスタブの関係を次の図に示します。

2. HCSC サーバのサービス部品呼び出し

図 2-21 SessionBean 使用時のユーザ電文とスタブの関係



注※ 識別情報とは、サービス名、クライアント相関IDなどです。

サービスアダプタを定義するときは、サービス部品のスタブを定義します（サービス部品のスタブには、作成したサービス部品のユーザクラスが含まれています。また、要求電文、応答電文および例外の形式が定義されています）。

サービス部品側の J2EE サーバに Cosminexus を使用している場合、サービス部品のスタブを取得するには、J2EE サーバの `cjgetstubsjar` コマンドを使用します。

サービスリクエストからサービス部品を呼び出すときは、製品提供のスタブを使用します。製品提供のスタブは、標準受付のインターフェースになっています。サービスリクエストから標準受付を呼び出すときには、サービス部品側の電文フォーマットに合わせた要求電文を事前に作成し、標準受付のパラメタ（ユーザ電文のパラメタ）に設定して実行します。

サービスリクエストからビジネスプロセスを呼び出すときは、要求電文はビジネスプロセスの受付アクティビティに定義した電文フォーマットになります。サービスリクエスト内で、ビジネスプロセスの受付アクティビティで定義されている要求電文の電文フォーマットに合わせた電文を作成し、作成した電文を標準受付のパラメタ（ユーザ電文のパラメタ）に設定して実行します。

サービスリクエストへの応答は、サービスアダプタで定義した応答電文の電文フォーマットを使用します。

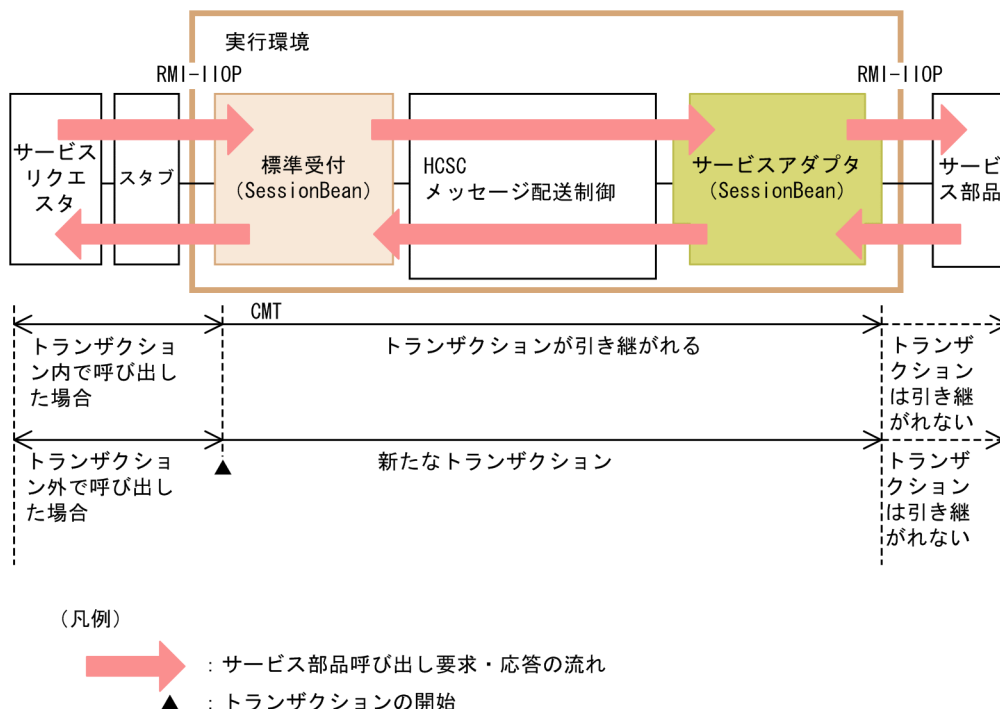
`cjgetstubsjar` コマンドの詳細については、マニュアル「Cosminexus アプリケーションサーバリファレンス コマンド編」の「`cjgetstubsjar`（アプリケーションの RMI-IIOP スタブおよびインタフェースの取得）」を参照してください。

2.3.3 SessionBean 使用時のトランザクション

SessionBean 使用時のトランザクションを次の図に示します。

2. HCSC サーバのサービス部品呼び出し

図 2-22 SessionBean 使用時のトランザクション

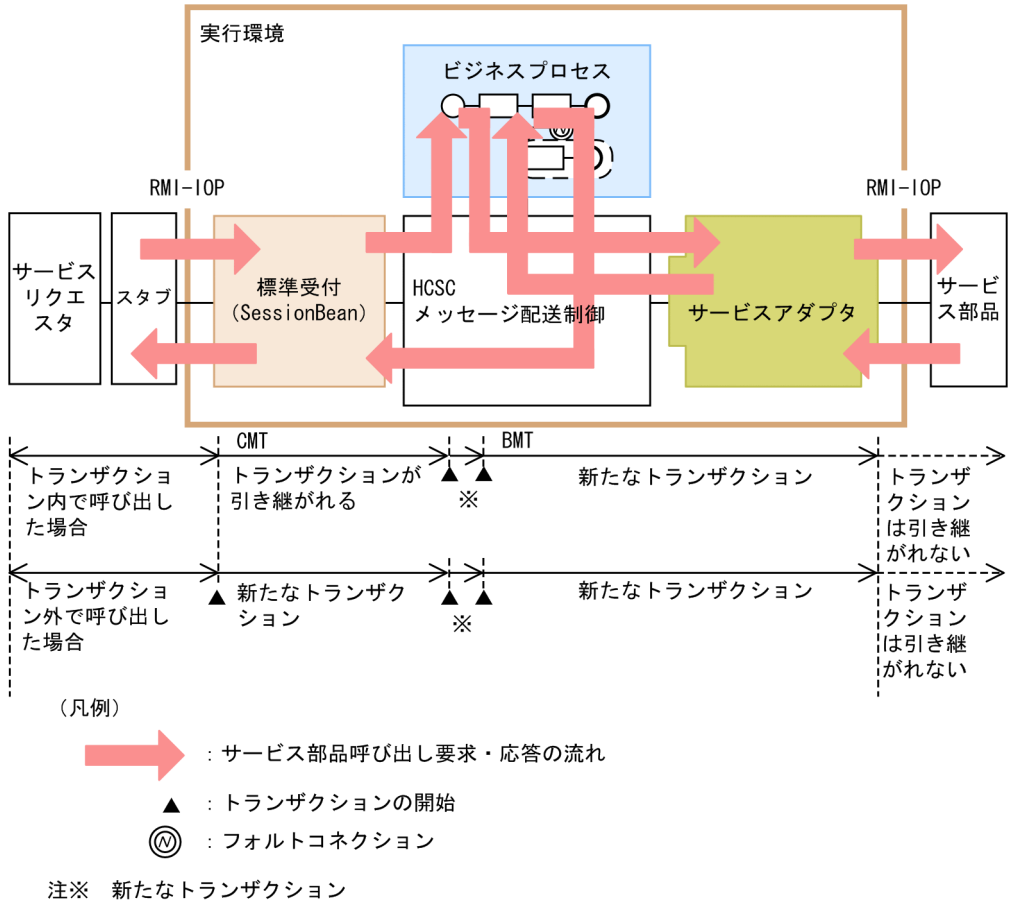


SessionBean 使用時，HCSC サーバ内は CMT (Container Managed Transaction) で動作します。また，標準受付のトランザクション属性は Required 属性です。そのため，サービスリクエスタがトランザクションスコープ内で HCSC サーバの標準受付を呼び出したとき，トランザクションコンテキストが HCSC サーバ側に引き継がれ，HCSC サーバ内の処理はサービスリクエスタ側のトランザクションスコープ内に入ります。また，サービスリクエスタがトランザクションスコープ外で HCSC サーバの標準受付を呼び出したとき，HCSC サーバ側で新しいトランザクションが開始されます。

なお，サービス部品のトランザクション属性は RequiresNew または NotSupported でなければなりません。そのため，トランザクションは，サービス部品側には引き継がれません。SessionBean を利用したサービス部品のトランザクション属性を指定できるかどうかについては，マニュアル「Cosminexus サービスプラットフォーム 開発ガイド」の「2.6.2 SessionBean を利用したサービス部品の適用範囲」を参照してください。

ビジネスプロセスは BMT (Bean Managed Transaction) で動作します。ビジネスプロセス自体は新たなトランザクションで制御します。ビジネスプロセスを呼び出す場合は，ビジネスプロセスのサービス呼出アクティビティに定義したサービス部品呼び出し処理ごとに，新たなトランザクションで制御します。そのため，サービスリクエスタがトランザクションスコープ内で HCSC サーバの標準受付を呼び出したときでも，トランザクションはサービスアダプタまで引き継がれません。ビジネスプロセスを介した場合のトランザクションを次の図に示します。

図 2-23 ビジネスプロセスを介した場合のトランザクション



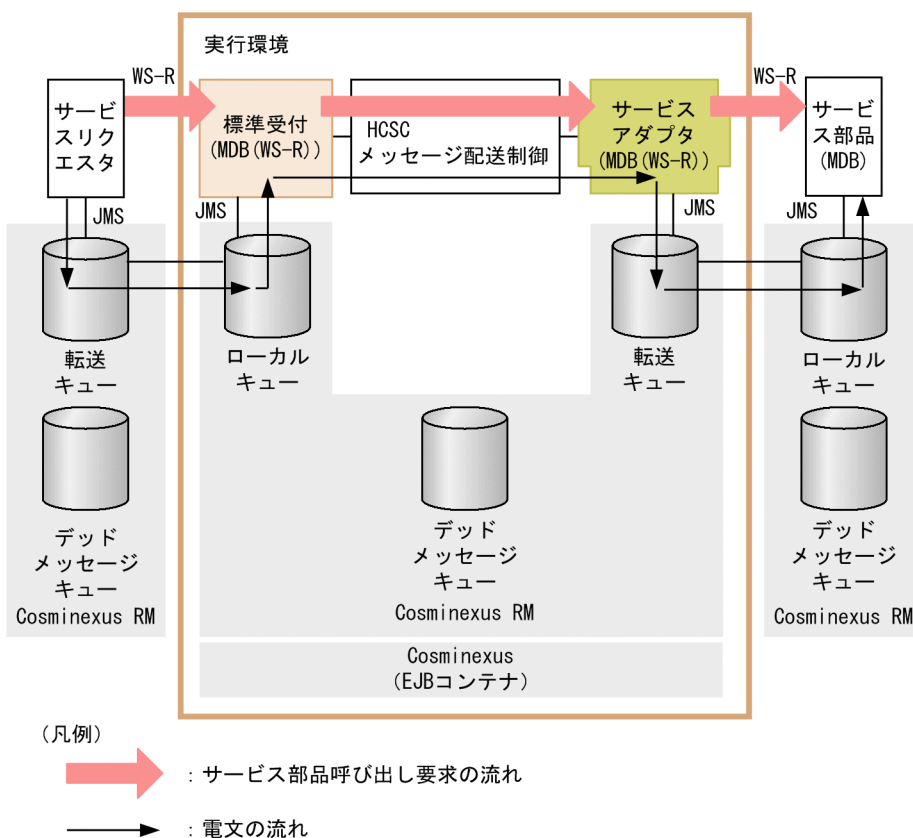
永続化するビジネスプロセスの場合、ビジネスプロセスを実行する過程でのトランザクションについては、「4.4 ビジネスプロセスのトランザクション」を参照してください。

2.4 MDB (WS-R) を使ったサービス部品呼び出し

2.4.1 Cosminexus RM との関係

非同期のサービス部品を呼び出すには、Cosminexus RM を使用します。また、サービスリクエスト側の転送キューと HCSC サーバ側のローカルキューの間、そして HCSC サーバ側の転送キューとサービス部品側のローカルキューの間は WS-R (WS-Reliability) を使用します。Cosminexus RM は、Cosminexus の EJB コンテナ上で動作します。Cosminexus RM との関係を図に示します。

図 2-24 Cosminexus RM との関係



2.4.2 サービス部品呼び出し時のメッセージの流れ

HCSC サーバの標準受付は、MDB (Message Driven Bean) で動作し、HCSC サーバ側のローカルキューを監視しています。HCSC サーバ側のローカルキューにメッセージが

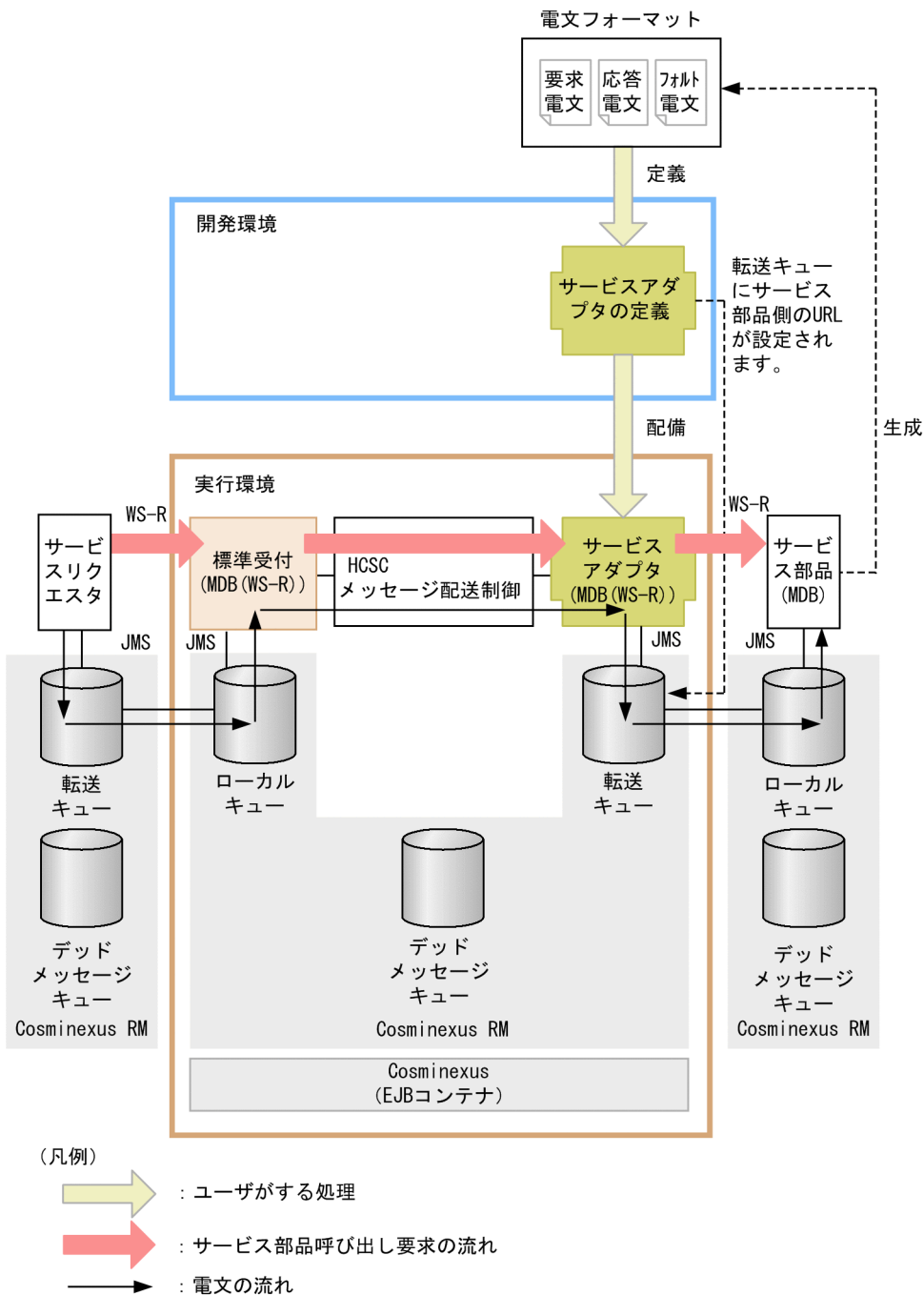
送信されると、標準受付はメッセージを取り出してサービス部品を呼び出します。

サービスリクエスタ側の転送キューには、HCSC サーバの宛先アドレスおよび転送先のローカルキューのキュー名を設定します。サービスリクエスタからは、サービスリクエスタ側の転送キューに対して、JMS メッセージを送信して、HCSC サーバの標準受付を呼び出します。

サービス部品を呼び出すときには、サービスアダプタ側の転送キューに送信します。そして、転送キューは、開発環境のサービスアダプタ定義画面（基本画面のアドレスに指定した転送先のキュー名および詳細画面の宛先 URL）で定義したサービス部品側のローカルキューへ送信します。サービス部品を MDB で作成することで、サービス部品側のローカルキューに送信されたメッセージを取り出して処理できます。非同期（MDB（WS-R））の場合の宛先の指定を次の図に示します。

2. HCSC サーバのサービス部品呼び出し

図 2-25 非同期 (MDB (WS-R)) の場合の宛先の指定



転送キューを作成する手順については、マニュアル「Cosminexus サービスプラットフォーム 開発ガイド」の「10.5.2 転送キューを作成する」を参照してください。JMSメッセージを生成する手順については、マニュアル「Cosminexus サービスプラット

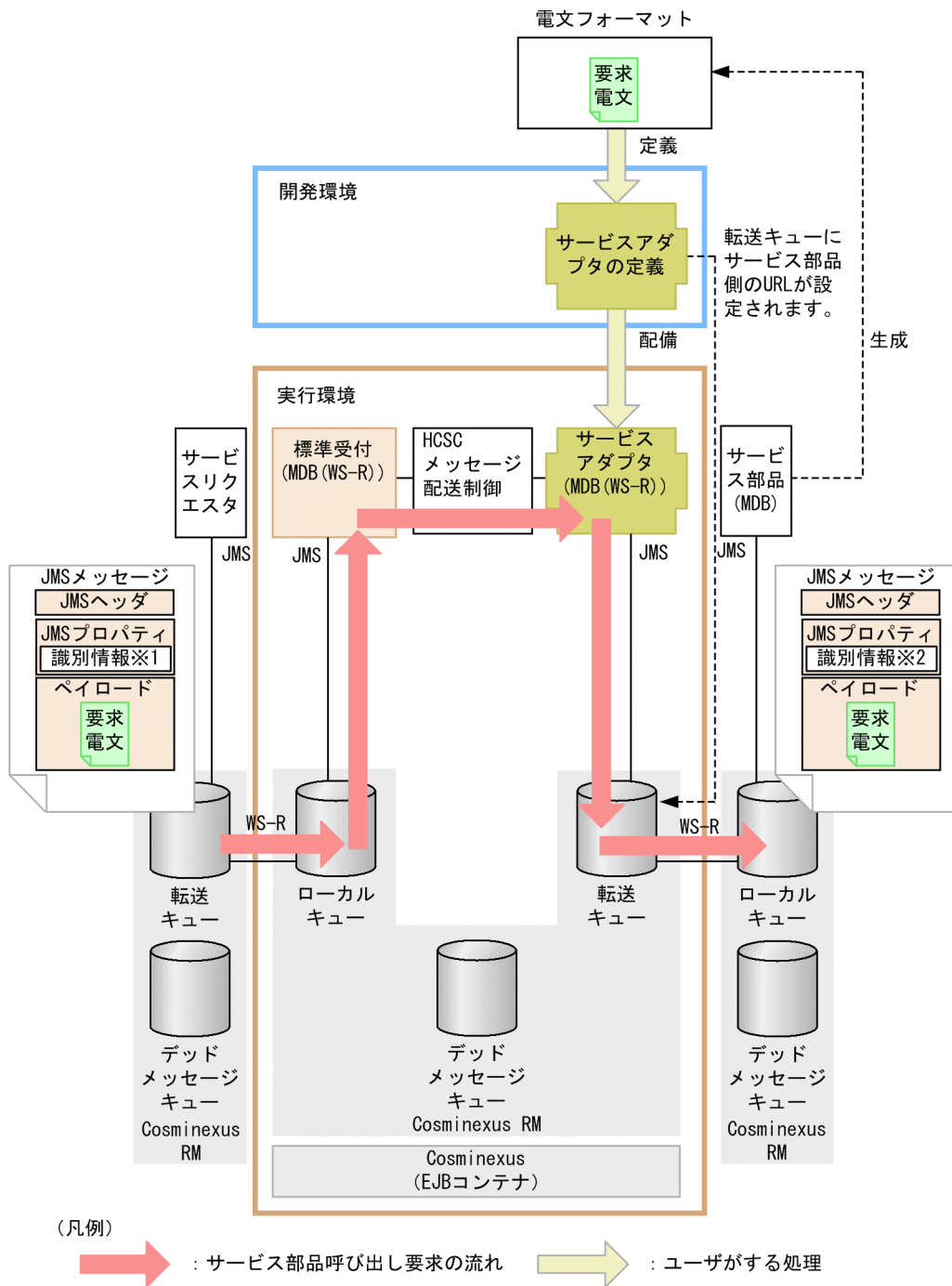
フォーム 開発ガイド」の「10.5.3 JMS メッセージを生成する」を参照してください。

サービスリクエストからサービス部品を呼び出すときに転送キューに送信する JMS メッセージの形式は、標準受付のインターフェースになります。

JMS メッセージのペイロード（ヘッダを除いたデータ本体）に設定した内容が、ユーザ電文としてサービス部品に伝わります。そのため、サービスリクエストから標準受付を呼び出すときに、サービス部品側の電文フォーマットに合わせた要求電文を事前に作成し、ペイロードに設定してメッセージを送信します。非同期サービス部品呼び出し時のユーザ電文と JMS メッセージの関係を次の図に示します。

2. HCSC サーバのサービス部品呼び出し

図 2-26 非同期サービス部品呼び出し時のユーザ電文と JMS メッセージの関係

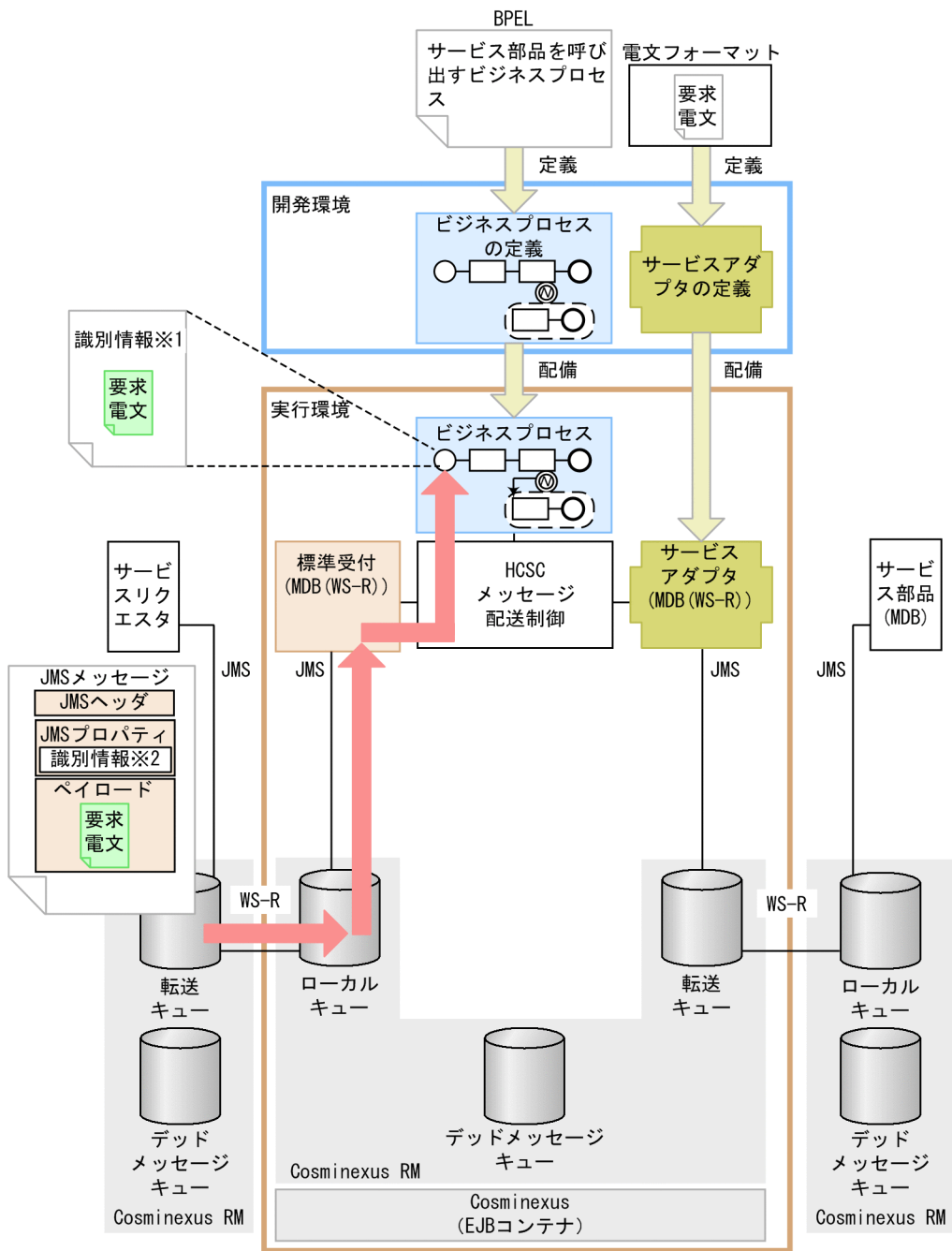


注※1 識別情報とは、サービス名、クライアント相関ID、およびユーザ定義プロパティです。
 注※2 識別情報とは、ユーザ定義プロパティです。

サービスリクエスタからビジネスプロセスを呼び出すときは、要求電文はビジネスプロセスの受付アクティビティに定義した電文フォーマットになります。サービスリクエスタ内でビジネスプロセスの受付アクティビティで定義されている要求電文の電文フォーマットに合わせた電文を作成し、ペイロードに設定してメッセージを送信します。ビジネスプロセスを呼び出す場合のユーザ電文と JMS メッセージの関係を次の図に示します。

2. HCSC サーバのサービス部品呼び出し

図 2-27 ビジネスプロセスを呼び出す場合のユーザ電文と JMS メッセージの関係



(凡例) ➡ : サービス部品呼び出し要求の流れ ➡ : ユーザがする処理

Ⓝ : フォルトコネクション

注※1 識別情報とは、サービス名、クライアント相関IDなどです。

注※2 識別情報とは、サービス名、クライアント相関ID、およびユーザ定義プロパティです。

2.4.3 同期のサービス部品を呼び出す場合の応答電文の扱い

標準受付 (MDB (WS-R)) は、非同期の要求電文用の標準受付で、基本的にサービス部品からの応答がない場合に使用します。しかし、非同期受付 (MDB (WS-R)) から同期のサービス部品 (Web サービスまたは SessionBean など) やビジネスプロセスを要求した場合、同期のサービス部品から応答が返ることがあります。そのため、HCSC サーバでは応答を応答用のキューに送信する手段を提供しています。

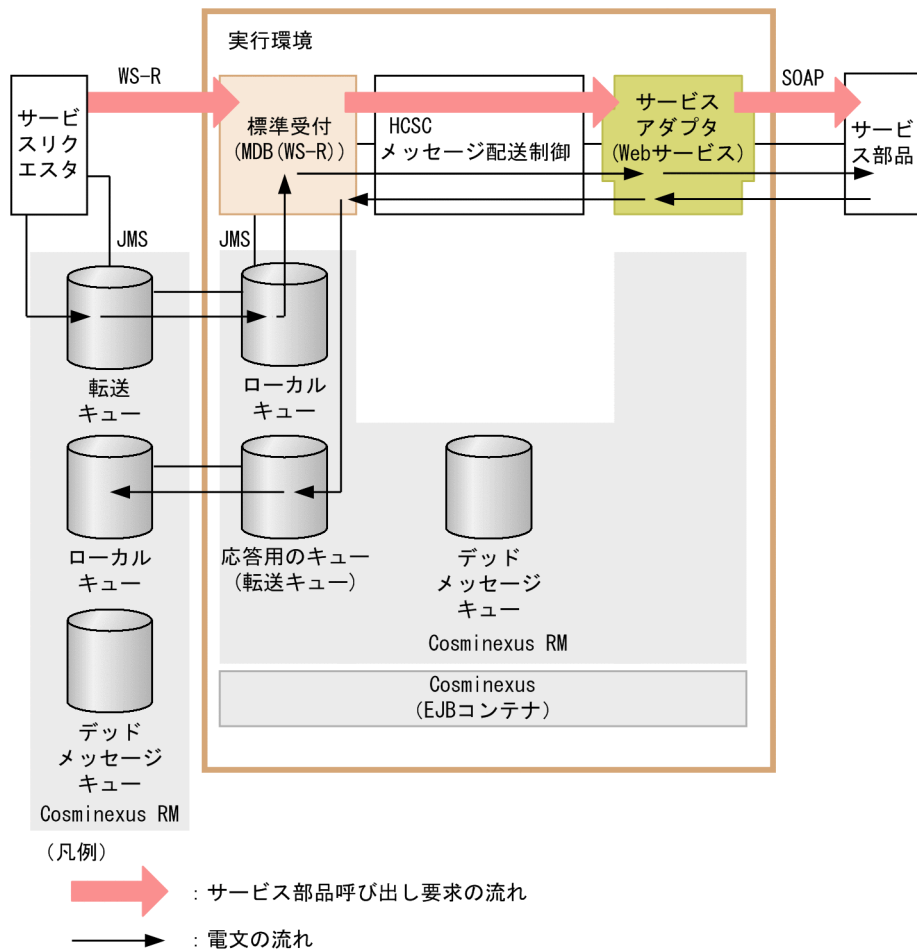
応答用のキューを利用するには、HCSC サーバの環境構築時に HCSC サーバ側に転送キュー (応答用のキュー) を作成し、転送キューにサービスリクエスト側マシンのキュー間転送用 Web アプリケーションの宛先アドレスおよび転送先キュー名を指定しておきます。

サービスリクエストから標準受付に対してサービス部品の呼び出しを要求するとき、標準受付のパラメタに作成した転送キュー (応答用のキュー) のキュー名を指定すると、サービス部品からの応答が返ってきたときに、指定されたキューに JMS メッセージが送信されます。

転送キュー (応答用のキュー) に送信された JMS メッセージは、サービスリクエスト側のローカルキューに転送されるため、ローカルキューからメッセージを取り出すことで、サービスリクエストは、サービス部品からの応答電文を取得できます。同期サービス部品を呼び出した場合の応答の扱いを次の図に示します。

2. HCSC サーバのサービス部品呼び出し

図 2-28 同期サービス部品を呼び出した場合の応答の扱い



応答を取り出す方法については、マニュアル「Cosminexus サービスプラットフォーム 開発ガイド」の「10.5.8 応答を取り出す」を参照してください。

同期サービス部品でも応答があるかどうかによって、次の表に示すように応答用のキューへメッセージを送信するかどうか異なります。

なお、サービスリクエストから応答用のキューのキュー名を指定していても、HCSC サーバ側に指定したキュー名のキューが作られていない場合は、応答用のキューにメッセージは送信できないため、サービス部品の呼び出し処理がエラーになります。

表 2-1 応答用のキューへメッセージを出力する場合

項番	応答用のキューへ出力する場合	応答電文の有無			
		応答電文がある場合	応答電文がない場合	サービス部品からのユーザ定義例外の場合	ユーザ定義例外以外のエラーの場合
1	呼び出し先のサービス部品が同期サービス部品の場合 (WebService, SessionBean およびカスタムアダプタなど)	ペイロードに応答電文を設定して送信	ペイロードなしのメッセージを送信	ペイロードに応答電文を設定して送信	x
2	呼び出し先のサービス部品が非同期サービス部品の場合 (MDB (WS-R) および MDB (DB キュー) など)	x	x	-	x
3	呼び出し先のサービス部品がビジネスプロセスの場合	ペイロードに応答電文を設定して送信	ペイロードなしのメッセージを送信	ペイロードに応答電文を設定して送信	x

(凡例)

x : メッセージを送信しません。

- : 非同期のサービス部品からユーザ定義例外は存在しないため該当しません。

注 エラーの情報はログおよびトレースに出力します。

サービスリクエストからの要求で応答用のキューのキュー名を指定していない場合、電文の実行履歴を採取している場合は、応答電文の内容は破棄されます (応答電文が破棄されると、KDEC00031-W メッセージが出力されます)。

また、サービスリクエスト側で相関関係を対応づけるために、サービスリクエストが指定した JMS プロパティが引き継がれます。引き継がれるヘッダおよびプロパティの詳細については、マニュアル「Cosminexus サービスプラットフォーム 開発ガイド」の「10.5.8 応答を取り出す」を参照してください。

2.4.4 MDB (WS-R) 使用時のトランザクション

MDB (WS-R) の場合、サービスリクエストから HCSC サーバの間、そしてサービスアダプタからサービス部品の間は、異なるトランザクションになります。

(1) 非同期のサービスアダプタへメッセージを送信する場合

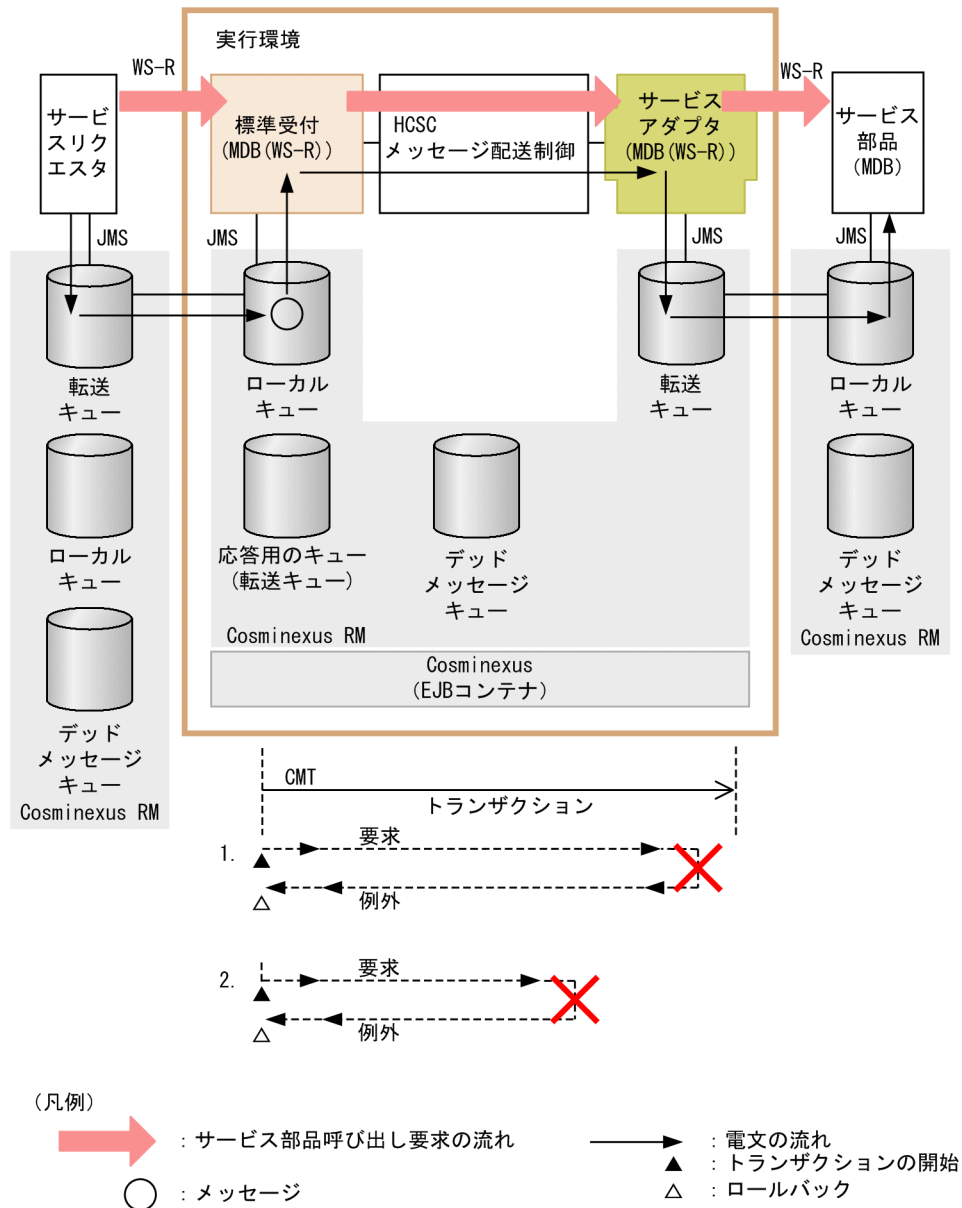
非同期 (MDB (WS-R)) の標準受付でサービス部品呼び出し要求を受け付け、非同期の (MDB (WS-R)) サービスアダプタへメッセージを送信する場合、標準受付でのローカルキューからのメッセージの取り出しとサービスアダプタでの転送キューへの送信は、一つのトランザクションで処理します。

2. HCSC サーバのサービス部品呼び出し

転送キューへのメッセージ送信が失敗した場合や、指定したサービスアダプタが起動していない場合など、何らかのエラーが発生した場合は、トランザクションをロールバックします。

ロールバックすると、標準受付で取り出したメッセージが復元され、再度ローカルキューからメッセージを取り出して、サービス部品を呼び出します。受付からサービス部品呼び出しまでが非同期の場合のトランザクションについて次の図に示します。

図 2-29 受付からサービス部品呼び出しまでが非同期の場合のトランザクション



エラーが発生したときの対処方法については、マニュアル「Cosminexus サービスプラットフォーム システム構築・運用ガイド」の「7.7.3 MDB (WS-R) 実行時の障害対策」を参照してください。

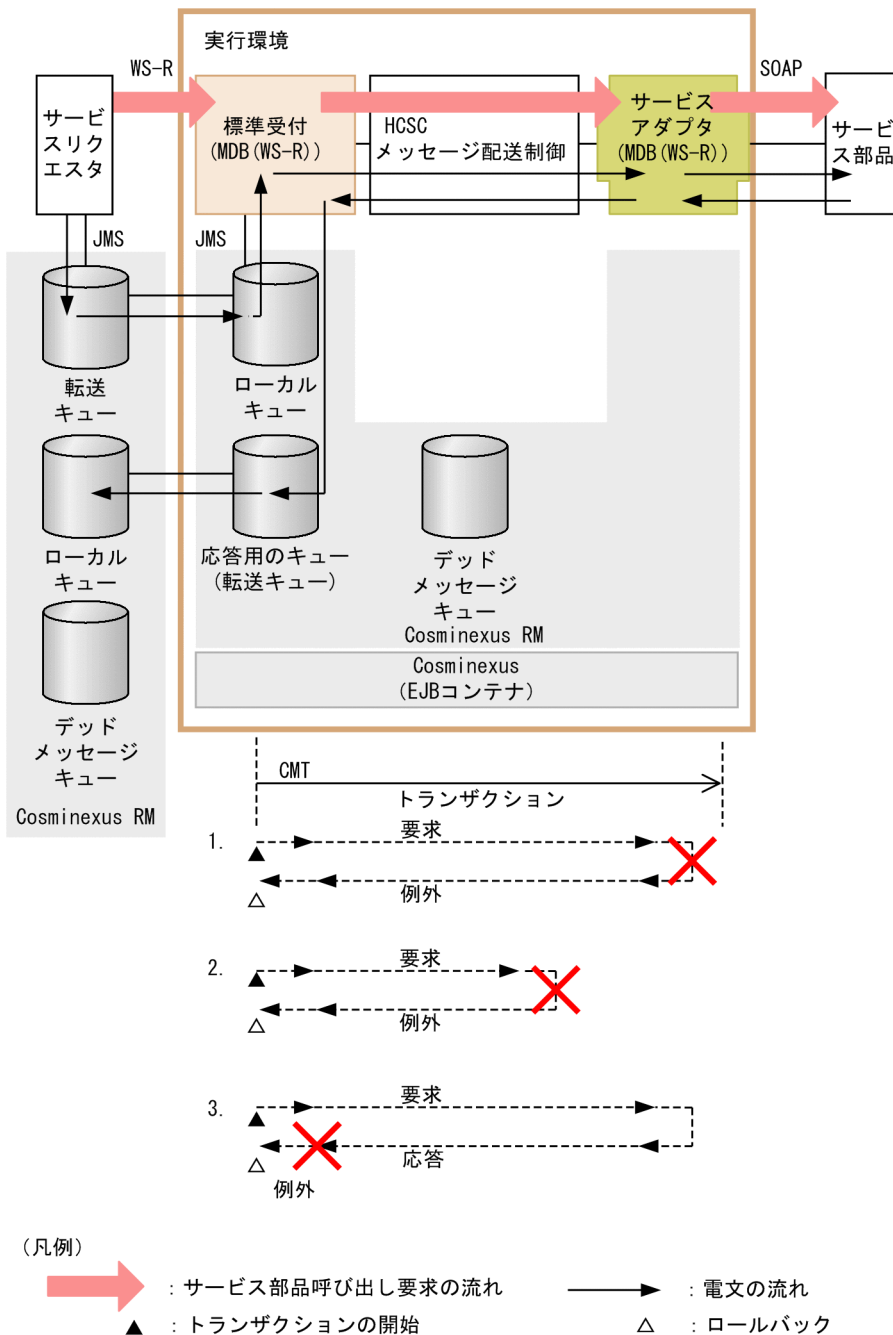
(2) 同期のサービス部品アダプタへメッセージを送信する場合

非同期 (MDB (WS-R)) の標準受付でサービス部品呼び出し要求を受け付け、同期のサービスアダプタへメッセージを送信する場合、標準受付でのローカルキューからのメッセージの取り出しからサービスアダプタでのサービス部品呼び出し、および応答が返ってきて応答用のキューへメッセージを送信するまでを、一つのトランザクションで処理します。

転送キューへのメッセージの送信が失敗した場合や、指定したサービスアダプタが起動していない場合、応答用のキューへのメッセージ送信が失敗した場合など、何らかのエラーが発生した場合は、トランザクションをロールバックします。ロールバックすると、標準受付で取り出したメッセージが復元され、再度ローカルキューからメッセージを取り出して、サービス部品を呼び出します。受付が非同期であるがサービス部品が同期の場合のトランザクションについて次の図に示します。

2. HCSC サーバのサービス部品呼び出し

図 2-30 受付が非同期であるがサービス部品が同期の場合のトランザクション



トランザクションが失敗すると、次のようになります。番号は、図中の番号と対応しています。

1. 転送キューへのメッセージの送信が失敗した場合

サービスアダプタから SOAP でのサービス部品呼び出しが失敗すると、ローカルキューにメッセージが戻り、再度標準受付でメッセージを取り出します。

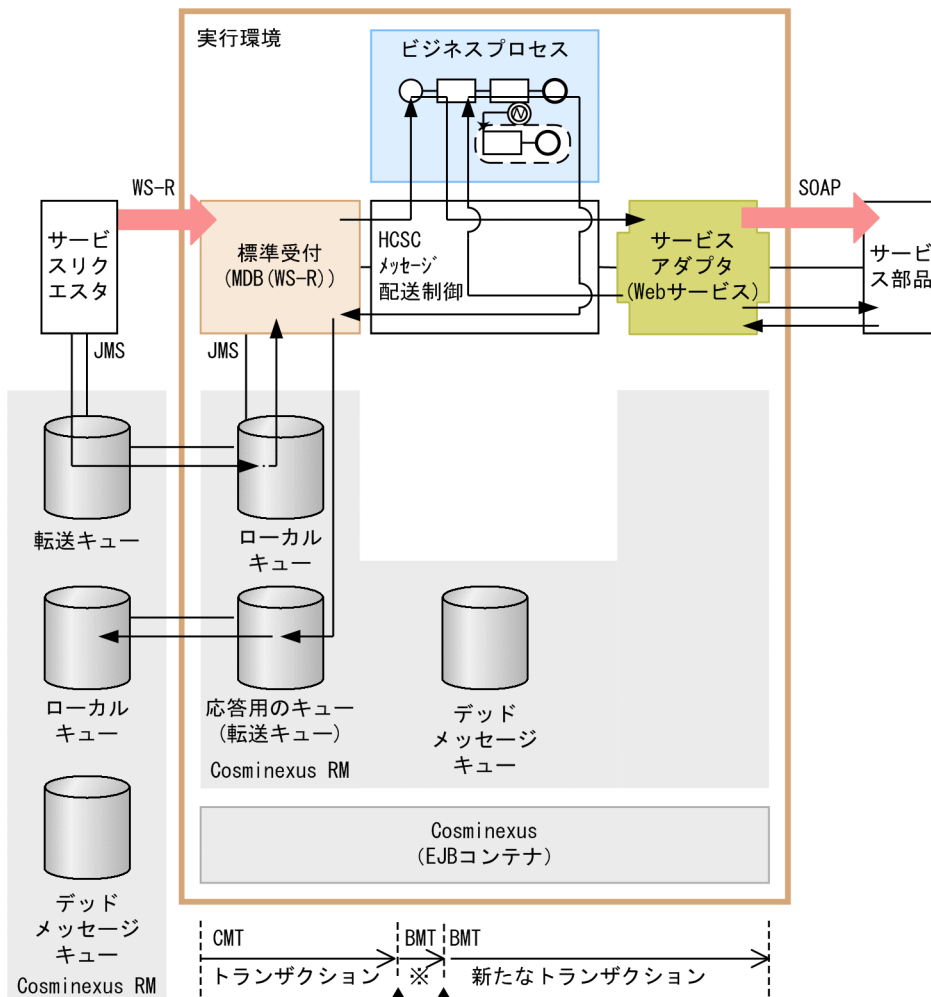
2. 指定したサービスアダプタが起動していない場合
指定したサービスアダプタが起動していないと、ローカルキューにメッセージが戻り、再度標準受付でメッセージを取り出します。
3. 応答用のキューへのメッセージ送信が失敗した場合
応答が返ってきて応答用のキューへのメッセージ送信が失敗すると、ローカルキューにメッセージが戻り、再度標準受付でメッセージを取り出します。

エラーが発生したときの対処方法については、マニュアル「Cosminexus サービスプラットフォーム システム構築・運用ガイド」の「7.7.3 MDB (WS-R) 実行時の障害対策」を参照してください。

ビジネスプロセスは BMT (Bean Managed Transaction) で動作します。ビジネスプロセス自体は新たなトランザクションで制御します。ビジネスプロセスを呼び出す場合は、ビジネスプロセスのサービス呼出アクティビティに定義したサービス部品呼び出し処理ごとに、新たなトランザクションで制御します。ビジネスプロセスを介した場合のトランザクションについて次の図に示します。

2. HCSC サーバのサービス部品呼び出し

図 2-31 ビジネスプロセスを介した場合のトランザクション



永続化するビジネスプロセスの場合、ビジネスプロセスを実行する過程でのトランザクションについては、「4.4 ビジネスプロセスのトランザクション」を参照してください。

2.4.5 デッドメッセージキュー

HCSC サーバ稼働マシンでのサービス部品の呼び出しがロールバックして、ロールバックの回数（メッセージの配送回数）が最大値に達した場合や、配送回数に無制限の設定をしている場合、そのメッセージはデッドメッセージキューに遷移します。

エラーが発生したときにデッドメッセージキューに遷移する仕組みについては、マニュアル「Cosminexus サービスプラットフォーム システム構築・運用ガイド」の「7.7.3 MDB (WS-R) 実行時の障害対策」を参照してください。

2.4.6 MDB (WS-R) で使用するキューの名称と個数

HCSC サーバの実行環境上では、次の表に示すキューを使用します。

表 2-2 MDB (WS-R) で使用するキューの名称と個数

項番	用途	説明	キューの種類	キューの名称	個数	備考
1	非同期受付 (MDB (WS-R)) 用	サービスリクエスト側の転送キューの宛先となるキュー。	ローカルキュー	<ul style="list-style-type: none"> 単独構成およびロードバランスクラスタの場合 CSC<HCSC サーバ名 >ACPT_RCVQ HA クラスタの場合 CSC<クラスタ名 >ACPT_RCVQ 	1 個	-
2	サービスアダプタ (MDB (WS-R)) 用	サービス部品側のローカルキューへ転送するキュー。	転送キュー	CSC_<クラスタ名 ><アダプタ名>	非同期 (MDB (WS-R)) サービスアダプタ数	-
3	デッドメッセージキュー	MDB (DB キュー) 用のデッドメッセージキューと共用。	ローカルキューまたは転送キュー	任意の名称	システムに 1 個	DB キューでのデッドメッセージキューと同一。
4	応答用のキュー	サービス部品呼び出しの応答を受け取る場合に使用。	ローカルキューまたは転送キュー	任意の名称	任意の個数	-

(凡例)

- : 該当しません。

2.5 MDB (DB キュー) を使ったサービス部品呼び出し

2.5.1 Cosminexus RM との関係

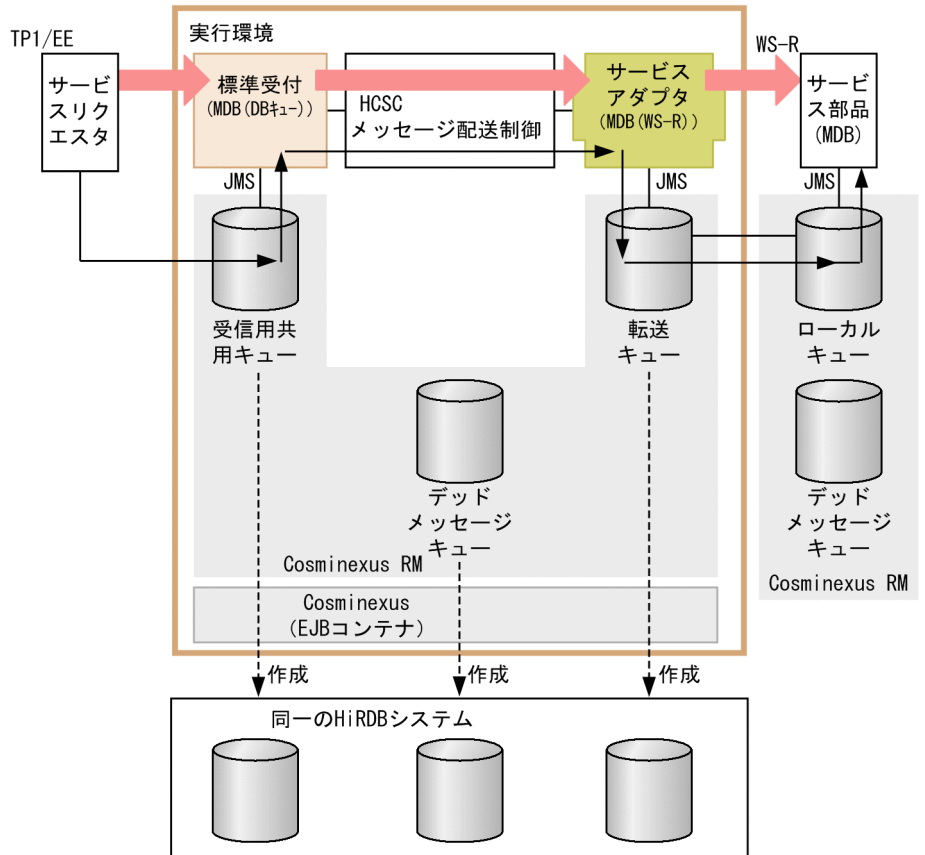
DB キューを使った非同期のサービス部品呼び出しでも、Cosminexus RM を使用します (通常は、TP1/EE と連携するために使用します。TP1/EE 側では DB キューという名前で呼びます)。

サービスリクエスタ側の TP1/EE とのインターフェースには受信用共用キューを、サービス部品側の TP1/EE とのインターフェースには送信用共用キューを使用します。

なお、非同期 (MDB (DB キュー)) の標準受付から、非同期 (MDB (DB キュー)) サービスアダプタを呼び出すことはできません。必ず異なるプロトコルのリクエスト受付またはサービスアダプタと組み合わせて使用する必要があります。Cosminexus RM との関係を次の図に示します。

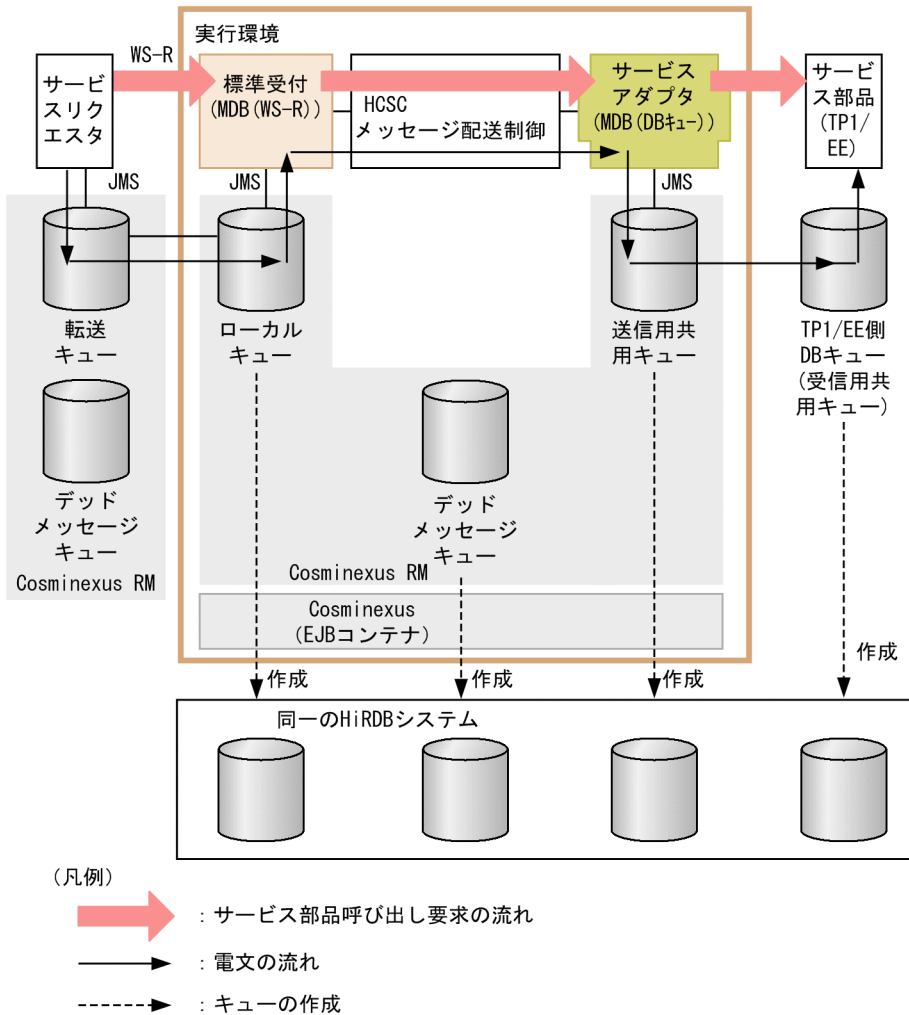
図 2-32 Cosminexus RM との関係

●サービスリクエスタ側がTP1/EEの場合



2. HCSC サーバのサービス部品呼び出し

●サービス部品側がTP1/EEの場合



! 注意事項

実行環境内とHiRDBに別々のキューがそれぞれ作成されるわけではありません。データベース上のキューは同一のシステム上に作成されるため、実行環境内のキューとHiRDBのキューは同じものになります。TP1/EE側のDBキューも同様です。

2.5.2 サービス部品呼び出し時のメッセージの流れ

HCSC サーバのリクエスト受付は、MDB (Message Driven Bean) で動作し、HCSC サーバ側の受信用共用キューを監視しています。HCSC サーバ側の受信用共用キューにメッセージが送信されると、メッセージを取り出してサービス部品を呼び出します。

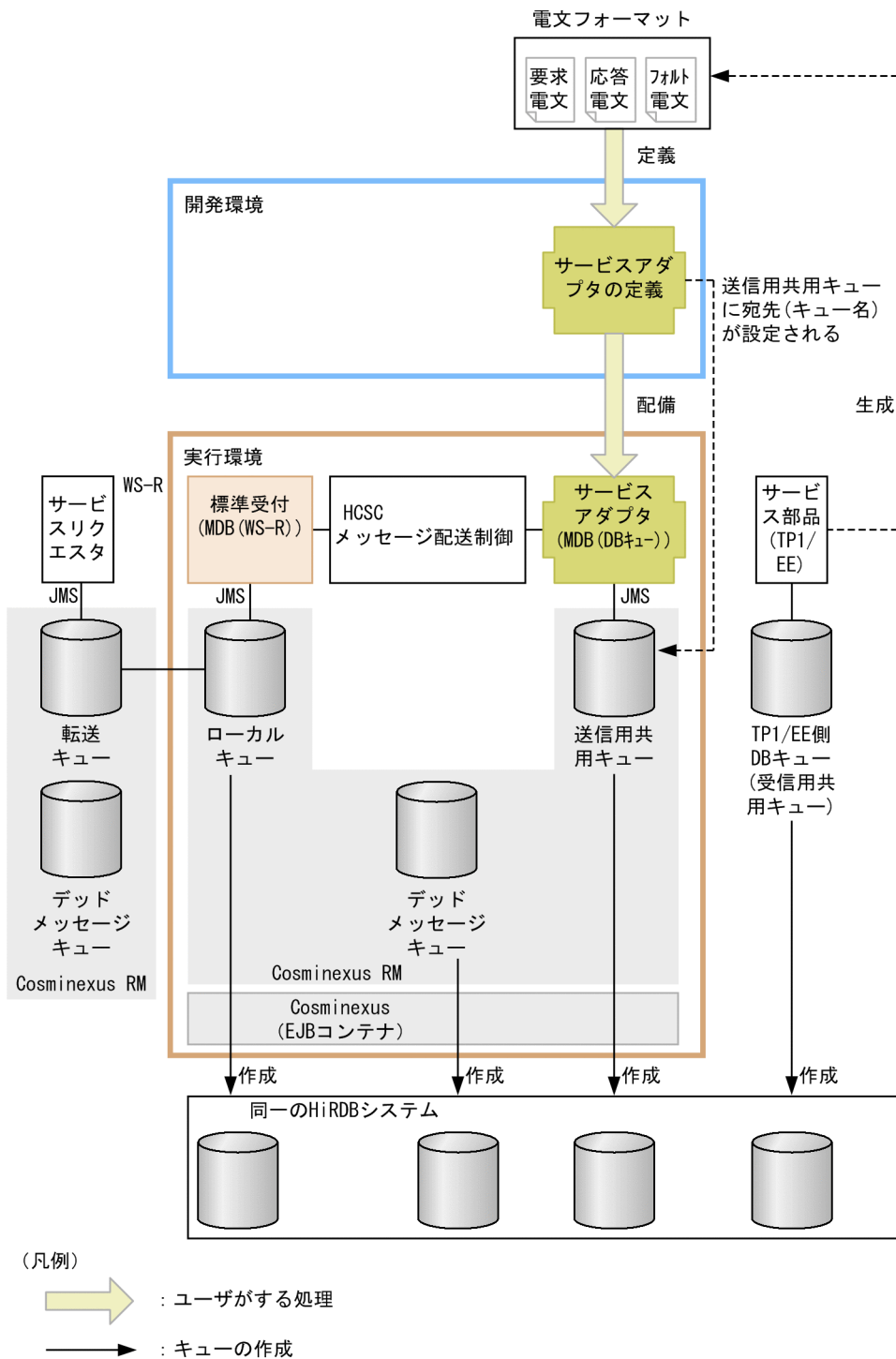
TP1/EE 側のサービスリクエストからは、受信用共用キューがあるデータベースに対して、直接アクセスしてメッセージを送信します。

サービスリクエスト側の送信用共用キューには、宛先となるキュー名を設定します（開発環境のサービスアダプタ定義画面で設定します）。

宛先となる TP1/EE 側の DB キュー（受信用共用キュー）は、TP1/EE のコマンドを使用して作成します。なお、TP1/EE 側の DB キューは、送信用共用キューがある同一データベース内（同一の HiRDB システム内）に作成する必要があります。非同期（MDB（DB キュー））の場合の宛先の指定について次の図に示します。

2. HCSC サーバのサービス部品呼び出し

図 2-33 非同期 (MDB (DB キュー)) の場合の宛先の指定



! 注意事項

実行環境内と HiRDB に別々のキューがそれぞれ作成されるわけではありません。データベース上のキューは同一のシステム上に作成されるため、実行環境内のキューと HiRDB のキューは同じものになります。TP1/EE 側の DB キューも同様です。

DB キューの作成の詳細については、マニュアル「TP1/Server Base Enterprise Option 使用の手引」の DB キューに関する内容を参照してください。

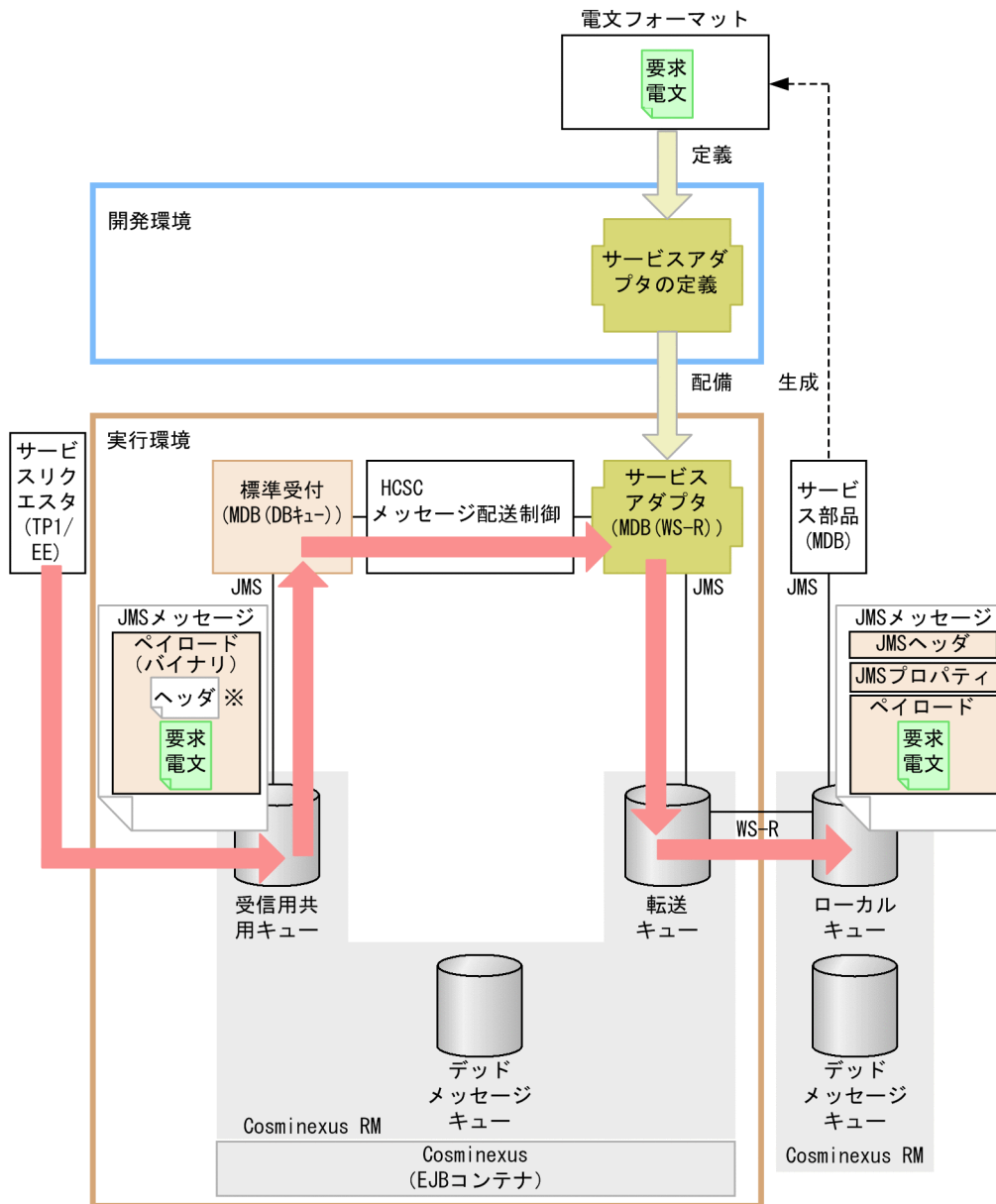
(1) TP1/EE からサービス部品を呼び出す場合

サービスリクエストからサービス部品を呼び出すときに、受信用共用キューに送信するメッセージの形式は、BytesMessage です。

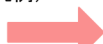
呼び出すサービス名やクライアント関連 ID などのパラメタを設定した HCSC サーバ用 DB キューヘッダをバイナリ形式で作成し、そのあとに要求電文 (バイナリ形式) を組み立てます。これによって、要求電文 (バイナリ形式) の内容が、ユーザ電文としてサービス部品に伝わります。サービスリクエストから標準受付を呼び出すとき、サービス部品側の電文フォーマットに合わせた要求電文を事前に作成し、ペイロードに設定してメッセージを送信します。TP1/EE からサービス部品を呼び出すときのユーザ電文と電文フォーマットの関係を示す図に示します。

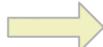
2. HCSC サーバのサービス部品呼び出し

図 2-34 TP1/EE からサービス部品を呼び出すときのユーザ電文と電文フォーマットの関係



(凡例)

 : サービス部品呼び出し要求の流れ

 : ユーザがする処理

注※ HCSCサーバ用DBキューヘッダです。

バイナリデータを作成する方法については、マニュアル「Cosminexus サービスプラットフォーム 開発ガイド」の「10.6.4 バイナリデータを作成する (TP1/EE, JMS)」を参照してください。

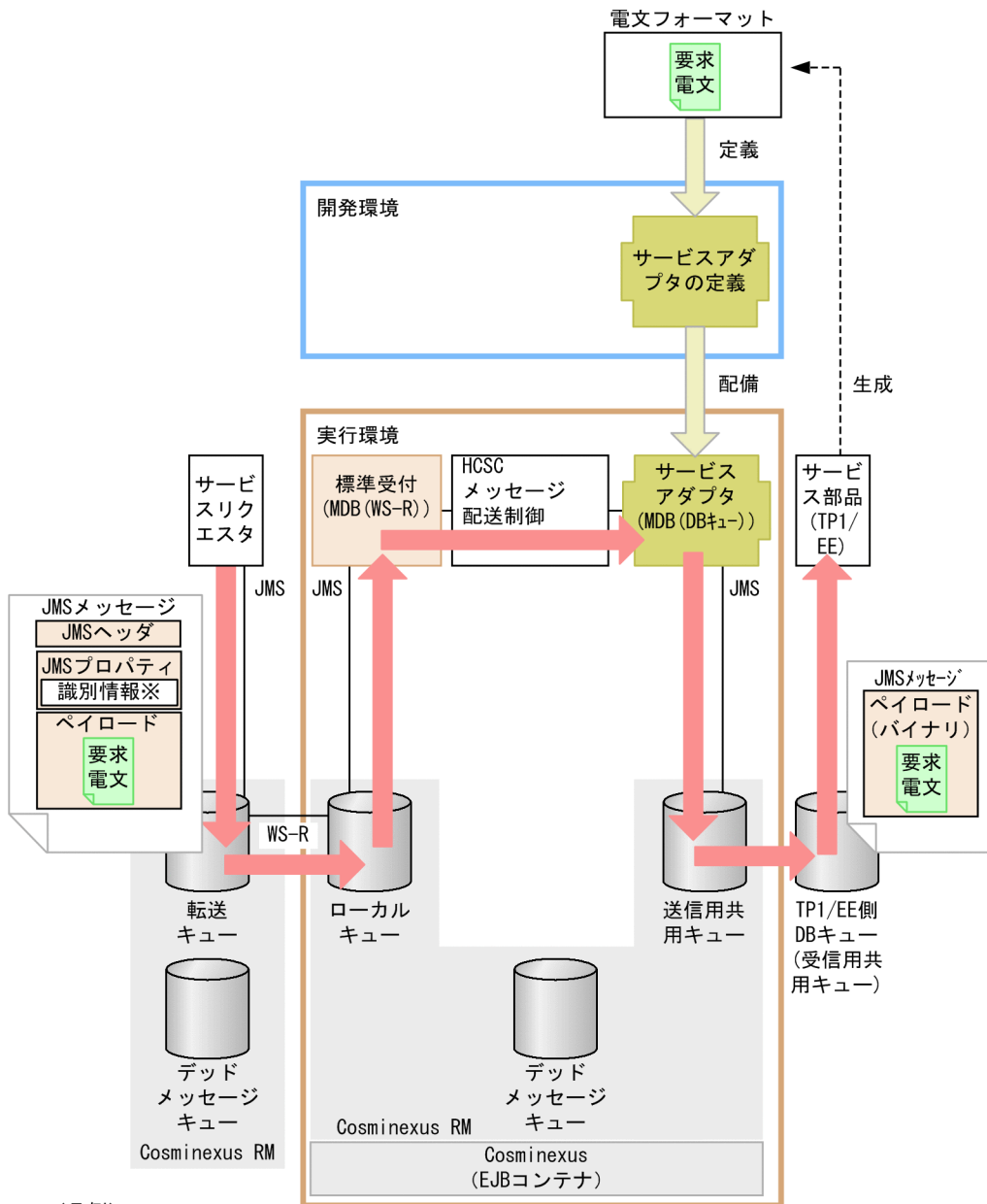
(2) TP1/EE のサービス部品を呼び出す場合

サービスアダプタからサービス部品を呼び出すときに送信用共用キューに送信するメッセージの形式は、`ByteMessage` です。

サービスアダプタに定義した電文フォーマットに合わせた要求電文をバイナリ電文としてメッセージに設定します。TP1/EE のサービス部品を呼び出すときのユーザ電文と電文フォーマットの関係は次の図に示します。

2. HCSC サーバのサービス部品呼び出し

図 2-35 TP1/EE のサービス部品を呼び出すときのユーザ電文と電文フォーマットの関係



(凡例)

- : サービス部品呼び出し要求の流れ
- : ユーザがする処理

注※ 識別情報とは、サービス名、クライアント相関ID、およびユーザ定義プロパティです。

2.5.3 同期のサービス部品を呼び出す場合の応答電文の扱い

標準の非同期受付 (MDB (DB キュー)) は、非同期の要求電文の標準受付で、基本的にサービス部品からの応答がない場合に使用します。しかし、非同期受付 (MDB (DB キュー)) から同期のサービス部品 (Web サービスまたは SessionBean など) やビジネスプロセスを要求した場合、同期のサービス部品から応答が返るときがあります。

そのため、HCSC サーバでは応答を応答用のキューに送信する手段を提供しています。

応答用のキューを利用するには、HCSC サーバの環境構築時に HCSC サーバ側に送信用共用キューを作成し、送信用共用キューに送信先のキュー名を指定しておきます。

サービスリクエストから標準受付に対してサービス部品の呼び出しを要求する場合、標準受付のパラメタに作成した送信用共用キューのキュー名を指定すると、サービス部品からの応答が返ってきた際に指定されたキューに JMS メッセージが送信されます。

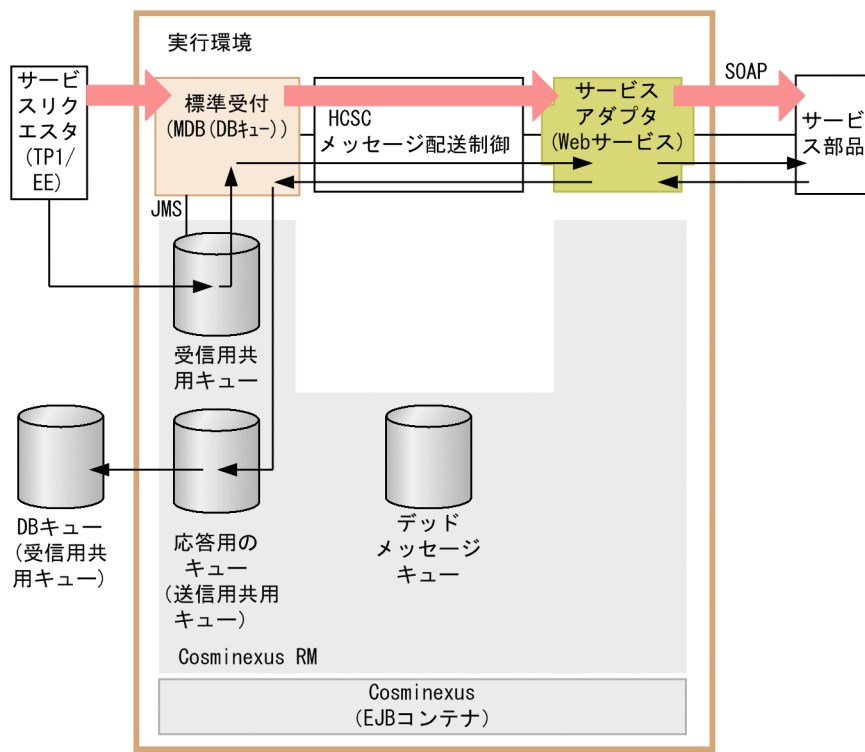
送信用共用キューからサービスリクエスト側の DB キュー (受信用共用キュー) にメッセージが転送されるため、DB キュー (受信用共用キュー) からメッセージを取り出すことで、サービス部品からの応答電文を取得できます。

応答を取り出す方法については、マニュアル「Cosminexus サービスプラットフォーム開発ガイド」の「10.6.10 応答を取り出す (TP1/EE, JMS)」を参照してください。

同期サービス部品を呼び出した場合の応答の扱いを次の図に示します。

2. HCSC サーバのサービス部品呼び出し

図 2-36 同期サービス部品を呼び出した場合の応答の扱い



(凡例)

- ➡ : サービス部品呼び出し要求の流れ
- ➡ : 電文の流れ

応答用のキューへメッセージを出力するのは、次に示す場合です。

同期サービス部品でも応答があるかどうかによって、次のような形式で応答用のキューへメッセージを送信します。

なお、サービスリクエストから応答用のキューのキュー名を指定していても、HCSC サーバ側に指定したキュー名のキューが作成されていない場合は、応答用のキューにメッセージは送信できず、サービス部品呼び出し処理がエラーになります。

表 2-3 応答用のキューへメッセージを出力するケース

項番	応答用のキューへ出力するケース	応答電文の有無			
		応答電文がある場合	応答電文がない場合	サービス部品からのユーザ定義例外の場合	ユーザ定義例外以外のエラーの場合
1	呼び出し先のサービス部品が同期サービス部品の場合 (WebService, SessionBean およびカスタムアダプタなど)	ペイロードに応答電文を設定	ペイロードなしのメッセージ	ペイロードに応答電文を設定	x
2	呼び出し先のサービス部品が非同期サービス部品の場合 (MDB (WS-R) および MDB (DB キュー) など)	x	x	-	x
3	呼び出し先のサービス部品がビジネスプロセスの場合	ペイロードに応答電文を設定	ペイロードなしのメッセージ	ペイロードに応答電文を設定	x

(凡例)

x : メッセージを送信しません。

- : 非同期のサービス部品からユーザ定義例外は存在しないため該当しません。

注 エラーの情報はログおよびトレースに出力します。

サービスリクエストからの要求で応答用のキューのキュー名を指定していない場合、および電文の実行履歴を採取している場合は、応答電文の内容は破棄されます (応答電文を破棄したことを示す KDEC00031-W メッセージが出力されます)。

なお、MDB (WS-R) の場合のように、JMS プロパティを引き継ぐことはできません。サービス部品呼び出しの相関関係に対応づける場合は、ユーザ電文内に相関関係を示す情報を埋め込みます。

2.5.4 MDB (DB キュー) 使用時のトランザクション

MDB (DB キュー) の場合、サービスリクエストから HCSC サーバの間、そしてサービスアダプタからサービス部品の間は、異なるトランザクションになります。

標準受付でのローカルキューからのメッセージの取り出しとサービスアダプタでの転送キューへの送信は、一つのトランザクションで処理します。MDB (WS-R) の場合のトランザクションと同一です。

永続化するビジネスプロセスの場合、ビジネスプロセスを実行する過程でのトランザクションについては、「4.4 ビジネスプロセスのトランザクション」を参照してください。

2.5.5 デッドメッセージキュー

HCSC サーバ稼働マシンでのサービス部品の呼び出しをロールバックして、ロールバックの回数（メッセージの配送回数）が最大値に達した場合や、配送回数に無制限の設定をしている場合、そのメッセージはデッドメッセージキューに遷移します。

エラーが発生したときにデッドメッセージキューに遷移する仕組みについては、マニュアル「Cosminexus サービスプラットフォーム システム構築・運用ガイド」の「7.7.3 MDB (WS-R) 実行時の障害対策」を参照してください。

2.5.6 MDB (DB キュー) で使用するキューの名称と個数

HCSC サーバの実行環境上では、次の表に示すキューの名称と個数を使用します。データベース (DB キュー) で使用するキューの名称と個数を次の表に示します。

表 2-4 データベース (DB キュー) で使用するキューの名称と個数

項番	用途	説明	キューの種類	キューの名称	個数	備考
1	非同期受付 (MDB (DB キュー)) 用	サービスリクエスト側の TP1/EE システムから要求を受信するキュー。	受信用共用キュー	<ul style="list-style-type: none"> 単独構成およびロードバランスクラスタの場合 CSC<HCSC サーバ名 >ACPT_DBQ HA クラスタの場合 CSC< クラスタ名 >ACPT_DBQ 	1 個	-
2	サービスアダプタ (MDB (DB キュー)) 用	サービス部品側の DB キューへ転送するキュー。	送信用共用キュー	CSC_< クラスタ名 >< アダプタ名 >	非同期 (MDB (DB キュー)) サービスアダプタ数	-
3	デッドメッセージキュー	MDB (WS-R) 用のデッドメッセージキューと共用。	ローカルキューまたは転送キュー	任意の名称	システムに 1 個	WS-R でのデッドメッセージキューと同一
4	応答用のキュー	サービス部品呼び出しの応答を受け取る場合に使用。	送信用共用キュー	任意の名称	任意の個数	-

(凡例)

- : 該当しません。

3

標準プロトコル以外のシステムとの接続

この章では、標準プロトコル（Web サービス、SessionBean、MDB（WS-R）、および MDB（DB キュー））以外のシステムと接続した場合の、サービス部品の呼び出しについて説明します。

3.1 データベースとの接続

3.2 OpenTP1 との接続

3.3 Object Wrapper システムとの接続

3.4 他社 ERP パッケージとの接続

3.5 ファイルを扱うシステムとの接続

3.6 メッセージキューとの接続

3.7 FTP クライアントおよび FTP サーバとの接続

3.8 カスタム受付を使用した場合のシステムとの接続

3.9 カスタムアダプタを使用した場合のシステムとの接続

3.1 データベースとの接続

サービス部品としてデータベースを呼び出す場合には、DB アダプタを使用します。DB アダプタを使用する場合のデータベースのアクセスと設定について説明します。

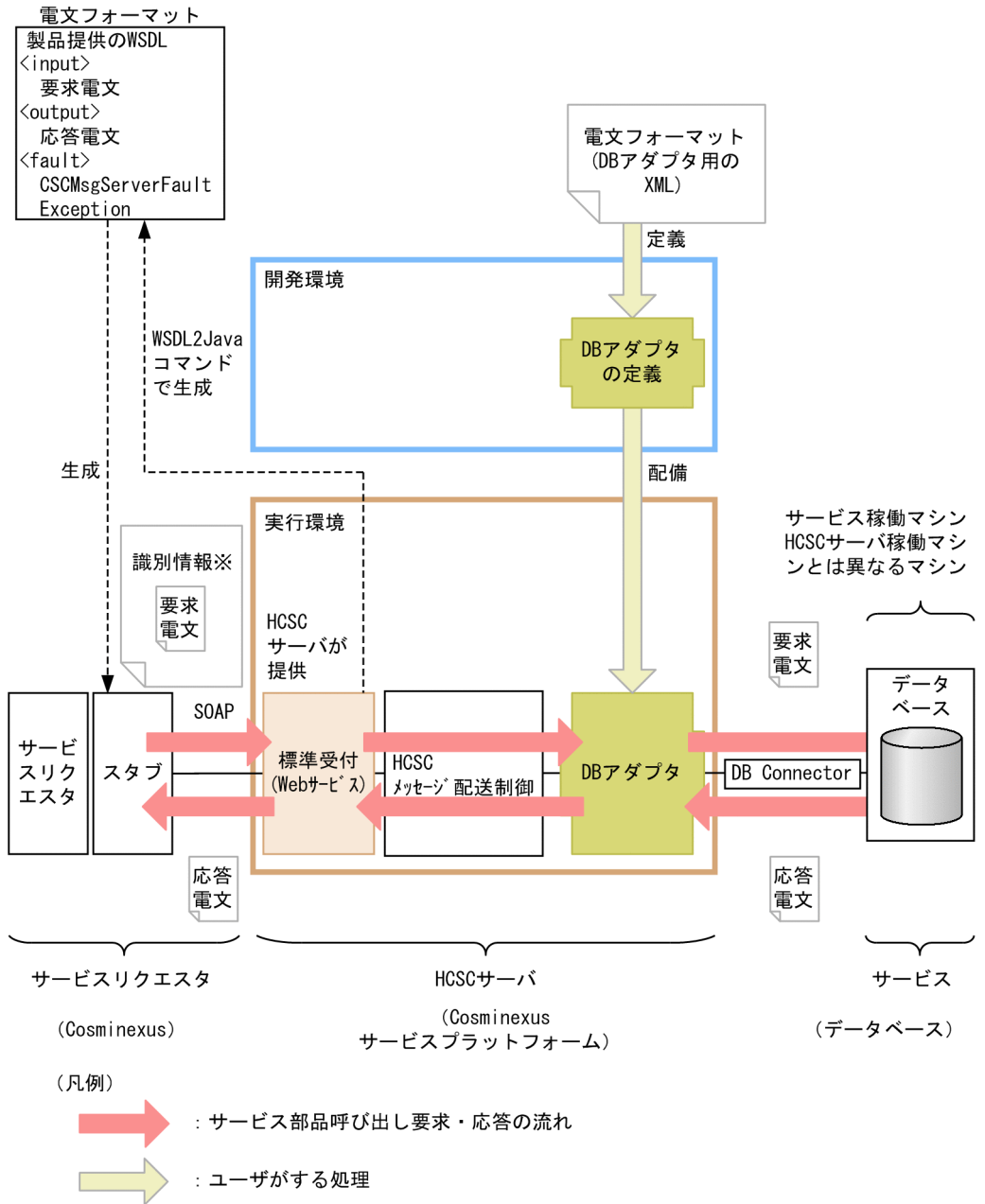
3.1.1 DB アダプタを使ったデータベースへのアクセス

DB アダプタは、サービスリクエストやビジネスプロセスからの要求を受けて、データベースに対して直接アクセスできます。データベースに対するアクセスは、DB アダプタ内で定義されている SQL に従います。

(1) DB アダプタによる SQL の実行

DB アダプタは、開発環境で定義します。DB アダプタに対して要求するサービスリクエストからは、実行する SQL 文を記述した XML 文書を作成して、作成した XML 電文を要求電文に設定して実行します。DB アダプタによる SQL の実行について次の図に示します。

図 3-1 DB アダプタによる SQL の実行



注※ 識別情報とは、サービス名、クライアント相関IDなどです。

DB アダプタを定義する手順については、マニュアル「Cosminexus サービスプラットフォーム 開発ガイド」の「5.3.5 DB アダプタを定義する」を参照してください。DB アダプタに要求を出すサービスリクエスタの作成については、マニュアル「Cosminexus

3. 標準プロトコル以外のシステムとの接続

サービスプラットフォーム 開発ガイド」の「10.8 DB アダプタに要求を出すサービスリクエスト」を参照してください。

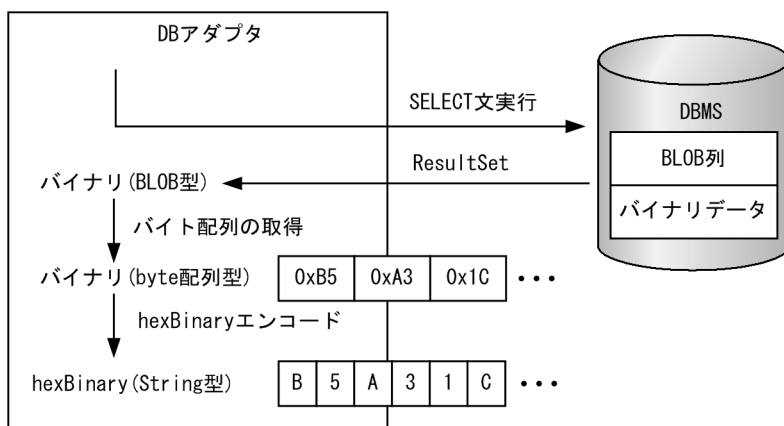
(2) バイナリデータ操作時の処理

DB アダプタ内部で、バイナリデータは byte 配列型、hexBinary 型は String 型として扱います。SELECT 文実行時と INSERT 文実行時に BLOB 型を操作する場合の例を次に説明します。

(a) SELECT 文実行時の処理

BLOB 列に対して SELECT 文を実行すると、DB アダプタでは byte 配列から XML Schema の hexBinary 型に変換（エンコード）して、要求元に BLOB 列の値を返却します。

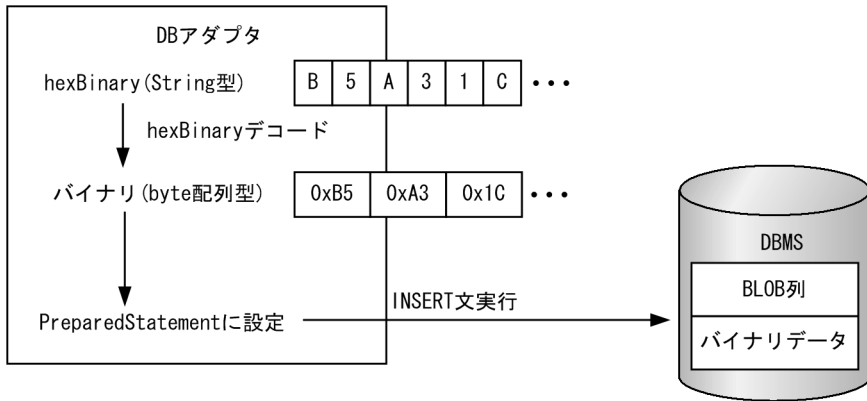
図 3-2 SELECT 文実行時の処理



(b) INSERT 文実行時の処理

BLOB 列に対して INSERT 文を実行する際は、要求元が XML Schema の hexBinary 型のデータを DB アダプタに渡します。DB アダプタでは hexBinary 型から byte 配列に変換（デコード）して、INSERT 文を実行します。

図 3-3 INSERT 文実行時の処理



3.1.2 DB アダプタが使用する DB Connector の設定

データベースにアクセスするには、HCSC サーバをセットアップするときに定義した DB Connector とは別に、サービス部品稼働マシン（データベースが稼働しているマシン）に対してアクセスする DB Connector を準備する必要があります。

(1) 使用できる DB Connector

HCSC サーバをセットアップするときに定義した DB Connector のトランザクション属性によって、使用できる DB Connector に制約があります。

DB アダプタで使用できる DB Connector は、HCSC サーバセットアップ定義ファイルの `dbcon-xadisplayname` プロパティに指定した XATransaction または LocalTransaction の値に依存します。そのため、HCSC サーバのセットアップ時には、`dbcon-xadisplayname` プロパティに指定した XATransaction または LocalTransaction の値を考慮する必要があります。

表 3-1 使用可能な DB Connector

使用している DB Connector	DB アダプタ		使用可否	否の理由
	使用可能な DB Connector	DB アダプタの接続先		
LocalTransaction	LocalTransaction	HCSC サーバが使用するデータベースと同じデータベース	可	-

3. 標準プロトコル以外のシステムとの接続

使用している DB Connector (dbcon-xadisplayname)	DB アダプタ		使用可否	否の理由
	使用可能な DB Connector	DB アダプタの接続先		
HCSC サーバ本体側 (dbcon-xadisplayname)	LocalTransaction	HCSC サーバが使用するデータベースと異なるデータベース	否	Local トランザクション内で複数の DB Connector を混在させることができないため
	XATransaction	HCSC サーバが使用するデータベースと異なるデータベース	否	Local トランザクション内で XA (グローバルトランザクション) を混在させることができないため
XATransaction	LocalTransaction	HCSC サーバが使用するデータベースと異なるデータベース	否	XA トランザクション内で複数の DB Connector を混在させることができないため
	XATransaction	HCSC サーバが使用するデータベースと同じデータベース	可	-
	XATransaction	HCSC サーバが使用するデータベースと異なるデータベース	可	-

(凡例) - : 該当しません。

(2) DB Connector のインポートおよび配備

DB アダプタが利用する DB Connector を HCSC サーバ (J2EE サーバ) にインポートし、配備します。

DB アダプタを利用する場合にインポートする RAR ファイルの選択方法については、マニュアル「Cosminexus サービスプラットフォーム システム構築・運用ガイド」の「3.1.2 実行環境に必要なソフトウェアの設定」を参照してください。

1. DB Connector のインポート

次に示すコマンドを使用して DB Connector をインポートします。

```
<Cosminexusインストールディレクトリ>%CC%admin%bin%cjimportres <J2EEサーバ名>
-nameserver corbaname::localhost:<インプロセスのネーミングサービス用のポート番号>
-type rar -f <Cosminexusインストールディレクトリ>%CC%DBConnector%<インポートするRARファイル>
```

2. DB Connector の配備

次に示すコマンドを使用して DB Connector を配備します。

```
<Cosminexusインストールディレクトリ>%CC%\admin\bin%cjdeployrar <J2EEサーバ名>
-nameserver corbaname::localhost:<インプロセスのネーミングサービス用のポート番号>
-resname <インポートしたDB Connectorの名称>
```

(3) DB Connector のプロパティ定義

DB Connector をインポートしたあと、プロパティを定義します。プロパティ定義ファイルの詳細については、マニュアル「Cosminexus アプリケーションサーバリファレンス 定義編（アプリケーション/リソース定義）」の「4.1 Connector 属性ファイル」を参照してください。

参考

配備した DB Connector に対してプロパティを設定するときに、新規にプロパティ定義ファイルの XML を作成するのではなく、配備した DB Connector のプロパティを取得し、取得した XML に対して編集をすると効率良く作成できます。

1. プロパティ定義用のファイルの取得

次に示すコマンドを使用して DB Connector のプロパティ定義用のファイルを取得します。

```
<Cosminexusインストールディレクトリ>%CC%\admin\bin%cjgetrarprop <J2EEサーバ名>
-nameserver corbaname::localhost:<インプロセスのネーミングサービス用のポート番号>
-resname <インポートしたDB Connectorの名称> -c property.xml
```

2. プロパティ定義用のファイルの編集

取得したプロパティ定義用の XML ファイルについて、各属性の値を変更します。

プロパティ定義用の XML ファイル（Connector 属性ファイル）の設定内容については、マニュアル「Cosminexus サービスプラットフォーム システム構築・運用ガイド」の「3.1.2 実行環境に必要なソフトウェアの設定」を参照してください。

Connector 属性ファイルの詳細については、マニュアル「Cosminexus アプリケーションサーバ アプリケーション設定操作ガイド」の「4.2.2 DB Connector のプロパティ定義」、およびマニュアル「Cosminexus アプリケーションサーバリファレンス 定義編（アプリケーション/リソース定義）」の「4.1 Connector 属性ファイル」を参照してください。

3. 編集したプロパティ定義用のファイルの設定

編集したプロパティ定義用の XML ファイルの内容を配備している DB Connector に設定します。

3. 標準プロトコル以外のシステムとの接続

次に示すコマンドを使用して DB Connector に定義内容を設定します。

```
<Cosminexusインストールディレクトリ>%CC%admin%bin%cjsetrarprop <J2EEサーバ名>  
-nameserver corbaname::localhost:<インプロセスのネーミングサービス用のポート番号>  
-resname <インポートしたDB Connectorの名称> -c property.xml
```

注 プロパティ定義用のファイル (property.xml) の編集後、各属性の値は上書きされます。

3.2 OpenTP1 との接続

OpenTP1 などの既存システムのサービス部品を呼び出したり、OpenTP1 などの既存システムからサービス部品を呼び出したりできます。呼び出し方で接続のしかたが異なります。

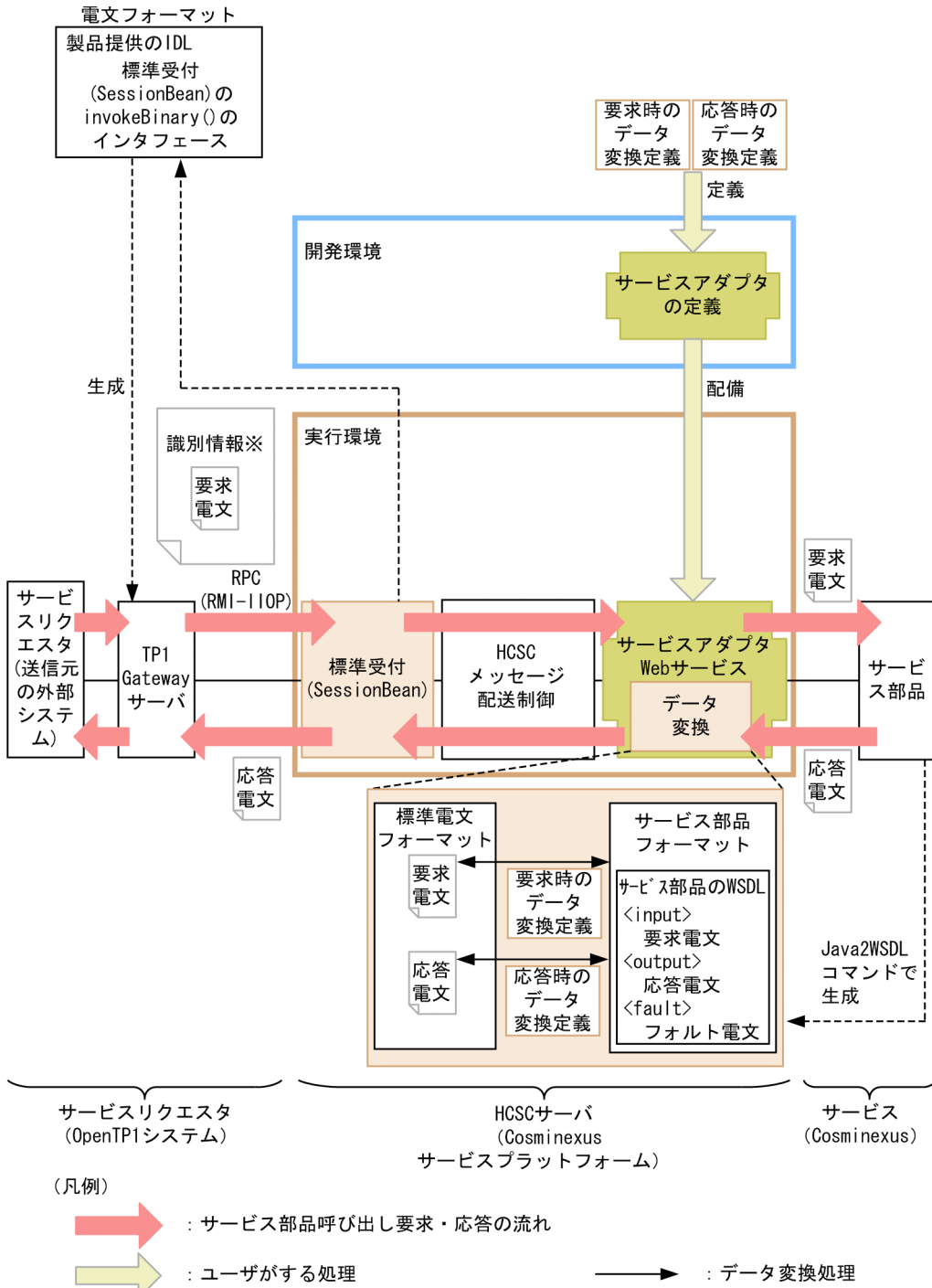
3.2.1 TP1 から同期の標準受付を使った HCSC サーバのサービス部品の呼び出し

Cosminexus サービスプラットフォームの実行環境で、OpenTP1 などの既存システムから、HCSC サーバを経由してサービス部品を呼び出す場合に uCosminexus TP1 Gateway を利用します。

uCosminexus TP1 Gateway では、HCSC サーバの標準受付 (SessionBean) に合わせて IDL を作成して、これを基にスタブを作成します。これによって、同期応答型の送信ができます。OpenTP1 システムからのサービス部品の呼び出しを次の図に示します。

3. 標準プロトコル以外のシステムとの接続

図 3-4 OpenTP1 システムからのサービス部品の呼び出し



注※ 識別情報とは、サービス名、クライアント相関IDなどです。

詳細は、マニュアル「uCosminexus TP1 Gateway ユーザーズガイド」を参照してください。

3.2.2 TP1 から非同期の標準受付を使った HCSC サーバのサービス部品の呼び出し

同期応答型の要求のほかに、非同期で OpenTP1 などの既存システムから、HCSC サーバを経由してサービス部品を呼び出せます。

HCSC サーバの標準受付 (MDB (DB キュー)) に対して要求する非同期応答型の送信ができます。

詳細は、「2.5 MDB (DB キュー) を使ったサービス部品呼び出し」、およびマニュアル「uCosminexus TP1 Gateway ユーザーズガイド」を参照してください。

3.2.3 TP1 から TP1/RPC 受付を使った HCSC サーバのサービス部品の呼び出し

HCSC サーバの標準受付 (SessionBean, MDB (DB キュー)) のほかに、ユーザ定義受付 (TP1/RPC 受付) を使用することで、既存の OpenTP1 システムから、HCSC サーバを経由してサービス部品を呼び出せます。

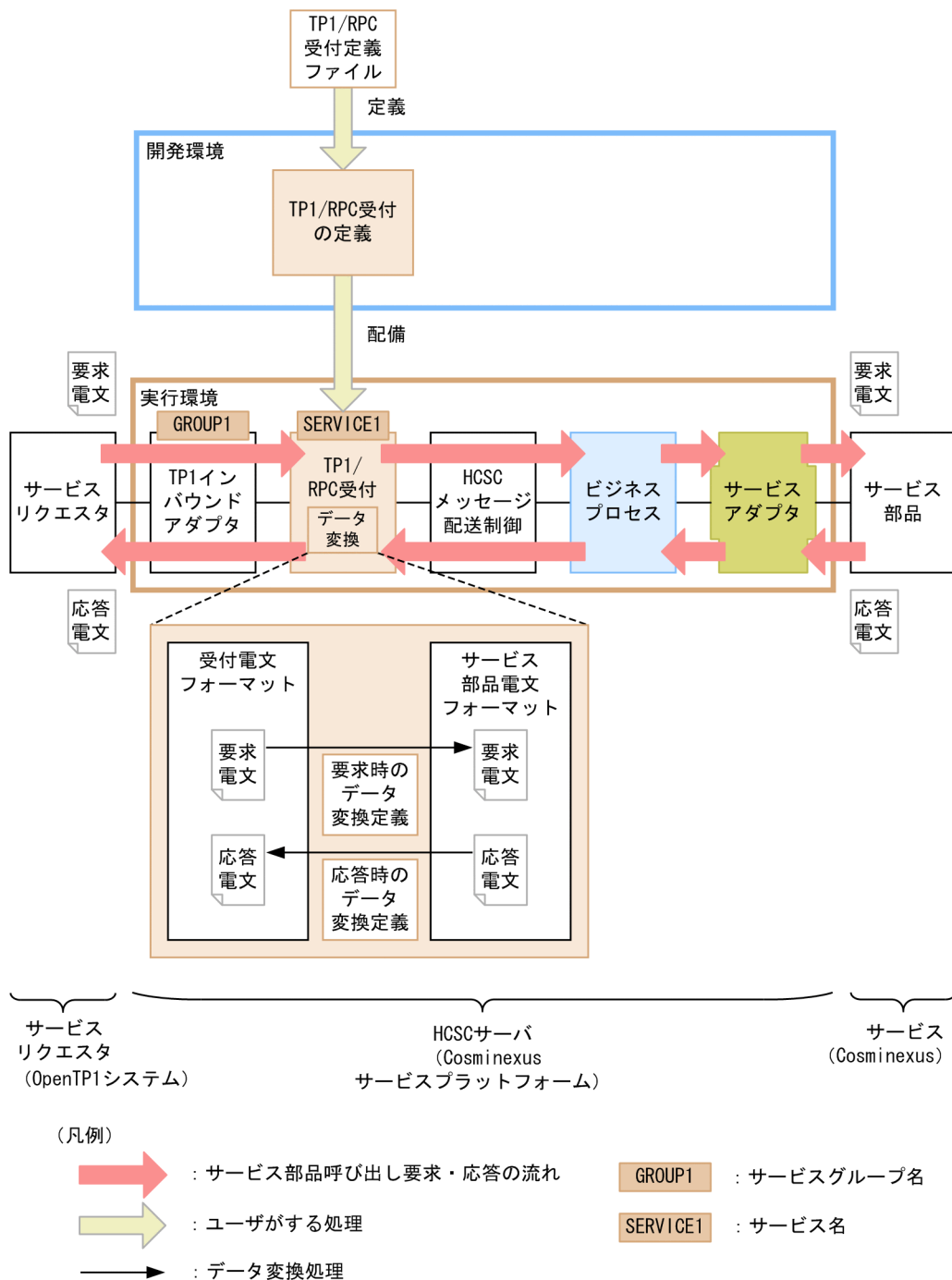
TP1/RPC 受付は、開発環境で TP1/RPC 受付定義ファイルを作成し、実行環境へ配備することで利用できます。なお、サービスリクエストからのサービス部品呼び出し要求を受けた場合、TP1/RPC 受付は、リソースアダプタである TP1 インバウンドアダプタから電文を受け取って処理をします。そのため、TP1/RPC 受付を使用する場合は、TP1 インバウンドアダプタのセットアップが必要になります。

TP1/RPC 受付を使用することで、サービスリクエストはビジネスプロセスを、リモートプロシジャコール通信 (同期応答型 RPC) で呼び出せます。サービスリクエストが呼び出すビジネスプロセスは、`dc_rpc_call` 関数に指定するサービスグループ名とサービス名で特定されます。このとき指定したサービスグループ名とサービス名が、それぞれ TP1 インバウンドアダプタと TP1/RPC 受付に結び付けられます。

OpenTP1 システムからのサービス部品の呼び出しを次の図に示します。

3. 標準プロトコル以外のシステムとの接続

図 3-5 TP1/RPC 受付を使用した OpenTP1 システムからのサービス部品の呼び出し

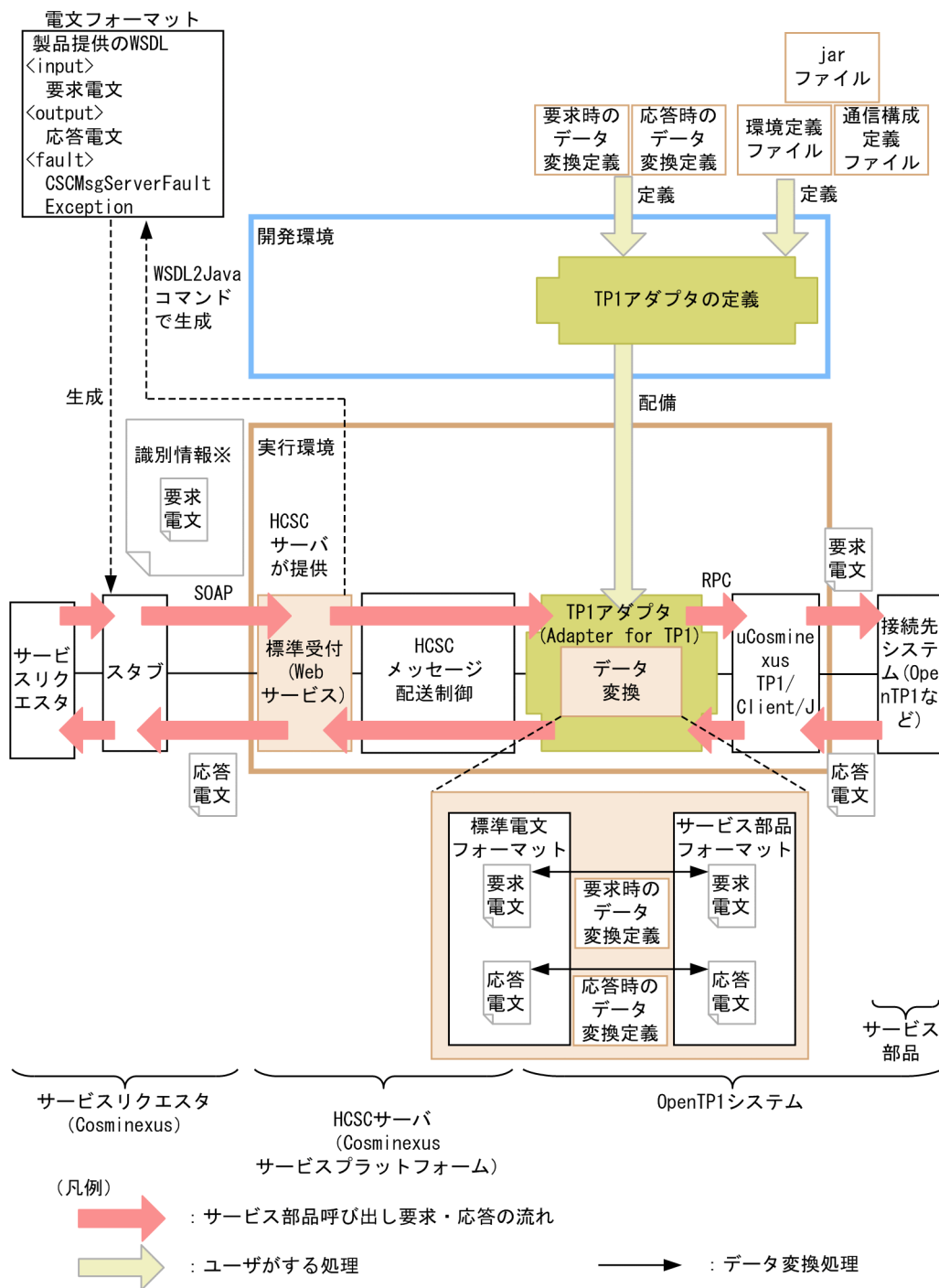


3.2.4 HCSC サーバから TP1 アダプタを使った TP1 サービス部品の呼び出し

TP1 アダプタを使用することで、OpenTP1 を使用したシステムのサービス部品を呼び出せます。TP1 アダプタを運用するには、開発環境で設定して、カスタムアダプタパッケージファイルを作成（パッケージング）し、実行環境へ配備することで利用できます。HCSC サーバからの OpenTP1 システムのサービス部品の呼び出しを次の図に示します。

3. 標準プロトコル以外のシステムとの接続

図 3-6 HCSC サーバからの OpenTP1 システムのサービス部品の呼び出し



開発環境で TP1 アダプタを作成する手順については、マニュアル「Cosminexus サービスプラットフォーム TP1 アダプタ ユーザーズガイド」を参照してください。

開発環境での操作については、マニュアル「Cosminexus サービスプラットフォーム 開発ガイド」を参照してください。

実行環境への配備方法については、マニュアル「Cosminexus サービスプラットフォーム システム構築・運用ガイド」の「3.1.8 サービスアダプタを配備する」を参照してください。

3.2.5 HCSC サーバからの MDB (DB キュー) サービスアダプタを使った TP1 サービス部品の呼び出し

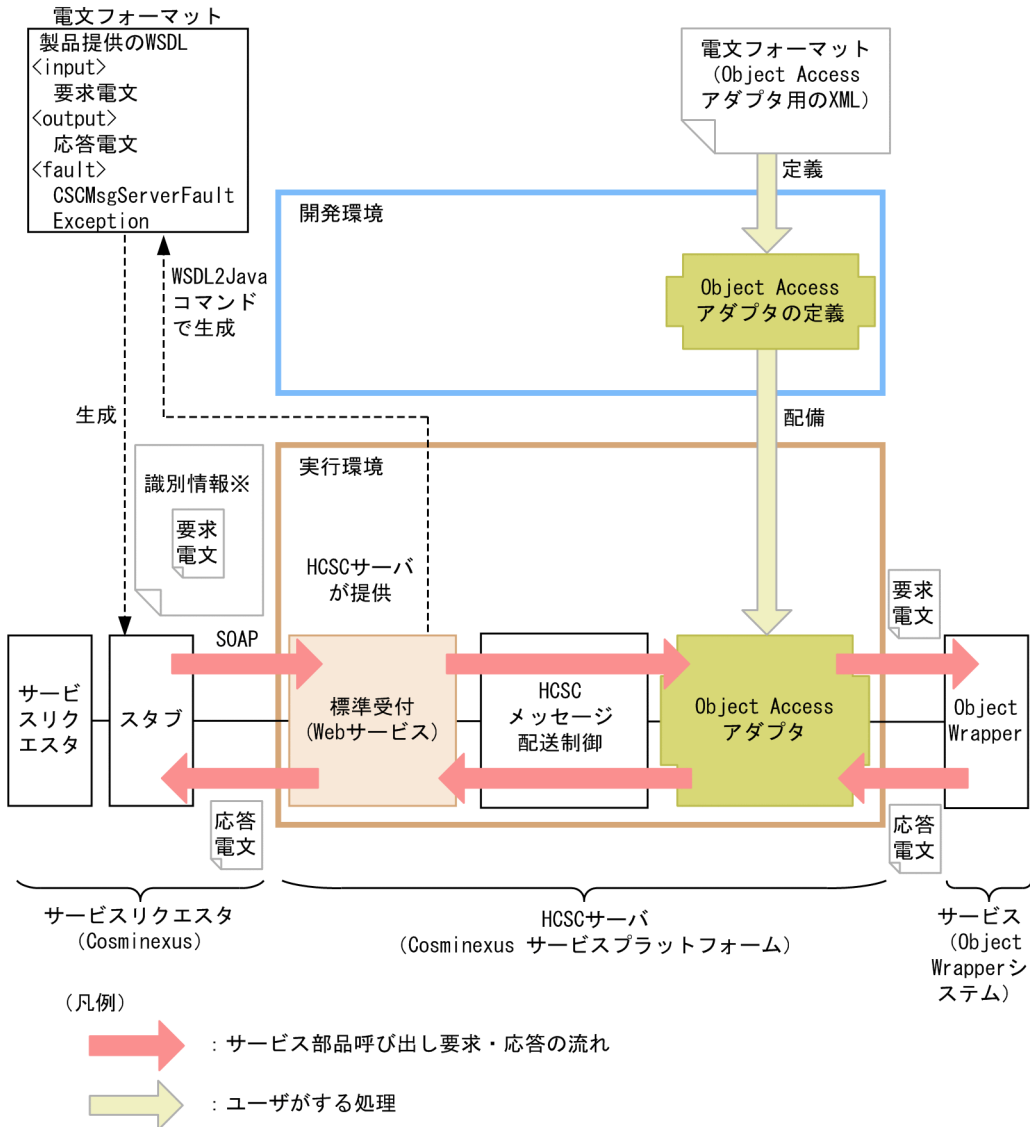
同期応答型のサービス部品呼び出しのほかに、非同期で HCSC サーバから、OpenTP1 システムのサービス部品を呼び出せます。非同期で HCSC サーバから、OpenTP1 システムのサービス部品を呼び出すには、HCSC サーバの非同期サービスアダプタ (MDB (DB キュー)) を実行します。

詳細は、「2.5 MDB (DB キュー) を使ったサービス部品呼び出し」を参照してください。

3.3 Object Wrapper システムとの接続

Object Access アダプタを使用することで、既存の TPBroker システム (Object Wrapper システム) のサービス部品を呼び出せます。HCSC サーバからの Object Wrapper システムのサービス部品の呼び出しを次の図に示します。

図 3-7 HCSC サーバからの Object Wrapper システムのサービス部品呼び出し

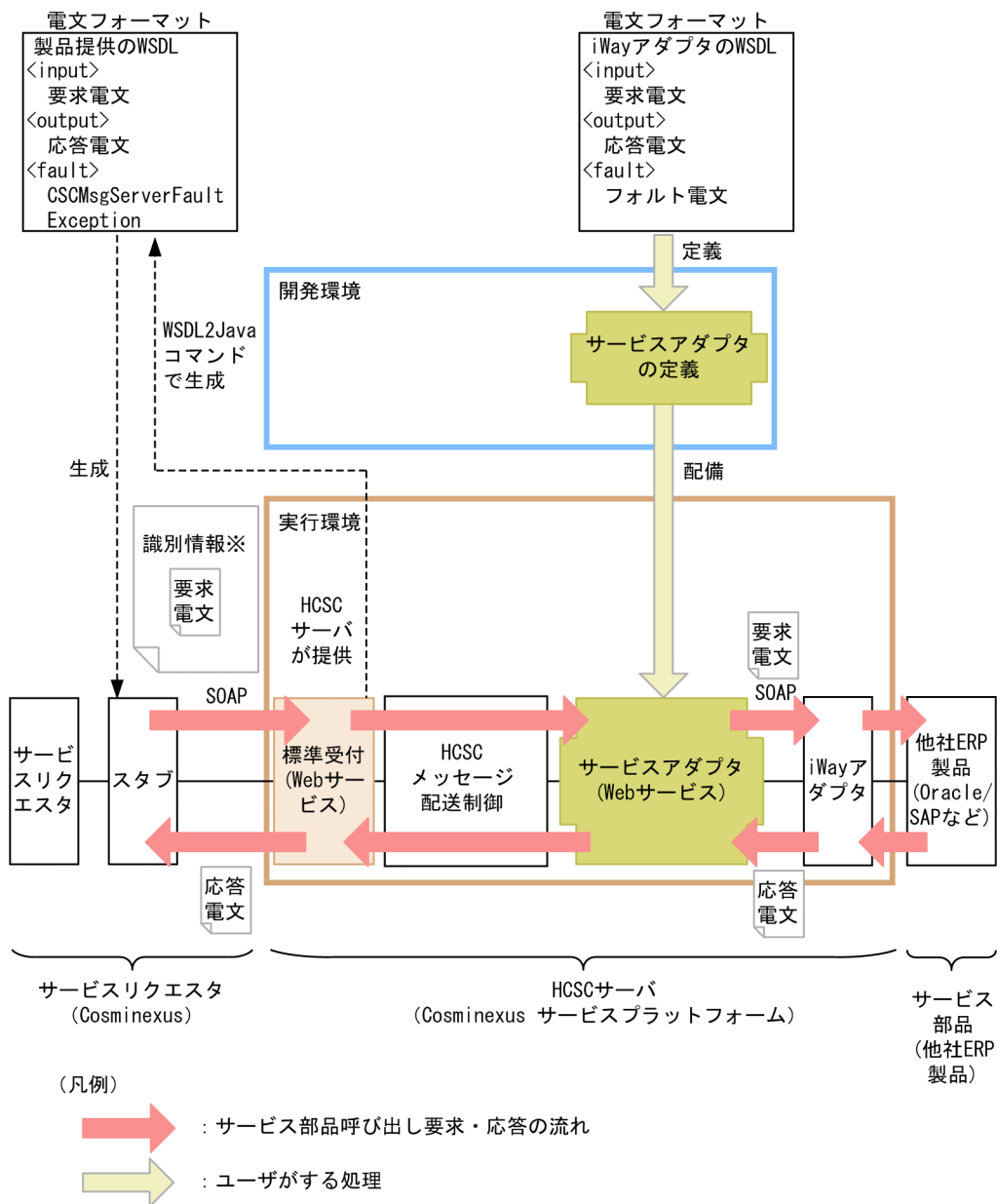


3.4 他社 ERP パッケージとの接続

iWay アダプタを使用することで、他社 ERP パッケージのサービス部品を呼び出せます。HCSC サーバからの他社 ERP パッケージのサービス部品の呼び出しを次の図に示します。

3. 標準プロトコル以外のシステムとの接続

図 3-8 HCSC サーバからの他社 ERP パッケージのサービス部品の呼び出し

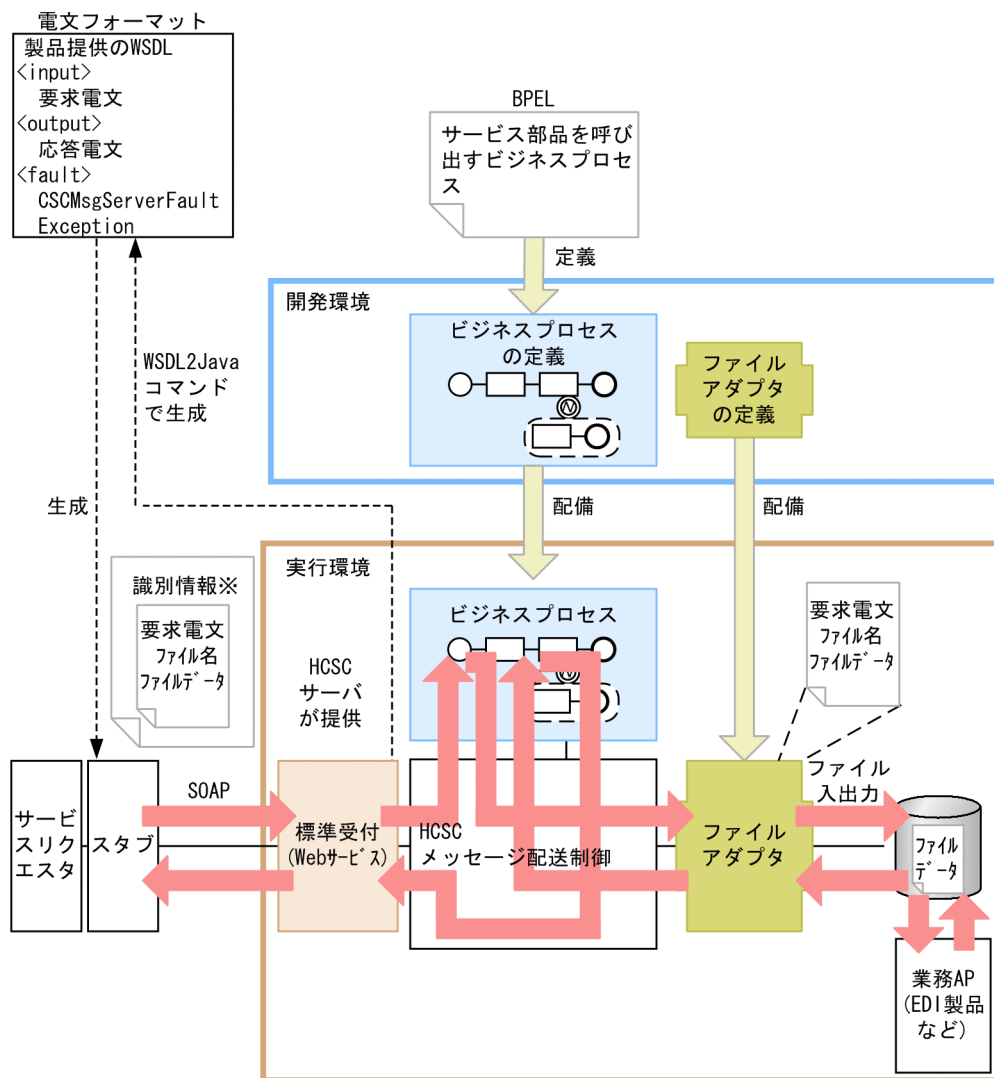


3.5 ファイルを扱うシステムとの接続

サービスリクエストやビジネスプロセスからの要求を受けて、HCSC サーバ上のローカルディスクに対して直接ファイルの入出力ができます。ファイルアダプタは、開発環境で定義します。ファイルアダプタによるファイルアクセスの実行について、次の図に示します。

3. 標準プロトコル以外のシステムとの接続

図 3-9 ファイルアダプタによるファイルアクセスの実行



(凡例)

➡ : サービス部品呼び出し要求・応答の流れ

➡ : ユーザがする処理

Ⓜ : フォルトコネクション

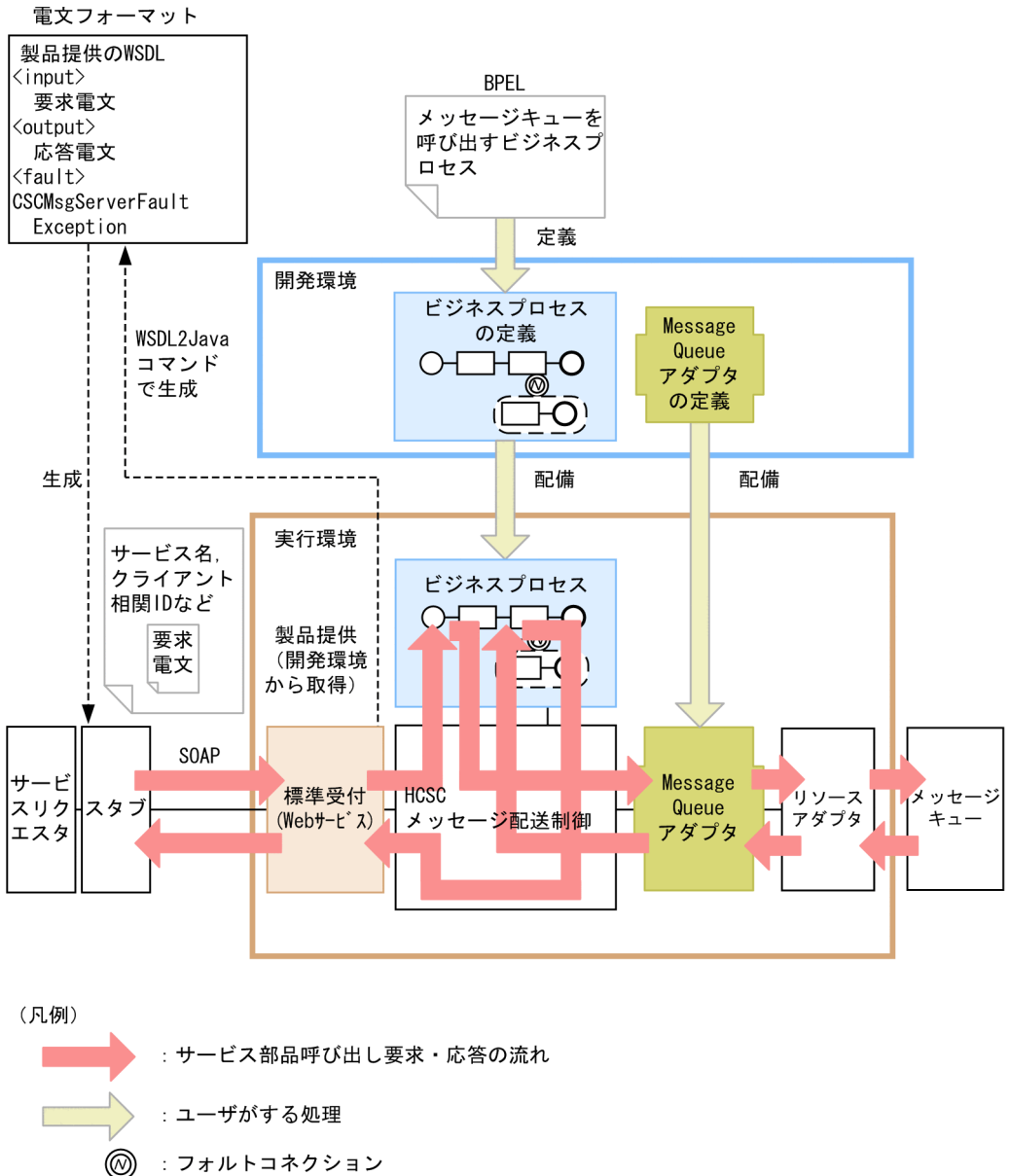
注※ 識別情報とは、サービス名、クライアント関連IDなどです。

ファイルアダプタを定義する手順の詳細については、マニュアル「Cosminexus サービスプラットフォーム ファイルアダプタ ユーザーズガイド」を参照してください。

3.6 メッセージキューとの接続

Message Queue アダプタを使用することで、既存のメッセージキュー（IBM WebSphere MQ システム）に対してメッセージの送受信ができます。Message Queue アダプタによるメッセージ送受信について、次の図に示します。

図 3-10 Message Queue アダプタによるメッセージ送受信の実行



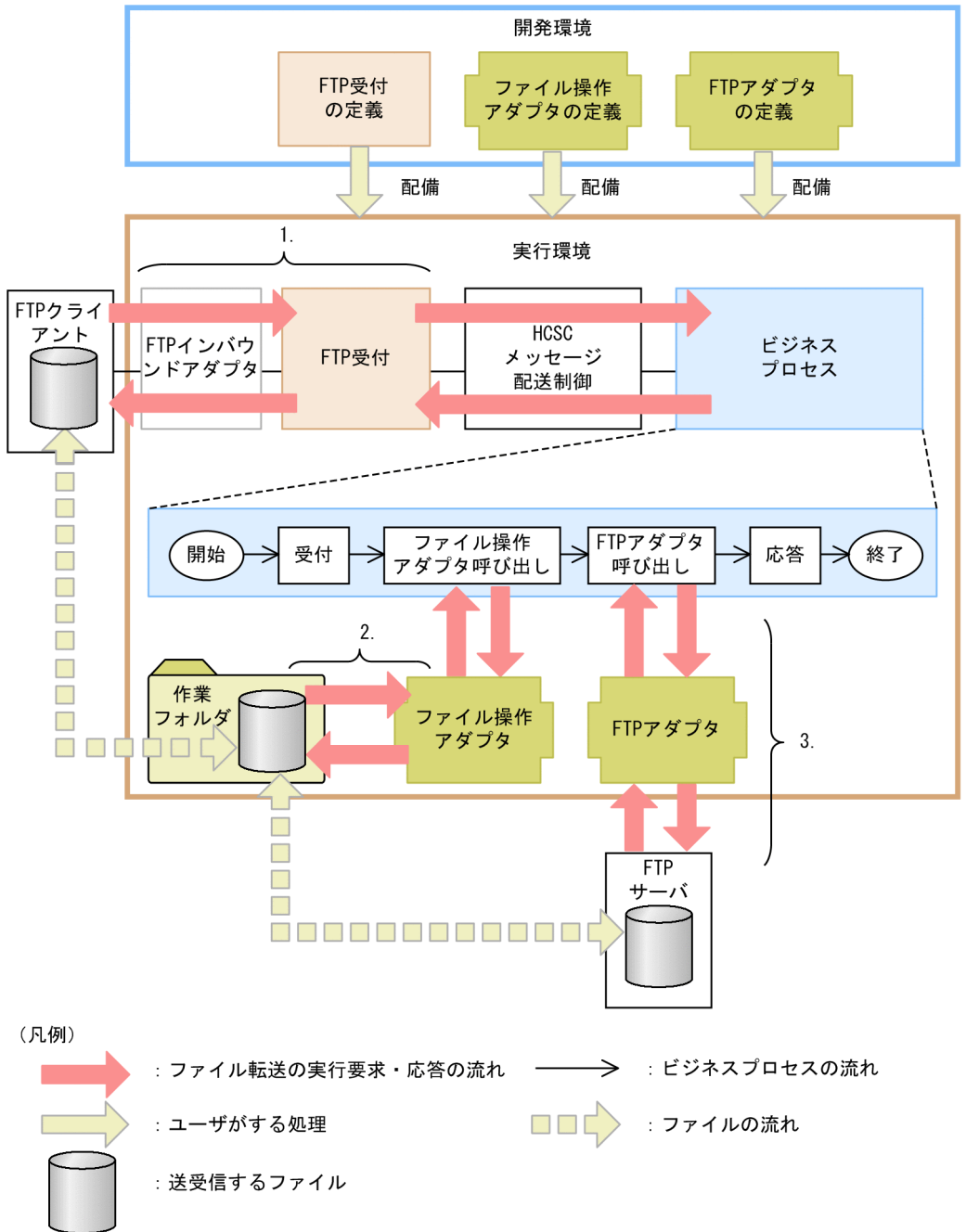
3.7 FTP クライアントおよび FTP サーバとの 接続

FTP アダプタ、ファイル操作アダプタ、および FTP 受付を使用することで、クライアント・サーバ間の FTP によるファイルの送受信を Cosminexus サービスプラットフォーム上で中継できます。

FTP アダプタ、ファイル操作アダプタ、および FTP 受付の機能詳細と定義手順については、マニュアル「Cosminexus サービスプラットフォーム 開発ガイド」の「第 3 編 FTP と連携したシステムの開発および運用」を参照してください。

FTP によるファイルの送受信の流れを次の図に示します。

図 3-11 FTP によるファイルの送受信の流れ



図中の 1. ~ 3. について説明します。

1. FTP クライアントからのリクエスト受付
FTP インバウンドアダプタおよび FTP 受付を使用して、FTP クライアントからの

3. 標準プロトコル以外のシステムとの接続

ファイル転送要求を受け付けます。FTP インバウンドアダプタでは、FTP クライアントの認証、ファイル転送処理などの処理を行います。FTP クライアントからのファイル転送要求は、FTP インバウンドアダプタを経由して FTP 受付に送られます。そのあと、FTP 受付からビジネスプロセスが呼び出されます。

ファイルの送受信は、FTP 受付が作成する HCSC サーバ上の作業フォルダを利用して行われます。

2. ファイルのデータ変換、複製、および削除

送信側のファイル形式と受信側のファイル形式が異なる場合、ファイル操作アダプタを使用してファイルのレイアウト変換、および文字コード変換を行います。

また、ファイル操作アダプタは、HCSC サーバ上の作業フォルダや任意のフォルダにアクセスし、フォルダ間のファイルの複製や削除、作業フォルダの削除などの操作を行います。

ファイル操作アダプタはビジネスプロセスから呼び出されます。

3. FTP サーバとの接続

FTP アダプタを使用して、FTP サーバとの接続や作業フォルダと FTP サーバとの間のファイル送受信を行います。

FTP アダプタはビジネスプロセスから呼び出されます。

3.8 カスタム受付を使用した場合のシステムとの接続

Cosminexus サービスプラットフォームでは、ユーザが任意の受付を作成できるようにカスタム受付フレームワークを提供しています。このカスタム受付フレームワーク上でユーザが作成した J2EE アプリケーションを動作させることで、任意のプロトコルに対応した受付（カスタム受付）を開発できます。

(1) カスタム受付の構成

カスタム受付は、開発者が作成する受付処理と、カスタム受付フレームワークから構成される J2EE アプリケーションです。

開発者が作成する受付処理の内容と、カスタム受付フレームワークで実行される処理の内容を次に示します。

開発者が作成する受付処理の内容

- サービスリクエストからのサービス部品呼び出し要求を受け付ける処理
- カスタム受付フレームワークに要求電文を渡す処理
- カスタム受付フレームワークから返ってきた応答電文を、サービスリクエストに返す処理

カスタム受付フレームワークで実行される処理

- 受付処理から渡された要求電文を、HCSC メッセージ配送制御に渡す処理
- HCSC メッセージ配送制御から返された応答電文を、受付処理に返す処理
- データ変換定義に従って、要求電文または応答電文のデータ変換を行う処理

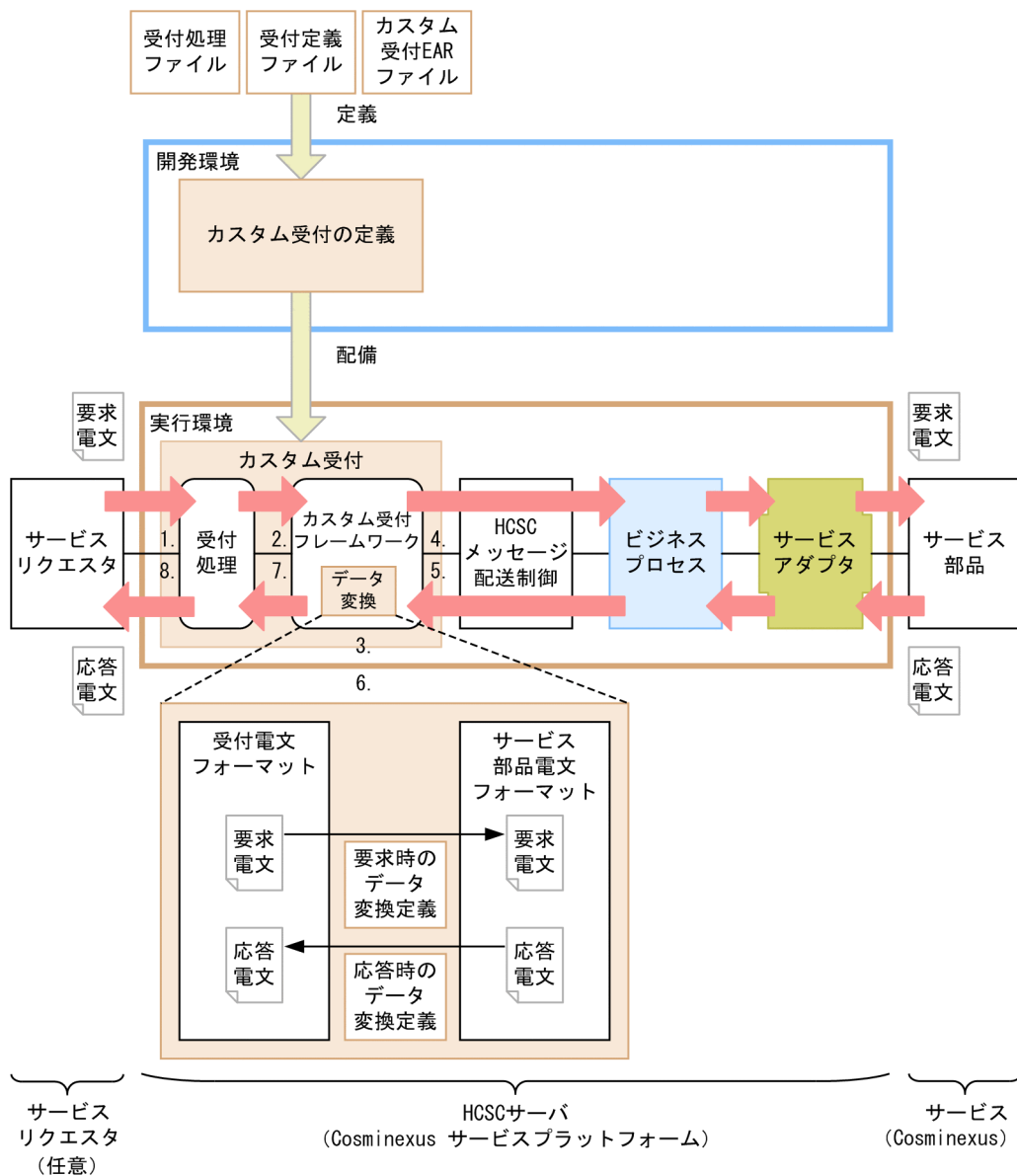
(2) カスタム受付フレームワークの概要

カスタム受付フレームワークは、カスタム受付を開発および動作させるためのフレームワークです。カスタム受付フレームワークは、開発者が作成したカスタム受付を、Cosminexus サービスプラットフォームの実行環境で受付として動作させるための仕組みを提供しています。

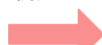

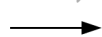
カスタム受付を使用したサービス部品の呼び出しを次の図に示します。

3. 標準プロトコル以外のシステムとの接続

図 3-12 カスタム受付を利用したサービス部品の呼び出し



(凡例)

-  : サービス部品呼び出し要求・応答の流れ
-  : ユーザがする処理
-  : データ変換処理

図中の 1. ~ 8. について説明します。

説明

1. 受付処理は、サービスリクエストからサービス部品呼び出し要求を受け付けます。
2. 受付処理は、サービスリクエストから受け取った要求電文を、カスタム受付フレームワークに渡します。
3. カスタム受付フレームワークは、要求電文のデータ変換をします。
4. カスタム受付フレームワークは、HCSC メッセージ配送制御を呼び出します。
5. カスタム受付フレームワークは、HCSC メッセージ配送制御から、応答電文を受け取ります。
6. カスタム受付フレームワークは、応答電文のデータ変換をします。
7. 受付処理は、カスタム受付フレームワークから応答電文を受け取ります。
8. 受付処理は、サービスリクエストに応答電文を返します。

3.9 カスタムアダプタを使用した場合のシステムとの接続

Cosminexus サービスプラットフォームが提供している次のアダプタでは対応できないプロトコルを持つシステムのサービス部品を呼び出して実行するには、カスタムアダプタの開発が必要です。

- サービスアダプタ (Web サービス , SessionBean , MDB (WS-R) , および MDB (DB キュー))
- DB アダプタ
- TP1 アダプタ
- Object Access アダプタ
- iWay アダプタ
- ファイルアダプタ
- Message Queue アダプタ
- FTP アダプタ
- ファイル操作アダプタ

Cosminexus サービスプラットフォームでは、カスタムアダプタの開発を容易にするためのフレームワークであるカスタムアダプタ開発フレームワークを提供しています。

カスタムアダプタの概要とカスタムアダプタ開発フレームワークの概要を次に示します。

(1) カスタムアダプタの概要

カスタムアダプタは、サービスリクエストからの要求を受けサービス部品を呼び出します。そのとき、次の二つの役割を果たします。

1. データ変換
アダプタにデータ変換定義が含まれる場合、事前に定義された情報に基づいて電文のデータ変換をします。
2. プロトコル変換
サービス部品のプロトコルにあわせてプロトコル変換し、サービス部品を呼び出します。

(2) カスタムアダプタ開発フレームワークの概要

カスタムアダプタが行う処理には、データ変換とプロトコル変換があります。このうち、すべてのカスタムアダプタに共通の処理であるデータ変換処理をカスタムアダプタ開発フレームワークで提供しています。このため、ユーザはプロトコル変換処理を実装することでカスタムアダプタを開発できます。

! 注意事項

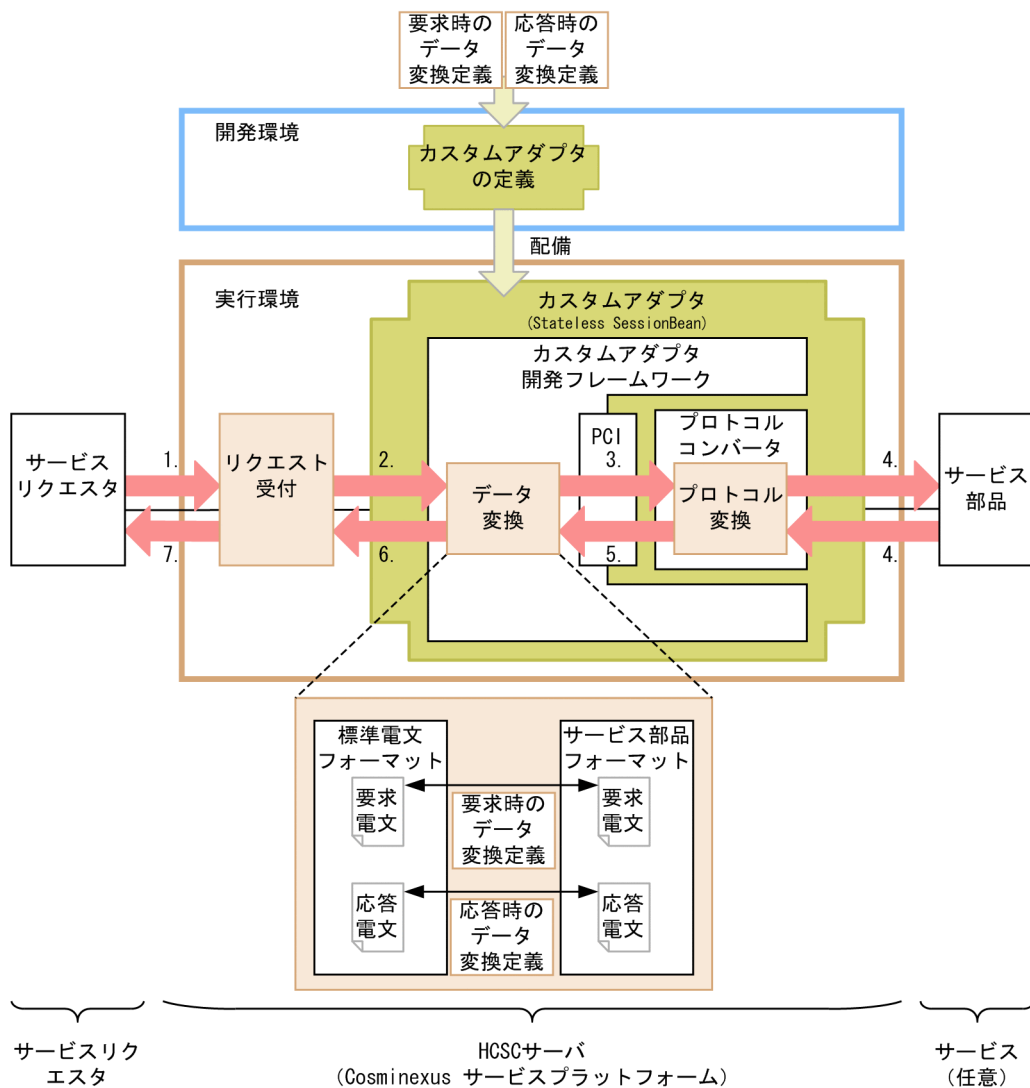
データ変換の定義は、別途必要となります。

カスタムアダプタは、カスタムアダプタ開発フレームワークの実装である Stateless SessionBean と、プロトコルコンバータで構成されます。プロトコルコンバータとは、プロトコル変換処理を実装する Java クラスのことです。

カスタムアダプタを使用したサービス部品の呼び出しを次の図に示します。

3. 標準プロトコル以外のシステムとの接続

図 3-13 カスタムアダプタを使用したサービス部品の呼び出し（すべて同期通信の場合）



(凡例)

PCI : プロトコルコンバータインタフェース (Protocol Converter Interface)

➡ : ユーザがする処理

➡ : サービス部品呼び出し要求・応答の流れ

➡ : データ変換処理

図中の 1. ~ 7. について説明します。

説明

1. HCSC サーバは、カスタムアダプタ経由で呼び出すサービス部品へのリクエスト

を受け取ります。

2. HCSC サーバは、EJB リモートインターフェースを使用して、カスタムアダプタ開発フレームワークで実装しているカスタムアダプタを呼び出します。
3. カスタムアダプタ開発フレームワークは、事前に定義された情報に基づいてデータ変換処理を実行し、プロトコルコンバータインターフェースを使用してプロトコルコンバータを呼び出します。
4. プロトコルコンバータは、サービス部品のプロトコルに応じた送受信処理を行います。
5. サービス部品の呼び出し結果をカスタムアダプタ開発フレームワークに返します。
6. カスタムアダプタ開発フレームワークは、事前に定義された情報に基づいてデータ変換処理を実行し、HCSC サーバに応答を返します。
7. サービスリクエストに応答を返します。

カスタムアダプタ開発フレームワークによるカスタムアダプタの開発については、マニュアル「Cosminexus サービスプラットフォーム 開発ガイド」の「付録 F カスタムアダプタ開発フレームワーク」を参照してください。

4

ビジネスプロセスを使った運用

この章では、ビジネスプロセスの機能、ビジネスプロセスの仕組み、ビジネスプロセスの永続化・非永続化、ビジネスプロセスのトランザクション、ビジネスプロセスの再実行、およびビジネスプロセスの待機処理について説明します。また、ビジネスプロセスと関係が深いプロセスインスタンス、アクティビティ、変数、および相関セットについても説明します。

4.1 ビジネスプロセスの機能

4.2 ビジネスプロセスの仕組み

4.3 永続化するビジネスプロセスと永続化しないビジネスプロセス

4.4 ビジネスプロセスのトランザクション

4.5 ビジネスプロセスの再実行

4.6 ビジネスプロセスを使用した待機処理

4.1 ビジネスプロセスの機能

ビジネスプロセスとは、複数のサービス部品を実行する処理の順番や条件などを定義して、一連の業務の流れとして定義したものです。

ビジネスプロセスは、Cosminexus サービスプラットフォームの開発環境で処理フローを定義します。処理フローは、ビジネスプロセス定義画面で定義します。画面上にアクティビティと呼ばれる要素を配置し、コネクションで連結して定義します。

ビジネスプロセス定義画面の詳細については、マニュアル「Cosminexus サービスプラットフォーム リファレンス」の「1.2.3 ビジネスプロセス定義画面」を参照してください。

ここでは、ビジネスプロセスを構成する主な次の八つの要素について説明します。

- 受付・応答
- サービス呼び出し
- Java 呼び出し
- データ変換
- 代入
- 繰り返し
- 分岐
- 並列処理

4.1.1 受付・応答

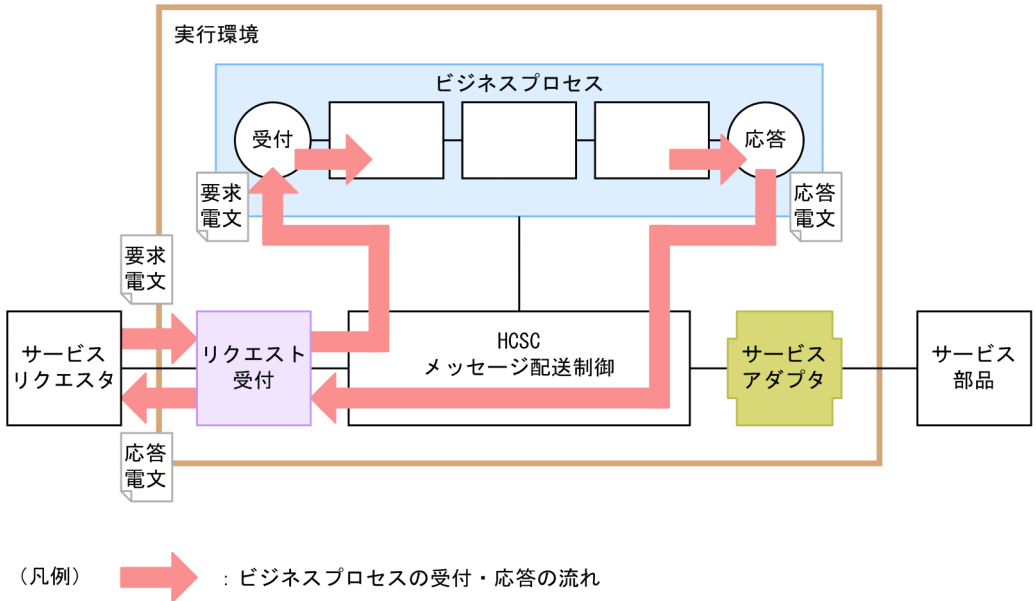
(1) 機能

ビジネスプロセスの受付でサービスリクエストからの要求を受け付け、応答でビジネスプロセスからの応答をサービスリクエストに返します。

サービスリクエストからの要求を受け付けるリクエスト受付には、標準受付とユーザ定義受付の両方を使用できます。

ビジネスプロセスの受付と応答の流れを次の図に示します。

図 4-1 ビジネスプロセスの受付と応答の流れ



(2) 使用するアクティビティ

受付には受付アクティビティを使用し、応答には応答アクティビティを使用します。

受付には、次のどちらかの通信モデルが定義できます。

- 同期で応答電文が返るビジネスプロセス
- 非同期で応答電文が返らないビジネスプロセス

HCSC サーバで受け付けた要求電文はビジネスプロセスに渡されるため、ビジネスプロセスで受け付けることができる XML 電文形式を受付アクティビティに定義します。また、応答アクティビティに定義した XML 電文形式で、サービスリクエスタに応答電文が返ります。

4.1.2 サービス呼び出し

(1) 機能

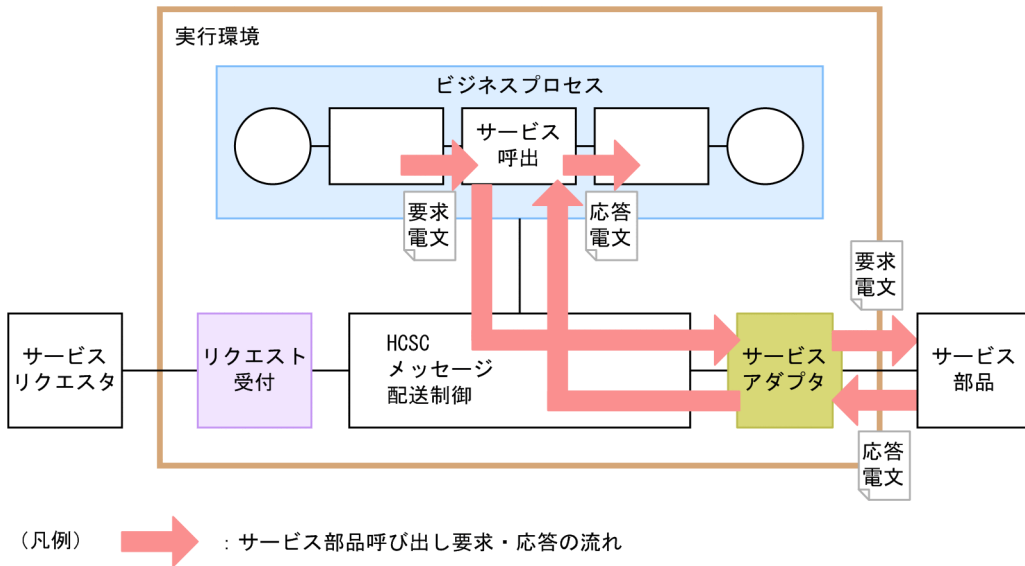
サービス呼び出しでは、Cosminexus サービスプラットフォームの開発環境で定義した各種サービスアダプタやビジネスプロセスなどのサービス部品を実行することができます。

サービスアダプタやビジネスプロセスの通信モデルによって、同期で応答電文が返る場合と、非同期で応答電文が返らない場合のそれぞれのサービス部品を呼ぶことができます。

サービス呼び出しでサービスアダプタを実行する場合の流れを次の図に示します。

4. ビジネスプロセスを使った運用

図 4-2 サービス呼び出しでサービスアダプタを実行する場合の流れ



(2) 使用するアクティビティ

サービス呼び出しには、サービス呼出アクティビティを使用します。

サービス呼出アクティビティの要求電文に定義した XML 電文形式でサービスアダプタまたはビジネスプロセスを実行します。

サービスアダプタまたはビジネスプロセスからの応答は、サービス呼出アクティビティの応答電文に定義した XML 電文形式になります。

4.1.3 Java 呼び出し

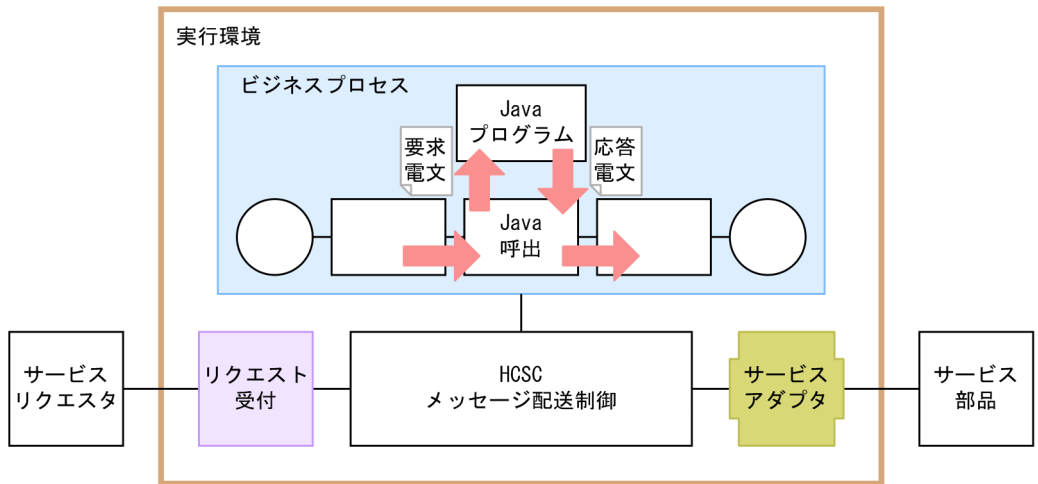
(1) 機能


Java 呼び出しでは、Cosminexus サービスプラットフォームが提供する専用のインターフェースを実装した Java クラスを実行することができます。

Cosminexus サービスプラットフォームが提供する機能だけでビジネスプロセスが定義できないような場合に、ユーザが作成する Java プログラムを使用して、XML 電文の変換や判定などの処理を実行できます。

Java 呼び出しで Java プログラムを実行する場合の流れを次の図に示します。

図 4-3 Java 呼び出しで Java プログラムを実行する場合の流れ



(凡例)  : Javaプログラム呼び出し要求・応答の流れ

(2) 使用するアクティビティ

Java 呼び出しには、Java 呼出アクティビティを使用します。

Java 呼出アクティビティの引数に定義した XML 電文形式やビジネスプロセス上の変数で Java プログラムを実行します。

Java プログラムの戻り値は、Java 呼出アクティビティの戻り値に定義した XML 電文形式やビジネスプロセス上の変数になります。

なお、実行する Java プログラムは、Cosminexus サービスプラットフォームの開発環境の Java 呼出アクティビティを使用し、Java エディタ上で作成する必要があります。

4.1.4 データ変換

(1) 機能

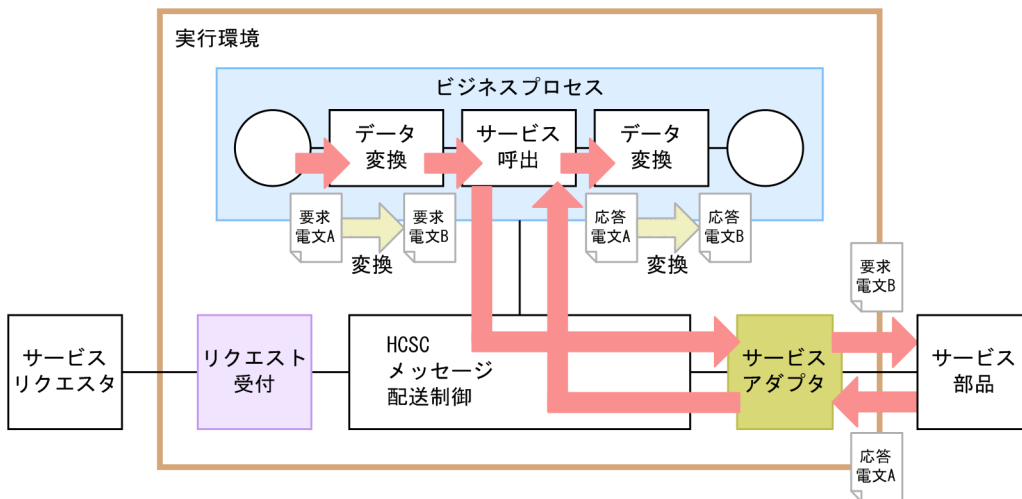
データ変換では、ビジネスプロセス上を流れる電文形式を異なる電文形式に変換したり、値を加工したりすることができます。また、複数の電文を組み合わせると一つの電文を作ることができます。

データ変換は、サービス部品を呼び出すときに、サービス部品の要求電文形式に合わせた XML 電文を作成する必要がある場合や、サービス部品からの応答電文を異なる XML 電文に変換したい場合などに使用します。

ビジネスプロセスでのデータ変換の流れを次の図に示します。

4. ビジネスプロセスを使った運用

図 4-4 ビジネスプロセスでのデータ変換の流れ



(凡例)

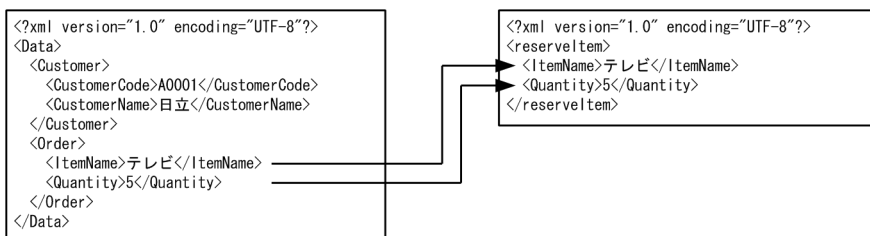
- : データ変換を含むサービス部品呼び出し要求・応答の流れ
- : データ変換の流れ

(2) 使用するアクティビティ

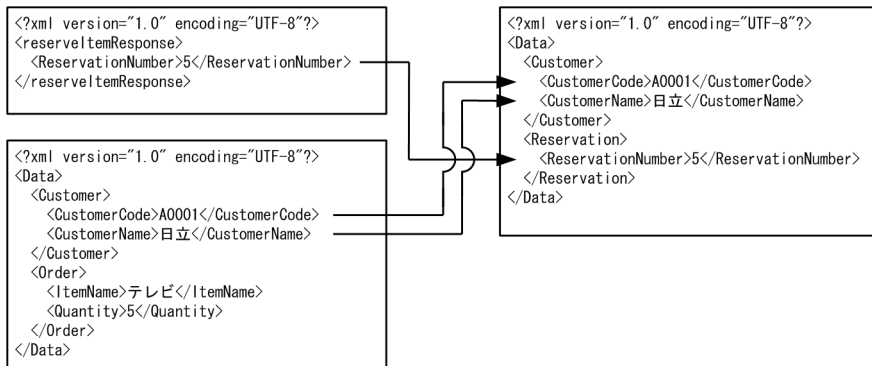
データ変換には、データ変換アクティビティを使用します。

データ変換の例を次に示します。

(例1) 1対1でのデータ変換



(例2) N対1でのデータ変換



XML 電文の値の加工のできる処理を次の表に示します。

表 4-1 XML 電文の値の加工のできる処理一覧

処理	説明
文字列連結	複数の文字列を連結します。
部分文字列取得	一つの文字列から一部分の文字列を取り出します。
文字数取得	文字列の文字数をマッピング先に代入します。
文字列調査	文字列に指定した文字列が含まれているか、または指定した文字列で開始されているかを調査し、マッピング先に論理値を代入します。
空白除去	文字列の先頭と末尾の空白を除去します。また、文字列間の連続した空白を一つの半角スペースに置換します。
数値フォーマット変換	数値をフォーマット変換します。
数値演算	数値を演算 (+, -, *, div, mod) します。
端数処理	小数点以下の端数を処理 (四捨五入, 切り捨て, および切り上げ) します。
総和	複数のノードセットの、ノードの数値を合計します。
ノード数取得	ノード数をマッピング先に代入します。
ノード名取得	ノード名をマッピング先に代入します。
ノード調査	指定した変換元ノードが存在するかを調査し、マッピング先にその論理値を代入します。
繰り返し	繰り返しを対応づけます。
選択	条件によって異なる値を出力します。
定数	指定した値をマッピング先に代入します。
値置換え	変換テーブルの指定に基づいて、マッピング元の値を変換し、マッピング先に代入します。
カスタム	ユーザが作成した Java プログラムで値を加工します。

4.1.5 代入

(1) 機能

代入では、ビジネスプロセス上を流れる電文形式を異なる電文形式に変換することができます。

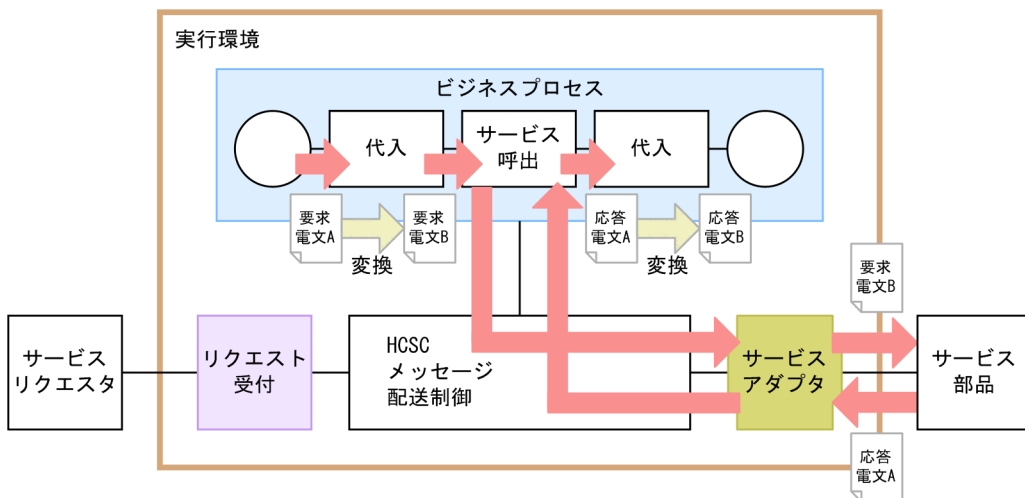
代入の主な機能を次に示します。

- XML 電文を異なる名称の XML 電文にコピーする。
- XML 電文の一部（部分木）をほかの XML 電文にコピー（挿入）する。
- XML 以外の boolean, numeric, string 等の変数をほかの変数にコピーする。



代入は、データ変換と同じような機能を持ちますが、XML 電文中の個々の値を対象とするのではなく、XML 全体または一部をそのままコピーしたい場合に適しています。

ビジネスプロセスでの代入の流れを次の図に示します。

図 4-5 ビジネスプロセスでの代入の流れ



(凡例)

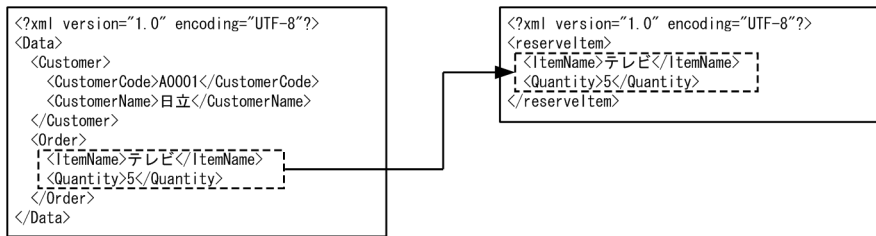
-  : 代入を含むサービス部品呼び出し要求・応答の流れ
-  : 代入の流れ

(2) 使用するアクティビティ

代入には、代入アクティビティを使用します。

代入の例を次に示します。

(例) 部分木のコピー



4.1.6 繰り返し

(1) 機能

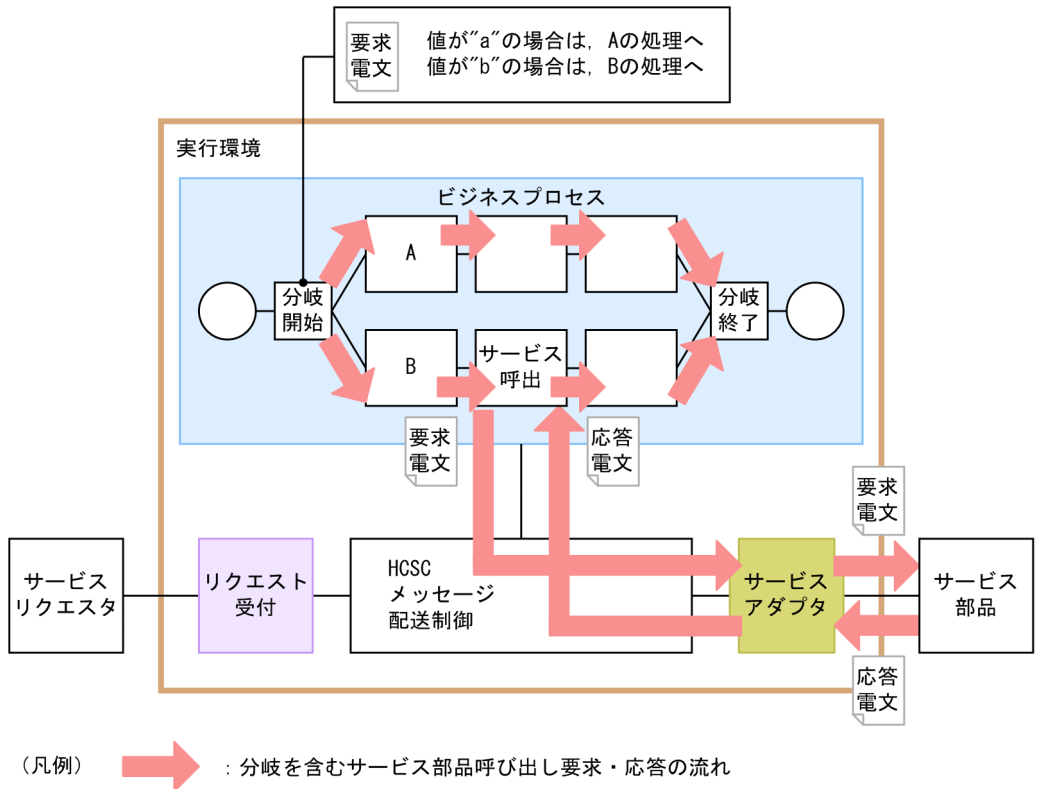
繰り返しでは、ビジネスプロセスの中で、一定の条件を満たすまでの間、処理を繰り返すこと（ループ処理）ができます。

繰り返し処理の中には、一つまたは複数のアクティビティを配置します。

繰り返しの判定は、前判定になります。判定結果が真の場合は繰り返しを続け、偽の場合は繰り返しから抜けて次の処理へ遷移します。

ビジネスプロセスでの繰り返しの流れを次の図に示します。

図 4-7 ビジネスプロセスでの分岐



(2) 使用するアクティビティ

分岐の開始個所には分岐開始アクティビティを使用し、終了個所には分岐終了アクティビティを使用します。分岐先は、複数（二つ以上）設定できます。

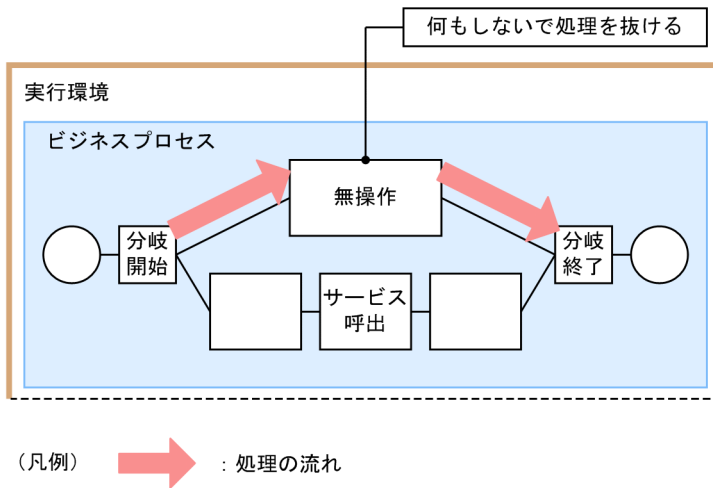
分岐の判定に用いる条件式は、XML 電文などの変数に対して XPath で指定します。この条件式を基に、それぞれの遷移先のアクティビティを指定します。

なお、分岐開始アクティビティと、対応する分岐終了アクティビティの間には、一つ以上のアクティビティを設定する必要があります。そのため、片方の分岐先で処理を何もしないで次へ遷移するようにしたい場合は、無操作アクティビティを配置します。

片方の分岐先で処理を何もしないで次へ遷移するようにしたい場合のアクティビティの配置例を次の図に示します。

4. ビジネスプロセスを使った運用

図 4-8 片方の分岐先で処理を何もしないで次へ遷移してほしい場合のアクティビティの配置例



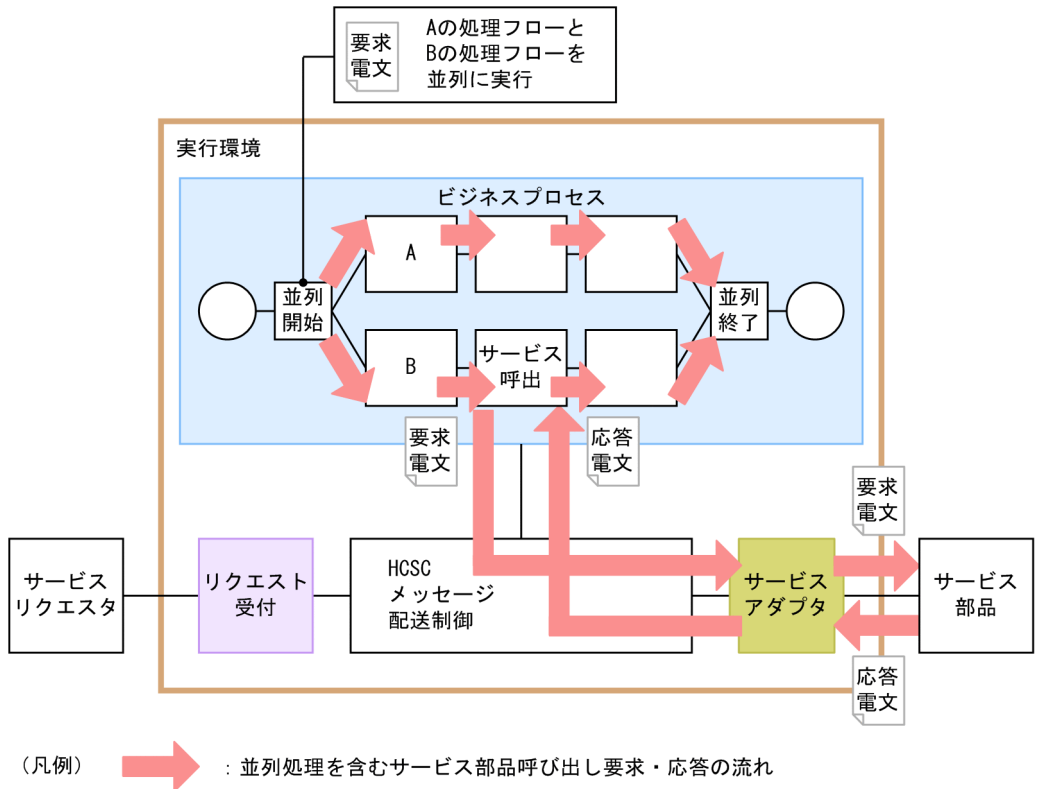
4.1.8 並列処理

(1) 機能

並列処理では、ビジネスプロセスの中で、処理フローを二つ以上に分け、フローごとに順次実行することができます。

ビジネスプロセスでの並列実行の流れを次の図に示します。

図 4-9 ビジネスプロセスでの並列実行の流れ



(2) 使用するアクティビティ

並列処理の開始個所には並列処理開始アクティビティを、終了個所には並列処理終了アクティビティを使用します。

また、リンクコネクションを用いることで、個々の並行に実行する処理フローの順番を制御することができます。

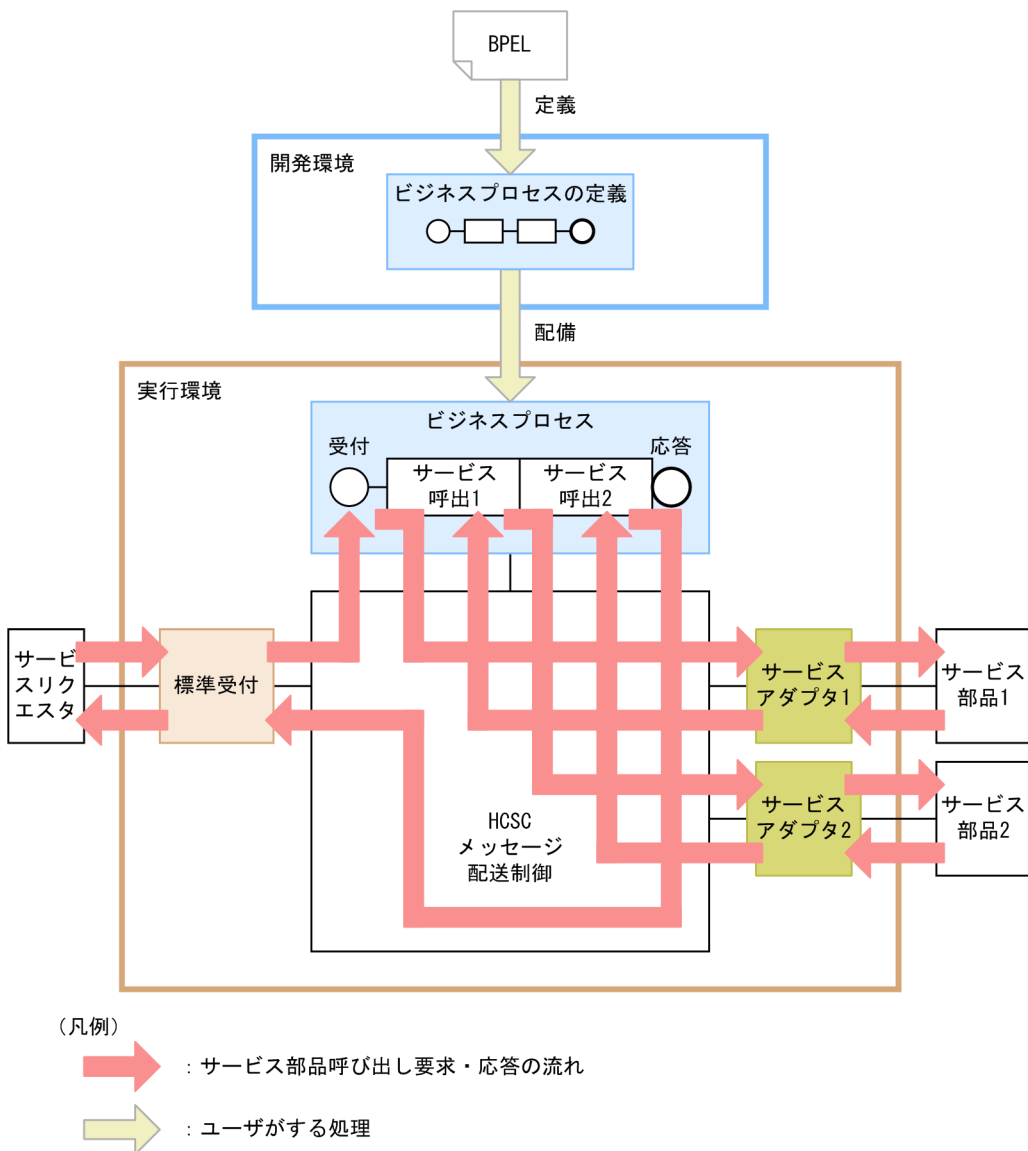
なお、並列処理は個々の処理フローを順次実行します。複数の処理フローを同時に実行することはできません。

4.2 ビジネスプロセスの仕組み

4.2.1 ビジネスプロセスの処理の流れ

サービスリクエスタからビジネスプロセスを呼び出すと、呼び出されたビジネスプロセスで定義された要素やアクティビティを一つ一つ実行することで、プロセスを実行します。ビジネスプロセスの定義は、開発環境で定義します。ビジネスプロセスの処理の流れを次の図に示します。

図 4-10 ビジネスプロセスの処理の流れ



4.2.2 ビジネスプロセスの通信モデル

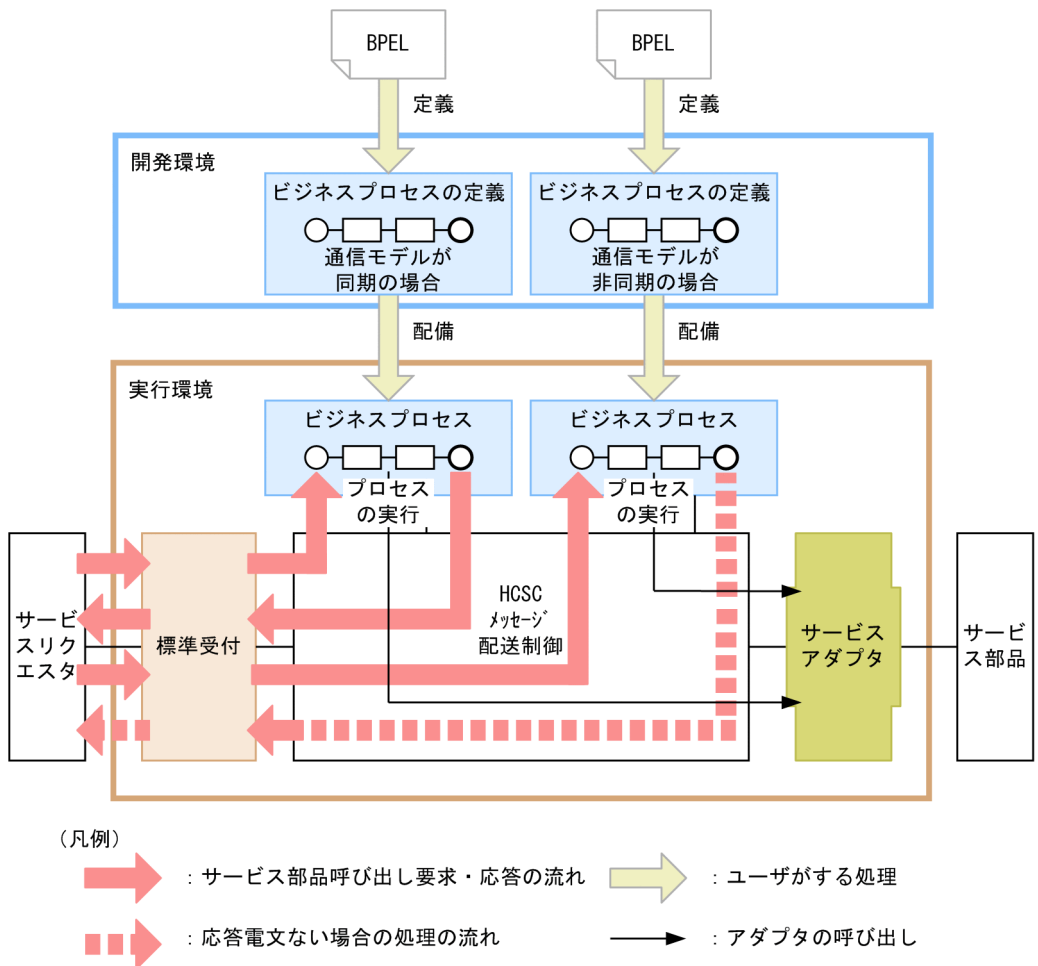
ビジネスプロセスは、同期と非同期の2種類の通信モデルがあります。

通信モデルが同期の場合、ビジネスプロセスを呼び出したサービスリクエストに対して、応答電文が返ります（応答電文は応答アクティビティに定義した電文フォーマットです）。

通信モデルが非同期の場合、ビジネスプロセスを呼び出したサービスリクエストに応答電文が返りません（処理だけがリターンします）。

通信モデルは、開発環境でビジネスプロセス定義するときに、受付アクティビティのダイアログで定義します。ビジネスプロセスでの通信モデルの違いを次の図に示します。

図 4-11 ビジネスプロセスでの通信モデルの違い



受付アクティビティダイアログの詳細については、マニュアル「Cosminexus サービスプラットフォーム リファレンス」の「1.4.7 受付アクティビティダイアログ」を参照し

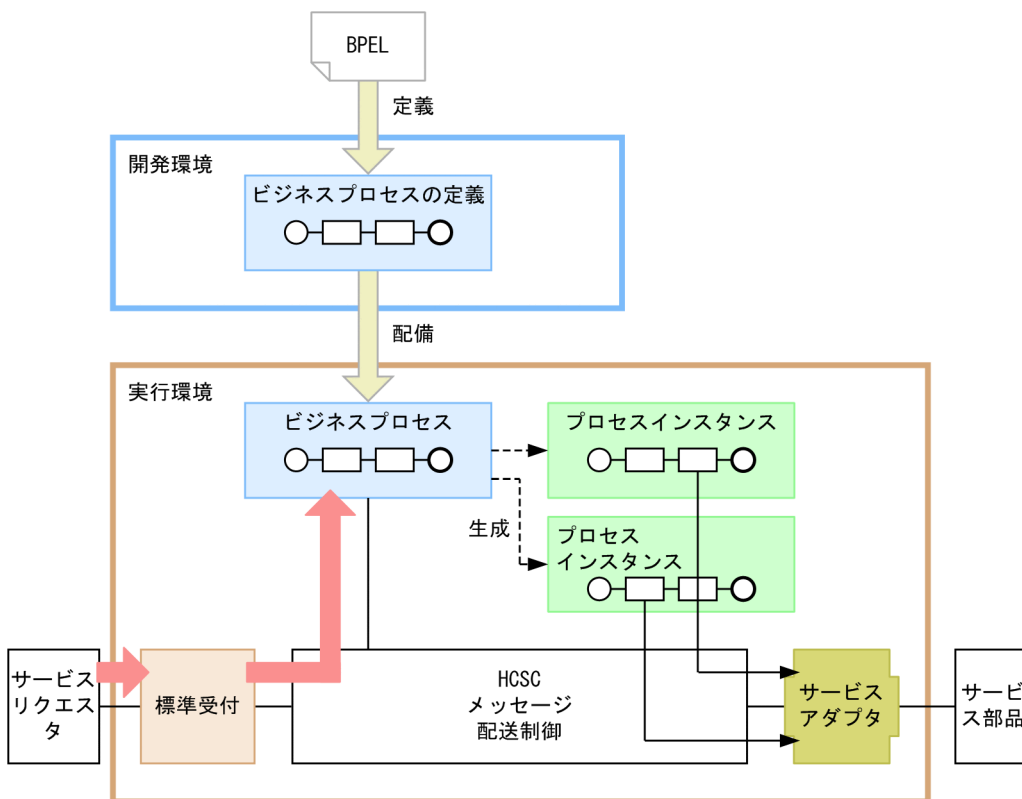
てください。

4.2.3 プロセスインスタンス

ビジネスプロセスが最初に要求を受け付けるときには、プロセスインスタンスを生成します。プロセスインスタンスを生成するには、開発環境でビジネスプロセスを定義するとき、最初に要求を受け付ける受付アクティビティを定義する受付アクティビティダイアログでインスタンス生成を「yes」に設定します。

プロセスインスタンスを生成するビジネスプロセスの場合、ビジネスプロセスが要求を受け付けるたびにプロセスインスタンスを生成します。そのため、複数の要求が同時に実行された場合でも、個々に独立してプロセスを実行します。ビジネスプロセスでのプロセスインスタンスの仕組みについて次の図に示します。

図 4-12 ビジネスプロセスでのプロセスインスタンスの仕組み



(凡例)

- ➡ : サービス部品呼び出し要求・応答の流れ
- ➡ : ユーザがする処理
- ➡ : プロセスインスタンスの生成
- ➡ : アダプタの呼び出し

受付アクティビティダイアログの詳細については、マニュアル「Cosminexus サービス

プラットフォームリファレンスの「1.4.7 受付アクティビティダイアログ」を参照してください。

プロセスインスタンスの属性情報を次の表に示します。

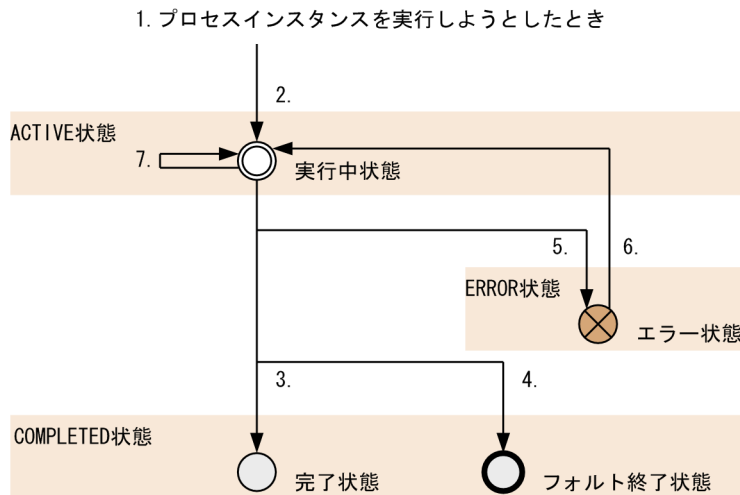
表 4-2 プロセスインスタンスの属性情報

属性	説明
識別子	個々のプロセスインスタンスを特定するための識別子です。同一のビジネスプロセス定義（バージョンは異なる場合もある）内で一意な識別子です。
ビジネスプロセス定義名	対象のプロセスインスタンスが所属するビジネスプロセスの定義名称です。サービスリクエストから呼び出すときのサービス名に相当します。
ビジネスプロセス定義バージョン	対象のプロセスインスタンスが所属するビジネスプロセス定義バージョンです。
状態	対象のプロセスインスタンスの状態です。
開始時刻	対象のプロセスインスタンスの開始時刻（GMT）です。 注 ACTIVE 状態に遷移した時刻
終了時刻	対象のプロセスインスタンスの終了時刻（GMT）です。 注 COMPLETED 状態に遷移した時刻

プロセスインスタンスの状態遷移を次の図に示します。

4. ビジネスプロセスを使った運用

図 4-13 プロセスインスタンスの状態遷移



(凡例)

- : 実行中のプロセスインスタンス
- : 終了したプロセスインスタンス
- : フォルト状態のプロセスインスタンス
- ⊗ : エラー状態のプロセスインスタンス
- : プロセスインスタンスの状態遷移

1. プロセスインスタンスを実行しようとしたとき。
2. プロセスインスタンスに所属するアクティビティを実行しようとしたとき。
3. プロセスインスタンスに所属するグローバルスコープが完了し、プロセスインスタンスが正常に終了したとき。
4. プロセス実行中にフォルトが発生し、グローバルスコープでもキャッチできないでプロセスインスタンスがフォルト終了したとき。
5. サービス呼び出しアクティビティで例外が発生し、プロセスインスタンスがエラーで中断したとき。
6. エラー状態のプロセスインスタンスに対して再実行を実施したとき。
7. 実行中状態のプロセスインスタンスに対して再実行を実施したとき。

図 4-13 で示すプロセスインスタンスの状態を次の表に示します。

表 4-3 プロセスインスタンスの状態

状態		説明
ACTIVE 状態	実行中状態 (Executing)	プロセスインスタンスが実行中の状態です。

状態		説明
ERROR 状態	エラー状態 (Error)	サービス呼び出しアクティビティで例外が発生し、プロセスインスタンスがエラーで中断した状態です。
COMPLETED 状態	完了状態 (Completed)	プロセスのグローバルスコープが完了して、プロセスインスタンスが完了した状態です。
	フォルト終了状態 (Faulted)	フォルトが発生し、グローバルスコープでもキャッチできないでプロセスインスタンスがフォルト終了した状態です。

4.2.4 アクティビティ

アクティビティは、ビジネスプロセスの構成要素となる部品で、処理の構造を表現したものです。複数のアクティビティを連結して、ビジネスプロセスの処理フローを定義します。

開発環境のビジネスプロセス定義画面でアクティビティを配置、連結して定義します。各アクティビティの属性情報を次の表に示します。

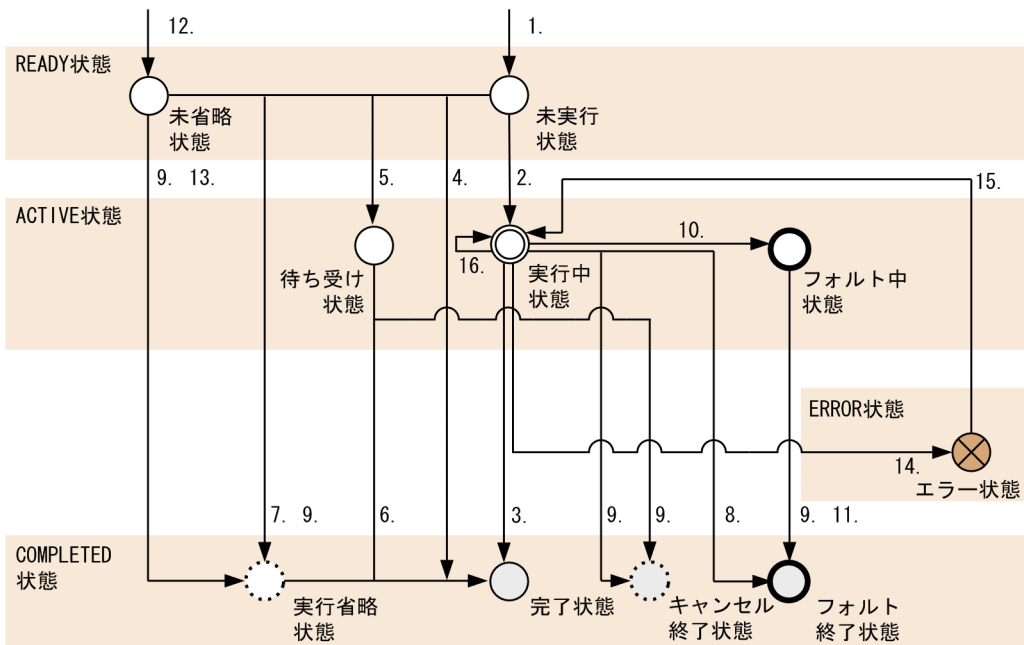
表 4-4 アクティビティの属性情報

属性	説明
プロセスインスタンス識別子	対象のアクティビティが所属するプロセスインスタンスの識別子です。
アクティビティ番号	アクティビティを識別するための番号です。アクティビティリストのインデックスに対応します。
ビジネスプロセス定義名	対象のアクティビティが所属するビジネスプロセスの定義名称です。サービスリクエストから呼び出すときのサービス名に相当します。
ビジネスプロセス定義バージョン	対象のアクティビティが所属するビジネスプロセス定義バージョンです。
アクティビティ定義名	対象のアクティビティのアクティビティ定義名称です。
アクティビティ種別	対象のアクティビティのアクティビティ種別です。
状態	対象のアクティビティの状態です。
開始時刻	対象のアクティビティの開始時刻 (GMT) です。 注 ACTIVE 状態に遷移した時刻
終了時刻	対象のアクティビティの終了時刻 (GMT) です。 注 COMPLETED 状態に遷移した時刻
待機解除時刻	対象のアクティビティの待機解除時刻 (GMT) です。 注 待機アクティビティのときに使用します。

アクティビティの状態遷移を次の図に示します。

4. ビジネスプロセスを使った運用

図 4-14 アクティビティの状態遷移



(凡例)

- ◎ : 実行中のアクティビティインスタンス
- : 実行中以外の終了していないアクティビティインスタンス
- : 終了したアクティビティインスタンス
- ⊙ : 途中で存在しなくなったアクティビティインスタンス
- ⦿ : フォルト状態のアクティビティインスタンス
- ⊗ : エラー状態のプロセスインスタンス
- : アクティビティインスタンスの状態遷移

1. 新たなアクティビティのインスタンスが生成されたとき。
2. アクティビティインスタンスを実行しようとしたとき。
3. アクティビティインスタンスの実行が完了したとき。
4. アクティビティインスタンスを実行したが、実行中にならないで実行が完了したとき。
5. サービスリクエストからの要求を受信した受付以外の受付アクティビティインスタンスを実行しようとしたとき。
6. 待ち受け状態の受付アクティビティインスタンスがサービスリクエストからの要求を受けたとき。
7. target link が false で活性化されたとき。
8. アクティビティインスタンスの実行時にフォルトが発生したとき。

9. フォルトが発生し、アクティビティの実行がキャンセル処理を依頼されたとき。
10. フォルトが発生し、フォルトハンドラでキャッチして対応するアクティビティインスタンスを実行中のとき。
11. フォルトが発生し、キャッチに対応するアクティビティインスタンスの実行が完了したとき。
12. 構造化アクティビティが実行省略された際に、内包するアクティビティを実行省略にするととき。
13. 未省略状態のアクティビティの実行を省略するとき。
14. サービス呼び出しアクティビティの処理で例外が発生し、アクティビティインスタンスの処理がエラーで中断したとき。
15. エラー状態のサービス呼び出しアクティビティのアクティビティインスタンスに対して再実行を実施したとき。
16. 実行中状態のアクティビティインスタンスに対して再実行を実施したとき。

図 4-14 で示すアクティビティの状態を次の表に示します。

表 4-5 アクティビティの状態

	状態	説明
READY 状態	未実行状態 (Unexecution)	アクティビティインスタンスが未実行で、今後実行される可能性のある状態です。
	未省略状態 (Unomission)	アクティビティインスタンスが未実行で、今後実行省略される状態です。
ACTIVE 状態	実行中状態 (Executing)	アクティビティインスタンスが実行中の状態です。
	待ち受け状態 (Awaiting)	受付アクティビティのインスタンスが受け付けられる状態です。
	フォルト中状態 (Faulting)	アクティビティインスタンスが faultHandler でキャッチしたアクティビティを実行している状態です。
ERROR 状態	エラー状態 (Error)	サービス呼び出しアクティビティの処理で例外が発生し、アクティビティインスタンスの処理がエラーで中断した状態です。
COMPLETED 状態	完了状態 (Completed)	アクティビティインスタンスが正常に実行され終了した状態です。
	キャンセル終了状態 (Canceled)	フォルトが発生しアクティビティインスタンスの実行がキャンセルされた状態です。
	フォルト終了状態 (Faulted)	アクティビティインスタンスの実行中にフォルトが発生して終了した状態です。
	実行省略状態 (Omitted)	アクティビティインスタンスの実行が省略された状態です。

各アクティビティの詳細については、マニュアル「Cosminexus サービスプラットフォーム 開発ガイド」の「6.6 アクティビティの定義」を参照してください。

4.2.5 変数の種類と役割

ビジネスプロセスでは、メッセージに含まれる値や業務で使用する値を変数で保持できます。変数にはグローバル変数とローカル変数の2種類があり、開発環境のビジネスプロセス定義画面で定義します。変数の役割を次の表に示します。

表 4-6 変数の役割

変数	説明
グローバル変数	グローバル変数は、一つのプロセスインスタンス内であれば、どこからでも参照できます。また、プロセスが完了しても変数の値はデータベースに記録されているため、ビジネスプロセス全体の進行状況を分析できます。変数名は、プロセス内で一意にします。
ローカル変数	ローカル変数は、宣言したスコープ内で参照できます。宣言しているスコープが終了すると、変数の値もデータベースから削除されるため、プロセス実行中の条件判定など一時的に値を参照するために利用できます。変数名は、スコープ内で一意にします。スコープが異なれば同じ変数名を宣言できます。ただし、変数を参照する場合に、参照個所で最も内側に近いスコープで宣言された変数を参照します。

変数の定義については、マニュアル「Cosminexus サービスプラットフォーム 開発ガイド」の「6.5.1 変数の定義」を参照してください。

4.2.6 相関セットの役割

相関セットはサービスリクエストが送信するメッセージを一意的に識別するための文字列です。相関セットで使用できるメッセージ種別は、XML 形式だけです。

一つのプロセスに対して、サービスリクエストからメッセージを受け取れるプロセスインスタンスが HCSC サーバ上に複数ある場合、HCSC サーバはメッセージに含まれる相関セットの値を利用してプロセスインスタンスを特定します。

サービスリクエストは相関セットを含めた要求電文を作成して、HCSC サーバに対してサービス部品の呼び出し（ビジネスプロセスの呼び出し）を要求します。ビジネスプロセスからの応答にも相関セットが含まれ、どの送信メッセージに対する結果かを識別できます。相関セットは開発環境のビジネスプロセス定義画面で定義します。

なお、相関セットは、ビジネスプロセスがサービス部品を呼び出すとき、およびビジネスプロセスがサービス部品から応答を受けたときにも、新たに相関セットを指定してメッセージを送信できます。

相関セットによるプロセスインスタンスの識別については、マニュアル「Cosminexus サービスプラットフォーム 開発ガイド」の「6.5.2 相関セットの定義」を参照してください。

4.3 永続化するビジネスプロセスと永続化しないビジネスプロセス

ビジネスプロセスには、データベースにプロセスの実行状態および実行履歴を永続化するビジネスプロセスと、永続化しないビジネスプロセスの2種類があります。永続化とは、プロセスの実行状態および実行履歴をデータベースに記録することです。

それぞれのビジネスプロセスの違いについて、次に説明します。

4.3.1 永続化するビジネスプロセス

永続化するビジネスプロセスは、プロセスの実行状態と実行履歴の記録があるため、障害が発生した場合に、プロセスの実行の進捗度がわかり、プロセスを再実行できます。開発環境でビジネスプロセスを追加するときに、永続化する設定にできます。

次に示す内容を実施する場合には、ビジネスプロセスを永続化に設定します。

プロセスインスタンスの実行履歴の管理

- コマンドによるプロセスインスタンスの実行履歴の取得
- 運用環境の画面によるプロセスインスタンスの実行履歴の取得

プロセスインスタンスの再実行

- コマンドによるプロセスインスタンスの再実行
- 運用環境の画面によるプロセスインスタンスの再実行
- サービスリクエスト (SOAP 通信 /SessionBean) によるプロセスインスタンスの再実行

ビジネスプロセスを新規に追加する方法については、マニュアル「Cosminexus サービスプラットフォーム 開発ガイド」の「6.2.1 ビジネスプロセスを新規に追加する」を参照してください。

4.3.2 永続化しないビジネスプロセス

永続化しないビジネスプロセスは、プロセスの実行状態と実行履歴の記録を採らないため、高性能を実現する必要があるビジネスプロセスの場合に設定します。開発環境でビジネスプロセスを追加するときに、永続化しない設定にできます。

ただし、永続化するビジネスプロセスではできることが、永続化しないビジネスプロセスではできないといった制約事項があります。永続化しないビジネスプロセスは、次の条件を満たす場合に定義できます。

1. 非同期の受付アクティビティを含まない。
2. 非同期のサービスアダプタに対して、サービスを呼び出すサービス呼出アクティビティを含まない。
3. 待機アクティビティを含まない。

4. ビジネスプロセスを使った運用

4. 応答アクティビティよりあとにアクティビティ（分岐（終了）アクティビティおよび終了アクティビティを除く）が存在しない。

例を次に示します。

（例1）「スコープ」の中と外で「受付」と「応答」を使用する場合は、スコープ内の「応答1」のあとに、スコープ外の「受付2」を実行することになるため、永続化しないビジネスプロセスとして定義できません。



（例2）「並列処理」の中に「受付」と「応答」を使用する場合は、次に示すように、並列処理が必ず同時に実行して終了するとは限らないため、永続化しないビジネスプロセスとして定義できません。

- 「応答1」を実行したあとに「受付2」、「サービス呼出2」、または「応答2」のどれかを実行する。
- 「応答2」を実行したあとに「受付1」、「サービス呼出1」、または「応答1」のどれかを実行する。



（例3）「並列処理」の中に「サービス呼出」だけを使用する場合は、永続化しないビジネスプロセスとして定義できます。



（例4）「分岐」の中に「応答」を使用する場合は、「応答1」と「応答2」のどちらか一方だけを実行することになるため、永続化しないビジネスプロセスとして定義できません。



5. プロセスインスタンスの再実行を使用しない（障害発生時は，サービスリクエスタからサービス呼び出し要求（ビジネスプロセスの呼び出し）を再度実行する）。

ビジネスプロセスを新規に追加する方法については，マニュアル「Cosminexus サービスプラットフォーム 開発ガイド」の「6.2.1 ビジネスプロセスを新規に追加する」を参照してください。

4.4 ビジネスプロセスのトランザクション

永続化するビジネスプロセスの場合、ビジネスプロセスを実行する過程でアクティビティの状態をデータベースに出力します。

ここでは、アクティビティの状態を確定するタイミング（トランザクションの開始とコミット）について説明します。

トランザクションの開始とコミットのタイミングを次の表に示します。

表 4-7 トランザクションの開始とコミットのタイミング

処理	タイミング
トランザクションの開始	受付アクティビティでメッセージを受信したとき。
	通信モデルが同期の場合のサービス呼出アクティビティで、メッセージを送信する前のトランザクションをコミットした直後。 また、ビジネスプロセスの状態の互換性（HCSC サーバセットアップ定義の bp-status-compatible）に OFF を指定した場合は、サービス呼出アクティビティでメッセージを送信する前のトランザクションをコミットした直後。通信モデルが同期、非同期のどちらでも同じタイミングになります。
	ビジネスプロセスの状態の互換性（HCSC サーバセットアップ定義の bp-status-compatible）に OFF を指定した場合、サービス呼出アクティビティの処理が完了したときのトランザクションをコミットした直後。
	応答アクティビティの処理のあとに存在するアクティビティを実行する直前。
	プロセス再実行ユーティリティによって、プロセスを再実行するとき。
	待機アクティビティに指定された間隔または期限でタイマが解除された直後。
	待機アクティビティの処理が完了したときのトランザクションをコミットした直後。
トランザクションのコミット	応答アクティビティでメッセージを応答する直前。
	通信モデルが同期の場合のサービス呼出アクティビティでメッセージを送信する直前。 また、ビジネスプロセスの状態の互換性（HCSC サーバセットアップ定義の bp-status-compatible）に OFF を指定した場合は、サービス呼出アクティビティでメッセージを送信する直前。通信モデルが同期、非同期のどちらでも同じタイミングになります。
	ビジネスプロセスの状態の互換性（HCSC サーバセットアップ定義の bp-status-compatible）に OFF を指定した場合、サービス呼出アクティビティの処理が完了したとき。
	実行可能なすべてのアクティビティを実行し、ビジネスプロセスの処理を終了するとき。
	待機アクティビティに指定された間隔または期限の J2EE サーバの TimerService にタイマを登録する直前。
	待機アクティビティの処理が完了したとき。

トランザクションの開始とコミットのタイミングは、ビジネスプロセスの状態の互換性（HCSC サーバセットアップ定義の bp-status-compatible）の指定が ON または OFF の場合で動作が異なります。

それぞれのトランザクションの開始とコミットのタイミングの例を次に示します。

4.4.1 ビジネスプロセスの状態の互換性の指定を ON にした場合のトランザクション

ビジネスプロセスの状態の互換性（HCSC サーバセットアップ定義の bp-status-compatible）に ON を指定した場合のトランザクションについて説明します。

なお、ビジネスプロセスの状態の互換性に ON を指定した場合は、Cosminexus サービスプラットフォームのバージョン 08-00 以前と同じ動作となります。

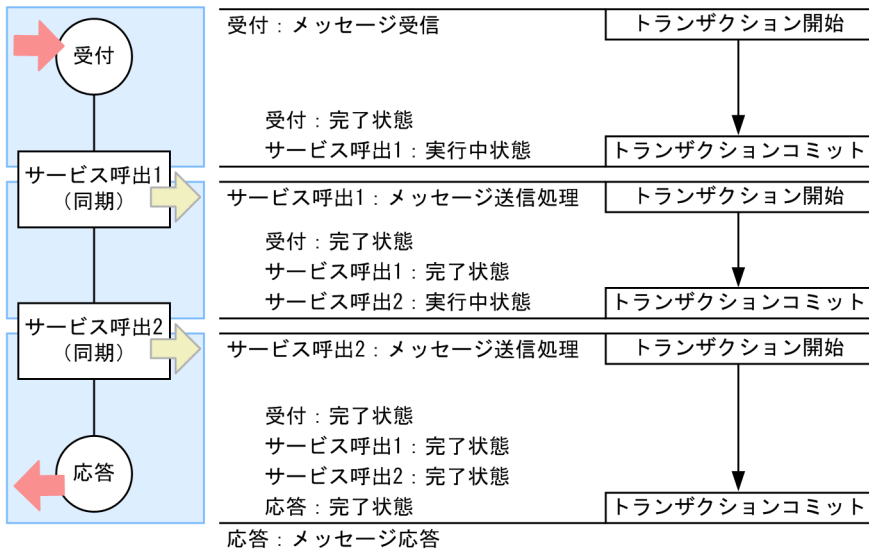
(1) 同期のサービス呼出アクティビティが二つ並んだ場合


正常に処理が進行した場合


それぞれのサービス呼出アクティビティでのメッセージ送信処理直前でトランザクションをコミットし、新たにトランザクションを開始します。

同期のサービス呼出で正常に処理が進行した場合のトランザクションの範囲を次の図に示します。

図 4-15 正常に処理が進行した場合のトランザクションの範囲（同期のサービス呼出）



(凡例)  : ビジネスプロセスの受付・応答

 : サービス部品呼出要求

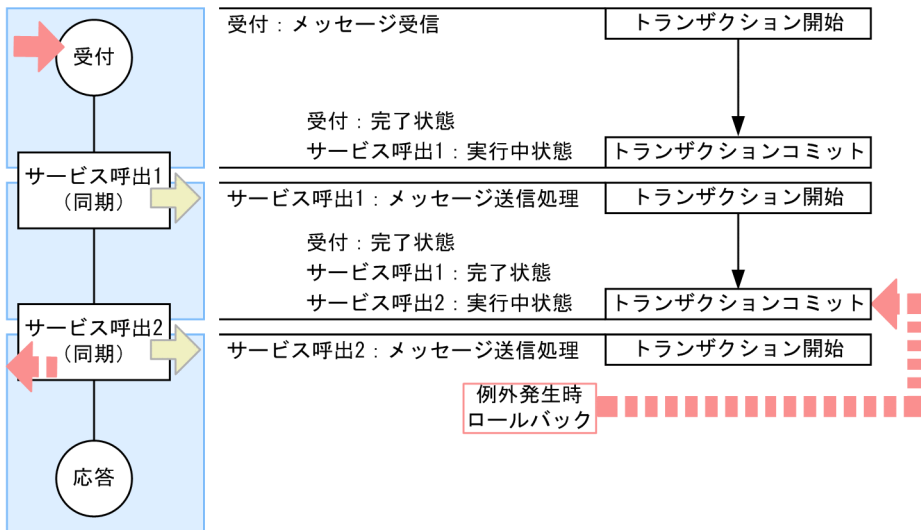
4. ビジネスプロセスを使った運用




例外が発生した場合

サービス呼出2の処理が例外でエラーとなった場合は、トランザクションがコミットした時点（受付：完了状態，サービス呼出1：完了状態，サービス呼出2：実行中状態）までロールバックします。

同期のサービス呼出で例外が発生した場合のトランザクションの範囲を次の図に示します。

図 4-16 例外が発生した場合のトランザクションの範囲（同期のサービス呼出）



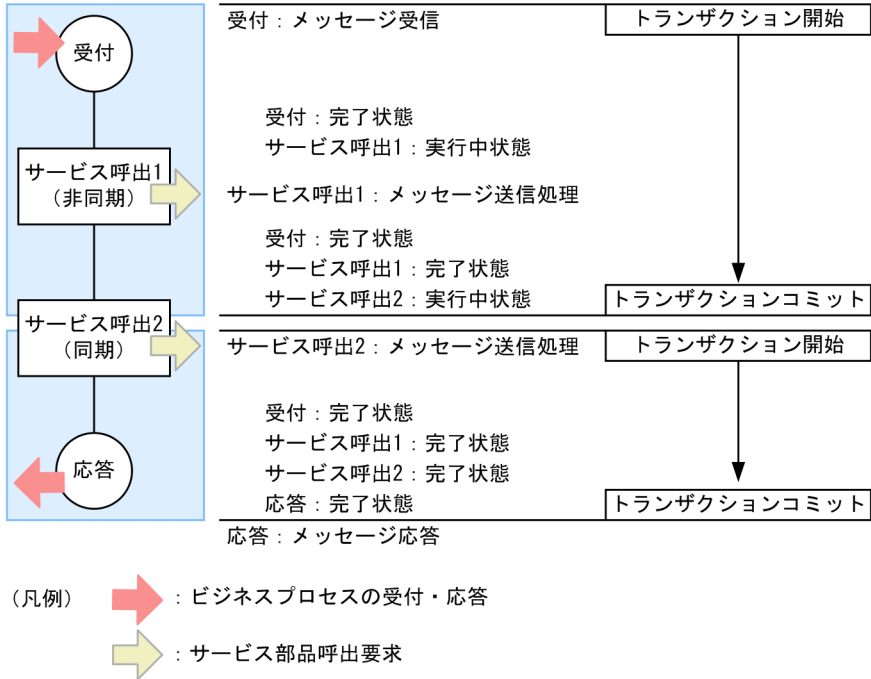
- (凡例)
-  : ビジネスプロセスの受付
 -  : サービス部品呼出要求
 -  : 例外発生

(2) 非同期と同期のサービス呼出アクティビティが二つ並んだ場合

正常に処理が進行した場合

非同期のサービス呼出アクティビティでのメッセージ送信処理ではトランザクションはコミットしないで、同期のサービス呼出アクティビティでのメッセージ送信処理直前でトランザクションをコミットし、新たにトランザクションを開始します。非同期と同期のサービス呼出で正常に処理が進行した場合のトランザクションの範囲を次の図に示します。

図 4-17 正常に処理が進行した場合のトランザクションの範囲（非同期と同期のサービス呼出）



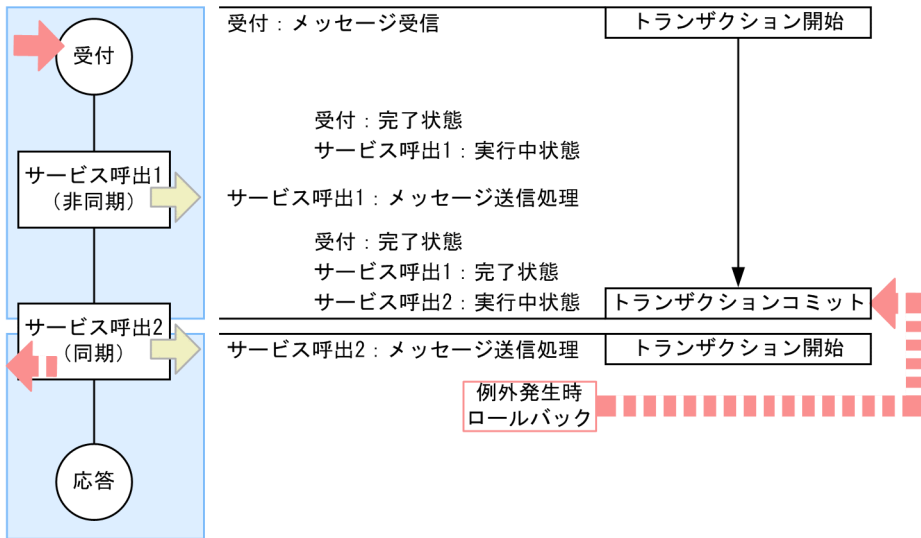
例外が発生した場合（非同期と同期のサービス呼出）

サービス呼出 1 が非同期，サービス呼出 2 が同期の構成で，サービス呼出 2 の処理が例外でエラーとなった場合は，トランザクションがコミットした時点（受付：完了状態，サービス呼出 1：完了状態，サービス呼出 2：実行中状態）までロールバックします。

非同期と同期のサービス呼出で例外が発生した場合のトランザクションの範囲を次の図に示します。

4. ビジネスプロセスを使った運用

図 4-18 例外が発生した場合のトランザクションの範囲（非同期と同期のサービス呼出）



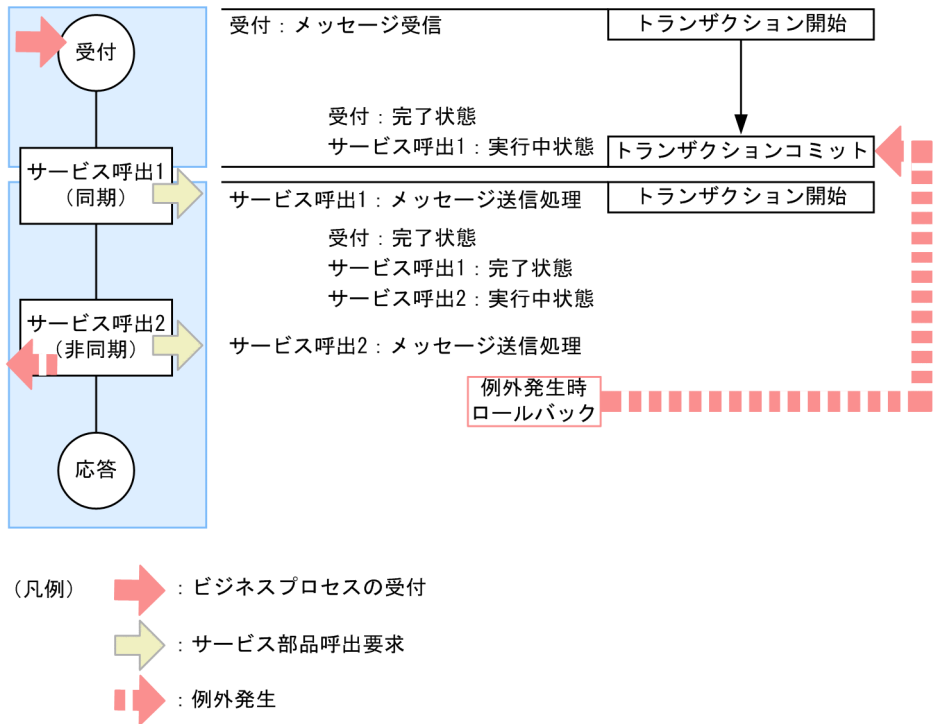
- (凡例)
- : ビジネスプロセスの受付
 - : サービス部品呼出要求
 - : 例外発生

例外が発生した場合（同期と非同期のサービス呼出）

サービス呼出 1 が同期、サービス呼出 2 が非同期の構成で、サービス呼出 2 の処理が例外でエラーとなった場合は、トランザクションがコミットした時点（受付：完了状態、サービス呼出 1：実行中状態）までロールバックします。

同期と非同期のサービス呼出で例外が発生した場合のトランザクションの範囲を次の図に示します。

図 4-19 例外が発生した場合のトランザクションの範囲（同期と非同期のサービス呼出）



(3) 非同期のサービス呼出アクティビティが二つ並んだ場合

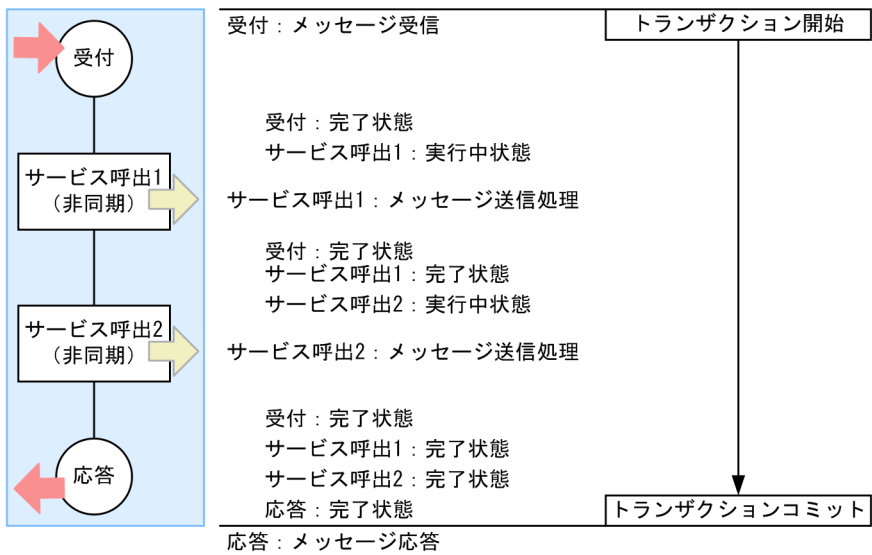
正常に処理が進行した場合

非同期のサービス呼出アクティビティでのメッセージ送信処理ではトランザクションはコミットしません。そのため、ビジネスプロセスが応答するときにトランザクションがコミットします。

非同期のサービス呼出で正常に処理が進行した場合のトランザクションの範囲を次の図に示します。

4. ビジネスプロセスを使った運用

図 4-20 正常に処理が進行した場合のトランザクションの範囲（非同期のサービス呼出）

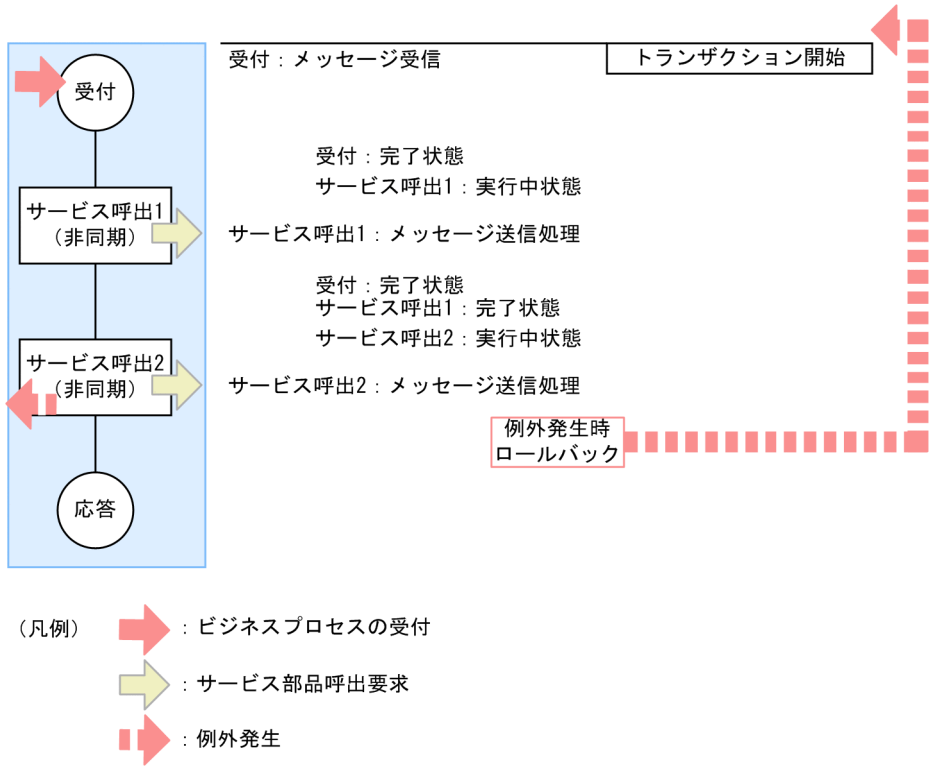


例外が発生した場合

サービス呼出2の処理が例外でエラーとなった場合は、トランザクションがコミットした時点（受付：未実行状態，サービス呼出1：未実行状態，サービス呼出2：未実行状態）までロールバックします。

非同期のサービス呼出で例外が発生した場合のトランザクションの範囲を次の図に示します。

図 4-21 例外が発生した場合のトランザクションの範囲（非同期のサービス呼出）



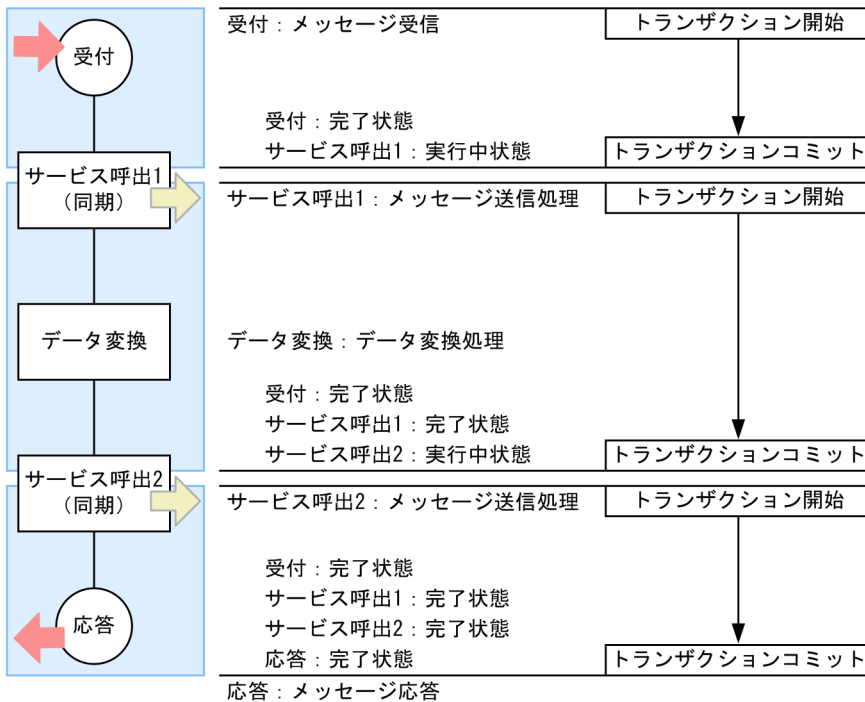
(4) 同期のサービス呼出アクティビティのあとにデータ変換アクティビティがある場合



正常に処理が進行した場合

それぞれのサービス呼出アクティビティでのメッセージ送信処理直前でトランザクションをコミットしますが、データ変換アクティビティではトランザクションがコミットしません。これは、表 4-7 で示した以外のアクティビティも同様です。データ変換などのアクティビティがある場合で正常に処理が進行したときのトランザクションの範囲を次の図に示します。

4. ビジネスプロセスを使った運用

図 4-22 正常に処理が進行した場合のトランザクションの範囲（データ変換などのアクティビティがある場合）



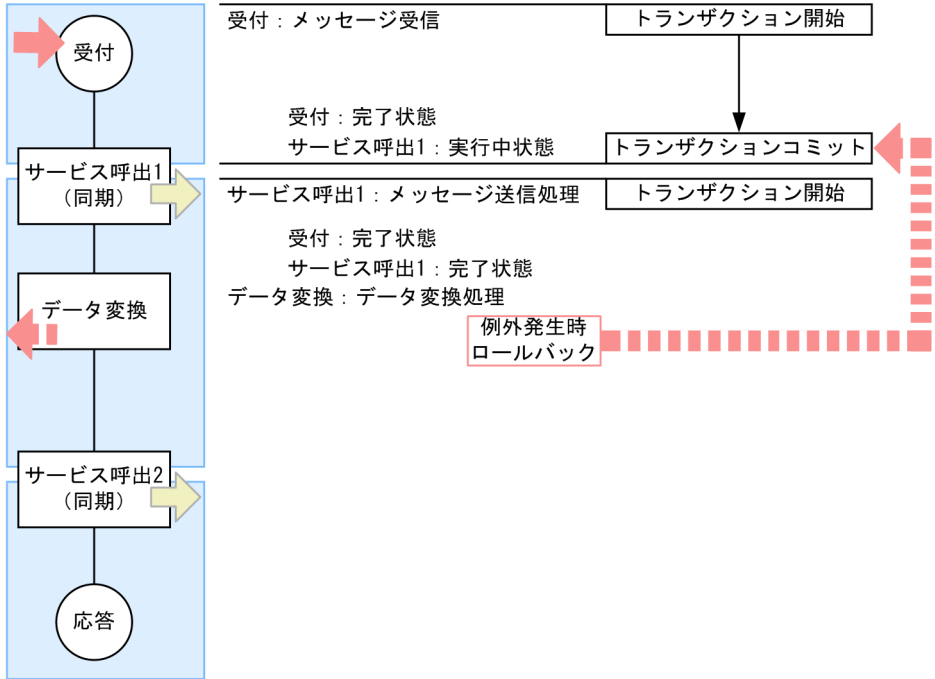
(凡例)  : ビジネスプロセスの受付・応答
 : サービス部品呼出要求

例外が発生した場合（同期サービス呼出のあとにデータ変換などのアクティビティがある場合）

サービス呼出 1 が同期の構成で、サービス呼出 1 のあとにデータ変換処理でエラーとなった場合は、トランザクションがコミットした時点（受付：完了状態、サービス呼出 1：実行中状態）までロールバックします。

同期サービス呼出のあとにデータ変換などのアクティビティがある場合で例外が発生したときのトランザクションの範囲を次の図に示します。

図 4-23 例外が発生した場合のトランザクションの範囲（同期サービス呼出のあとにデータ変換などのアクティビティがある場合）



- (凡例)
- : ビジネスプロセスの受付
 - : サービス部品呼出要求
 - : 例外発生

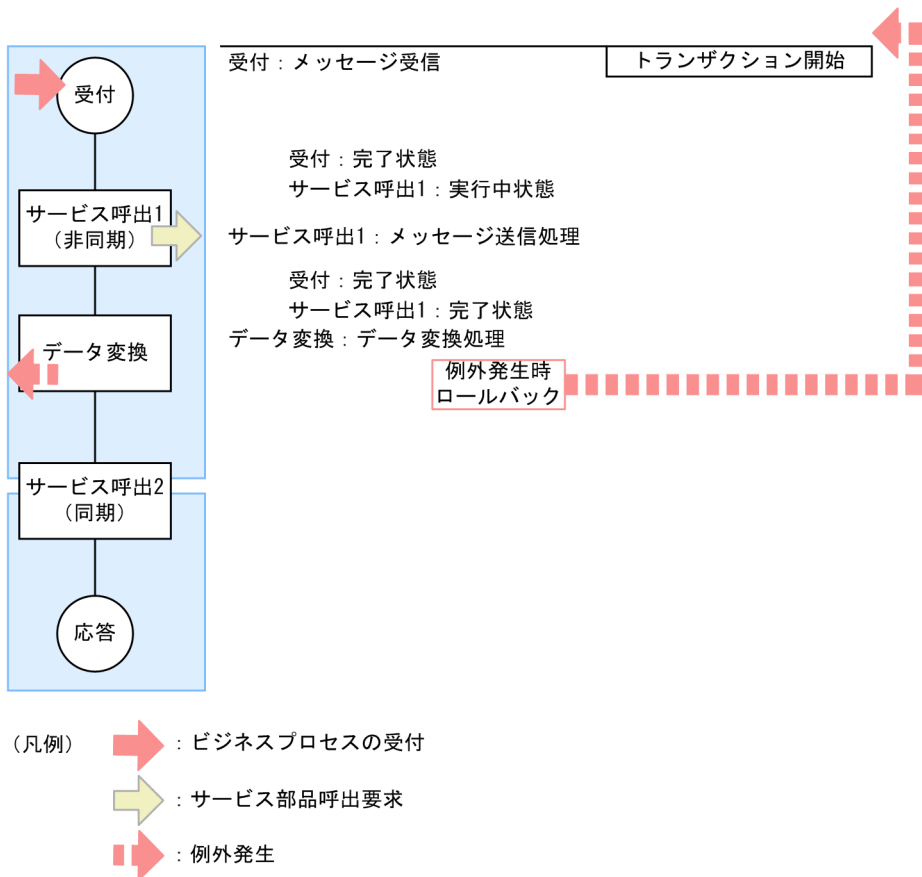
例外が発生した場合（非同期サービス呼出のあとにデータ変換などのアクティビティがある場合）

サービス呼出 1 が非同期の構成で、サービス呼出 1 のあとにデータ変換処理でエラーとなった場合は、トランザクションがコミットした時点（受付：未実行状態，サービス呼出 1：未実行状態）までロールバックします。

非同期サービス呼出のあとにデータ変換などのアクティビティがある場合で例外が発生したときのトランザクションの範囲を次の図に示します。

4. ビジネスプロセスを使った運用

図 4-24 例外が発生した場合のトランザクションの範囲（非同期サービス呼出のあとにデータ変換などのアクティビティがある場合）



4.4.2 ビジネスプロセスの状態の互換性の指定を OFF にした場合のトランザクション

ビジネスプロセスの状態の互換性（HCSC サーバセットアップ定義の bp-status-compatible）に OFF を指定した場合のトランザクションについて説明します。

ビジネスプロセスの状態の互換性に OFF を指定した場合は、サービス呼出アクティビティの処理が完了したタイミングでトランザクションを終了します。ビジネスプロセスのサービス呼出アクティビティ実行中に例外が発生した場合、プロセスインスタンスおよびサービス呼出アクティビティの状態にエラー発生（Error）を設定します。

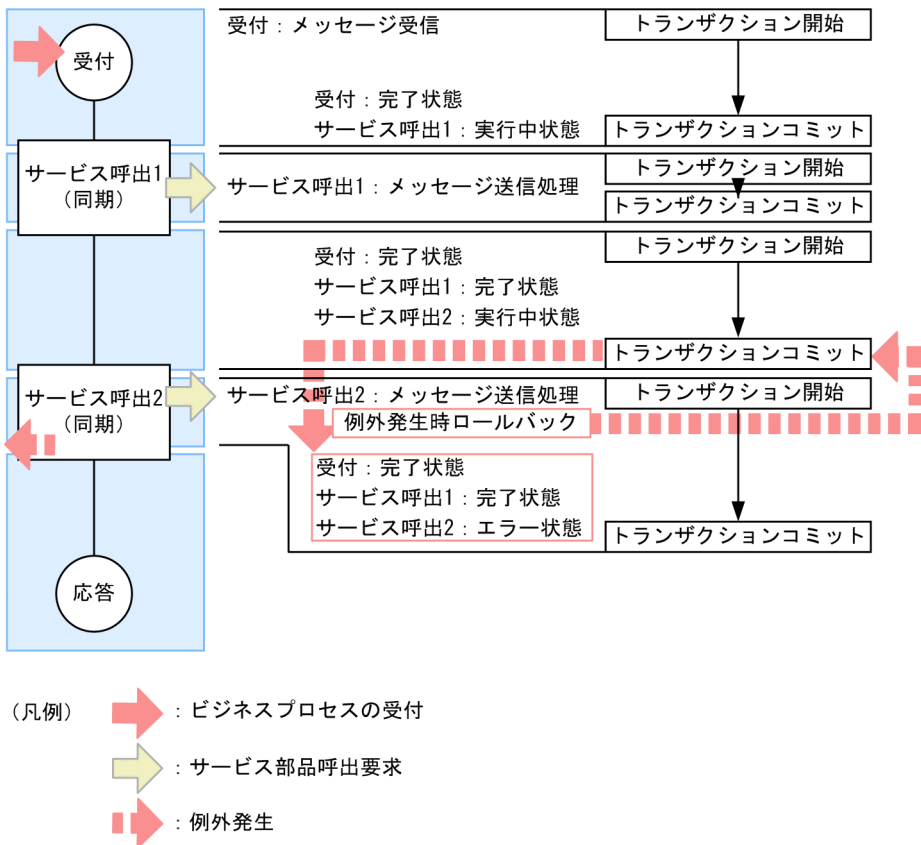
(1) 同期のサービス呼出アクティビティが二つ並んだ場合

正常に処理が進行した場合

それぞれのサービス呼出アクティビティでのメッセージ送信処理直前でトランザク

4. ビジネスプロセスを使った運用

図 4-26 例外が発生した場合のトランザクションの範囲（同期のサービス呼出）



(2) 非同期と同期のサービス呼出アクティビティが二つ並んだ場合

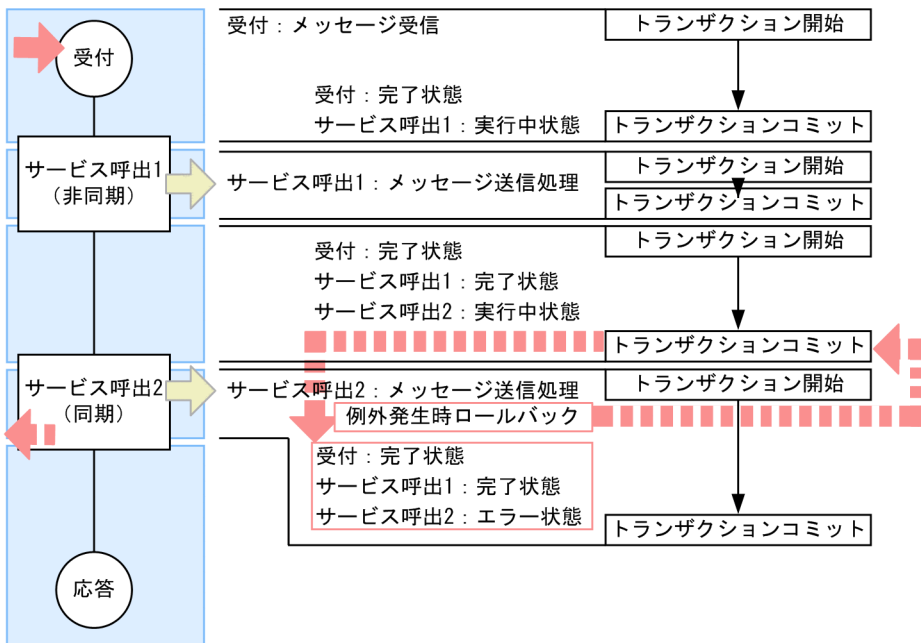
正常に処理が進行した場合

それぞれのサービス呼出アクティビティでのメッセージ送信処理直前でトランザクションをコミットし、新たにトランザクションを開始します。また、メッセージ送信処理が完了したあともトランザクションをコミットし、新たにトランザクションを開始します。

非同期と同期のサービス呼出で正常に処理が進行した場合のトランザクションの範囲を次の図に示します。

4. ビジネスプロセスを使った運用

図 4-28 例外が発生した場合のトランザクションの範囲（非同期と同期のサービス呼出）



- (凡例)
- ➡ : ビジネスプロセスの受付
 - ➡ : サービス部品呼出要求
 - - - ➡ : 例外発生

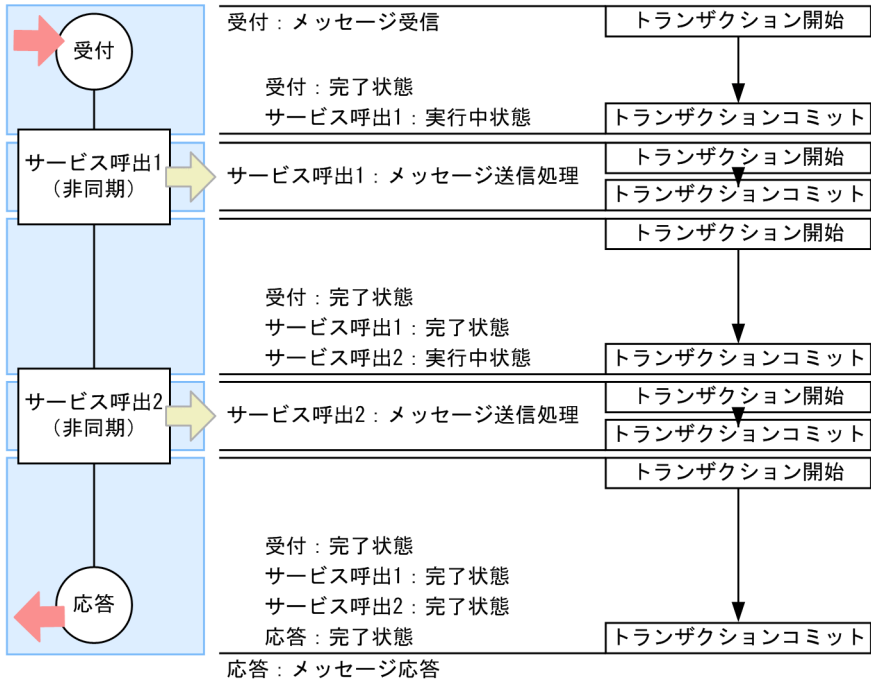
(3) 非同期のサービス呼出アクティビティが二つ並んだ場合

正常に処理が進行した場合

それぞれのサービス呼出アクティビティでのメッセージ送信処理直前でトランザクションをコミットし、新たにトランザクションを開始します。また、メッセージ送信処理が完了したあともトランザクションをコミットし、新たにトランザクションを開始します。

非同期のサービス呼出で正常に処理が進行した場合のトランザクションの範囲を次の図に示します。

図 4-29 正常に処理が進行した場合のトランザクションの範囲（非同期のサービス呼出）



(凡例) : ビジネスプロセスの受付・応答
 : サービス部品呼出要求

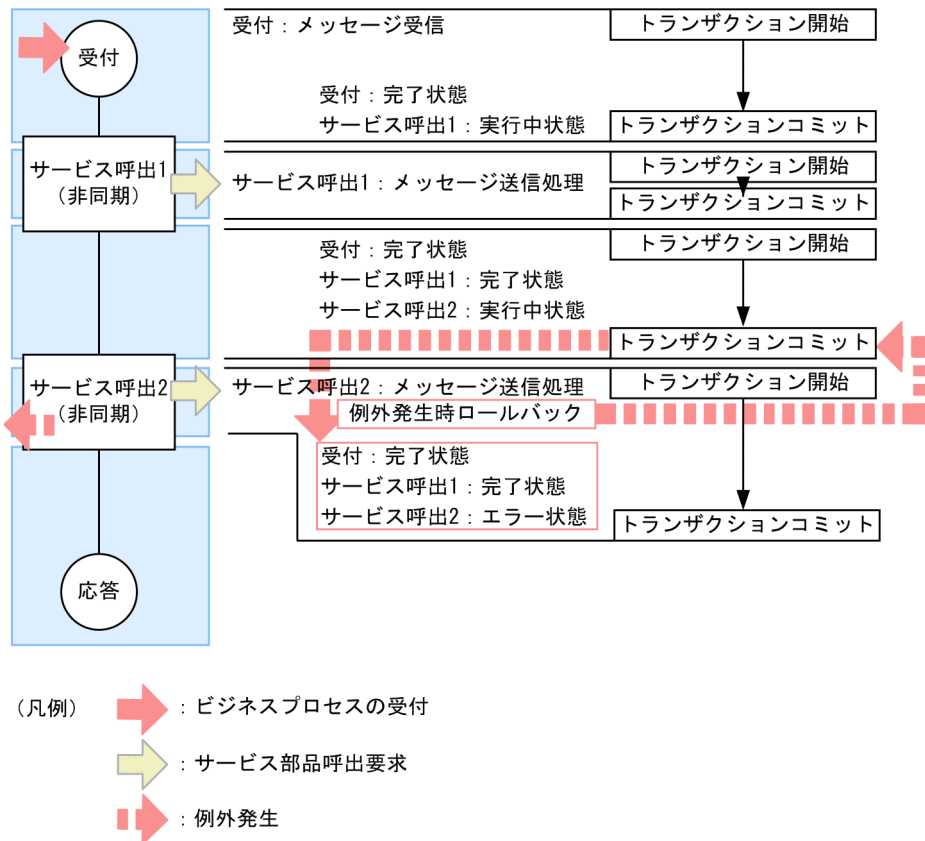
例外が発生した場合

サービス呼出2の処理が例外でエラーとなった場合は、トランザクションがコミットした時点（受付：完了状態，サービス呼出1：完了状態，サービス呼出2：実行中状態）にロールバックしたあと、プロセスインスタンスおよびサービス呼出アクティビティの状態にエラー発生（Error）を設定して、トランザクションをコミットします。

非同期のサービス呼出で例外が発生した場合のトランザクションの範囲を次の図に示します。

4. ビジネスプロセスを使った運用

図 4-30 例外が発生した場合のトランザクションの範囲（非同期のサービス呼出）



(4) 同期のサービス呼出アクティビティのあとにデータ変換アクティビティがある場合

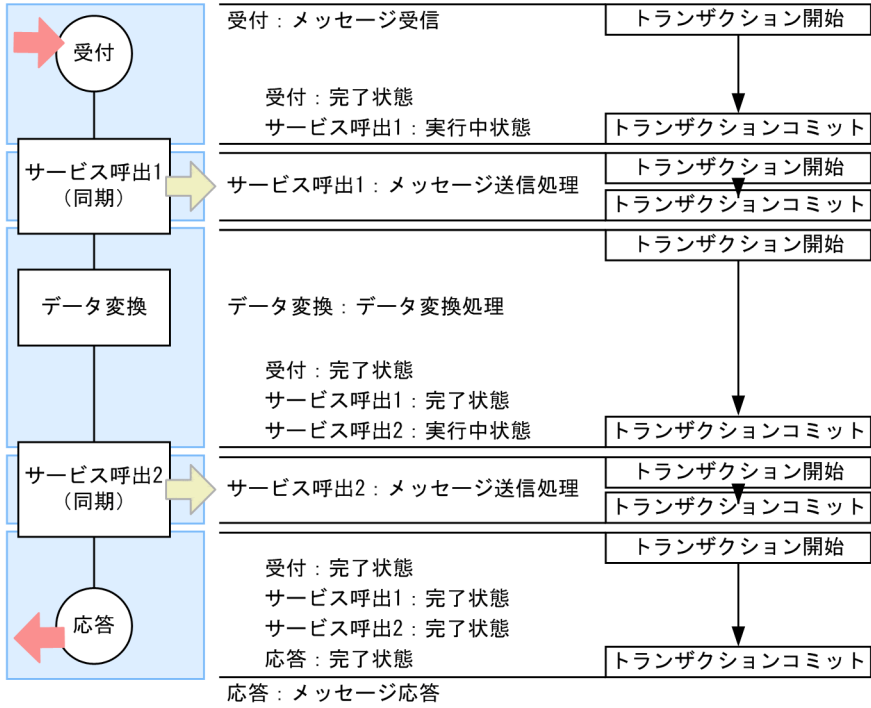
正常に処理が進行した場合



それぞれのサービス呼出アクティビティでのメッセージ送信処理直前でトランザクションをコミットし、新たにトランザクションを開始します。また、メッセージ送信処理が完了したあとにもトランザクションをコミットし、新たにトランザクションを開始します。

ただし、データ変換アクティビティではトランザクションがコミットしません。これは、表 4-7 で示した以外のアクティビティも同様です。

データ変換などのアクティビティがある場合で正常に処理が進行したときのトランザクションの範囲を次の図に示します。

図 4-31 正常に処理が進行した場合のトランザクションの範囲（データ変換などのアクティビティがある場合）



(凡例)  : ビジネスプロセスの受付・応答
 : サービス部品呼出要求

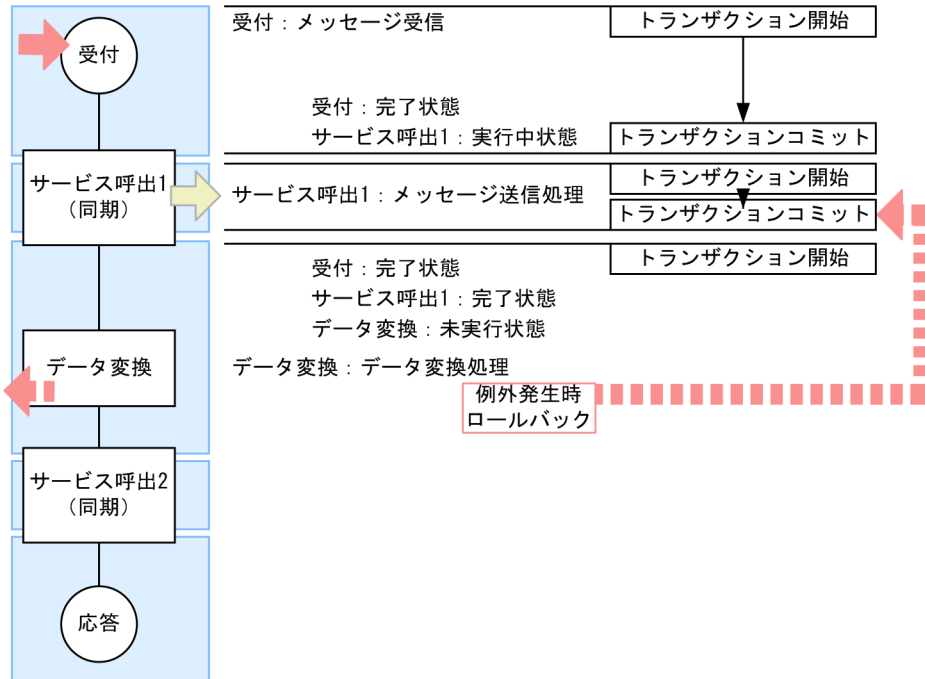
例外が発生した場合（データ変換などのアクティビティがある場合）




サービス呼出 1 のあとのデータ変換処理でエラーとなった場合は、トランザクションがコミットした時点（受付：完了状態，サービス呼出 1：完了状態，データ変換：未実行状態）までロールバックします。

データ変換などのアクティビティがある場合で例外が発生したときのトランザクションの範囲を次の図に示します。

4. ビジネスプロセスを使った運用

図 4-32 例外が発生した場合のトランザクションの範囲（データ変換などのアクティビティがある場合）



- (凡例)
-  : ビジネスプロセスの受付
 -  : サービス部品呼出要求
 -  : 例外発生

4.5 ビジネスプロセスの再実行

ビジネスプロセス実行中に、実行基盤のあるマシンが停止したり、サービス部品の呼び出しに失敗したりした場合、障害が発生して中断したプロセスインスタンスを再実行できます。プロセスインスタンスを再実行することで、ビジネスプロセスの内容が処理できます。プロセスインスタンスを再実行できる条件を次に示します。

- プロセスインスタンスの状態がエラー（Error）または実行中（Executing）であり、かつサービス呼出アクティビティの状態がエラー（Error）または実行中（Executing）の場合
- プロセスインスタンスの状態が実行中（Executing）であり、かつ待機アクティビティの状態が実行中（Executing）の場合

4.5.1 運用環境の画面からの再実行

運用環境の画面から、中断したプロセスインスタンスを一件ずつ再実行できます。

画面によるプロセスインスタンスの再実行の手順については、マニュアル「Cosminexus サービスプラットフォームシステム構築・運用ガイド」の「6.1.5 プロセスインスタンスを再実行する」を参照してください。

4.5.2 コマンドからの再実行

コマンドを使用して、中断したプロセスインスタンスを一件ずつ再実行できます。

また、複数のプロセスインスタンスが同一のビジネスプロセス定義名である場合、複数のプロセスインスタンスを一括して再実行したり、一つの HCSC サーバ上にある複数のプロセスインスタンスを一括して再実行したりできます。プロセスインスタンスの実行履歴の一括再実行は、コマンドでだけ実行できます。

コマンドによるプロセスインスタンスの再実行の手順については、マニュアル「Cosminexus サービスプラットフォームシステム構築・運用ガイド」の「6.1.5 プロセスインスタンスを再実行する」を参照してください。

4.5.3 サービスリクエストからの再実行（SOAP 通信）

サービス部品の呼び出し処理で例外になって中断したプロセスインスタンスは、システム運用管理者だけでなく、サービスリクエストからも再実行できます。

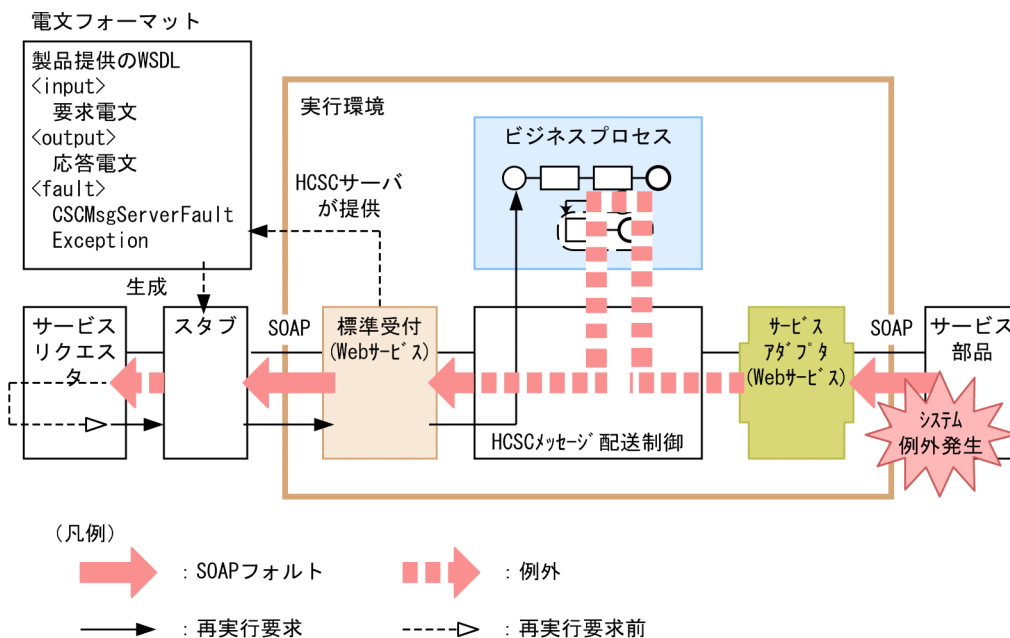
サービスリクエストから再実行するには、標準の同期受付（Web サービス）に再実行要求を出すサービスリクエストを作成します。

サービス部品の呼び出しがエラーになり、ビジネスプロセスインスタンス ID がエラー情報として返ってきた場合は、そのビジネスプロセスインスタンス ID を指定して再実行し

4. ビジネスプロセスを使った運用

まず、SOAP 通信でのサービスリクエストからの要求の再実行を次の図に示します。

図 4-33 SOAP 通信でのサービスリクエストからの要求の再実行



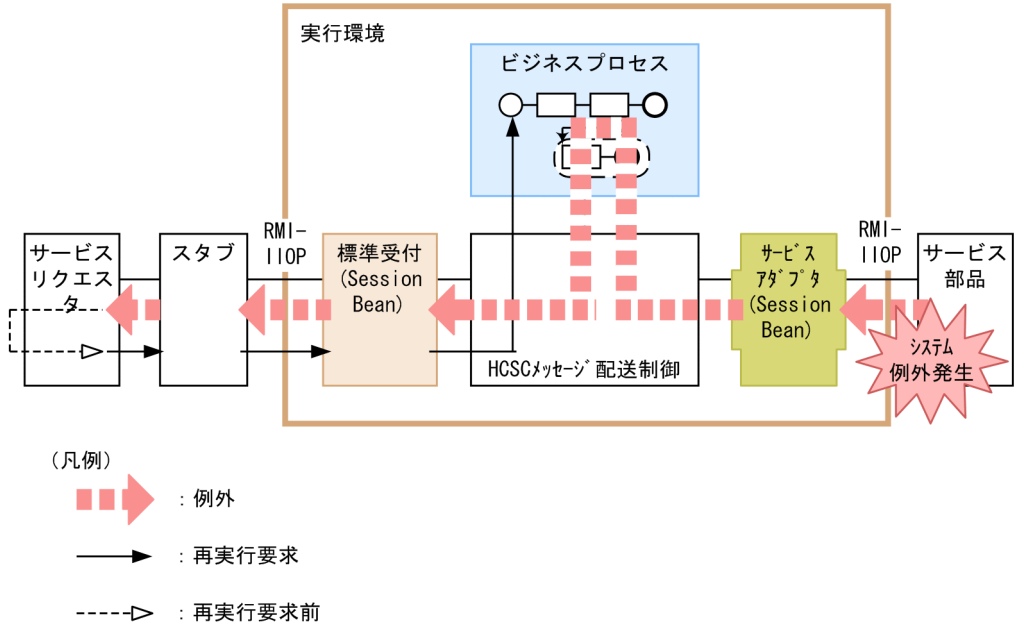
ビジネスプロセスの再実行を要求するサービスリクエストの作成については、マニュアル「Cosminexus サービスプラットフォーム 開発ガイド」の「10.2.9 ビジネスプロセスの再実行を要求するサービスリクエストの作成 (Web サービス・SOAP 通信基盤)」を参照してください。

4.5.4 サービスリクエストからの再実行 (SessionBean)

標準の同期受付 (SessionBean) に要求を出すサービスリクエストからも、ビジネスプロセスの再実行要求ができます。サービスリクエストから再実行するには、標準の同期受付 (SessionBean) に要求を出すサービスリクエストを作成します。

サービス部品呼び出しがエラーになり、ビジネスプロセスインスタンス ID が例外 (エラー情報) として返ってきた場合は、そのビジネスプロセスインスタンス ID を指定して再実行します。SessionBean でのサービスリクエストからのビジネスプロセスの再実行を次の図に示します。

図 4-34 SessionBean でのサービスリクエストからのビジネスプロセスの再実行

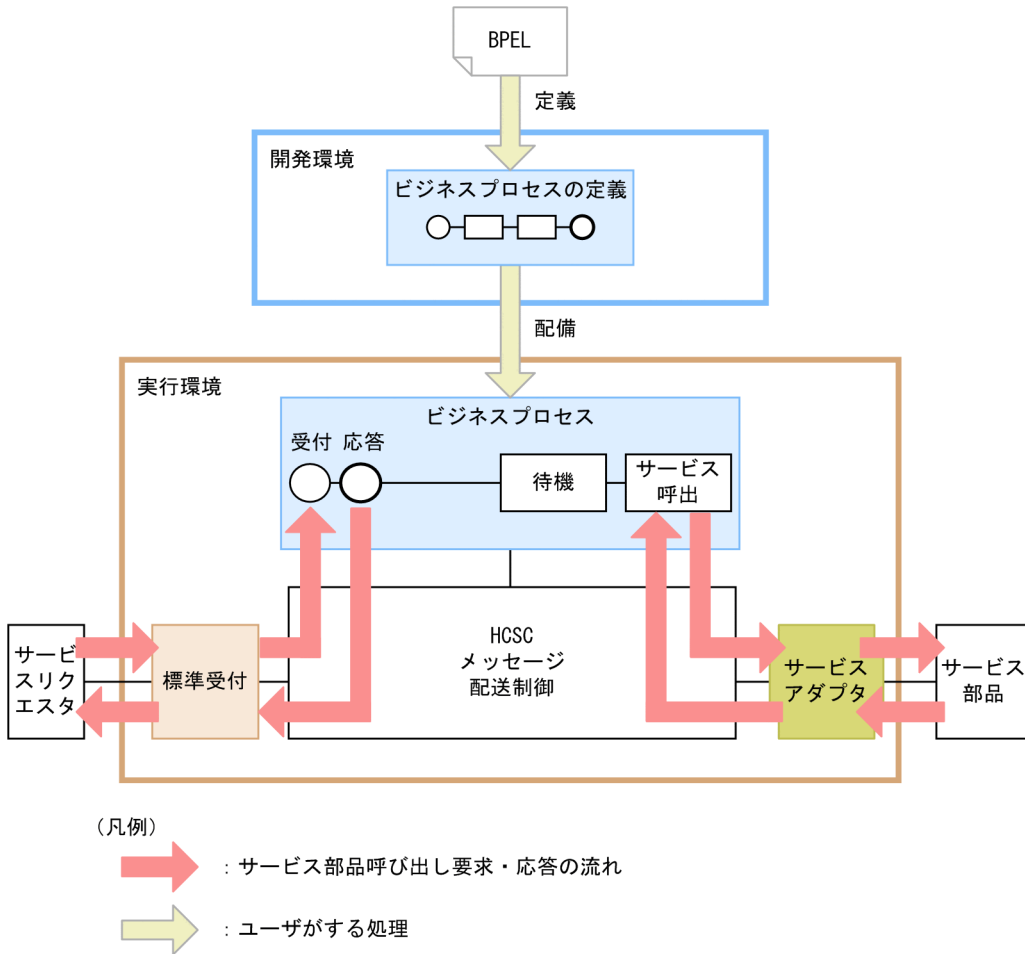


ビジネスプロセスの再実行を要求するサービスリクエストの作成については、マニュアル「Cosminexus サービスプラットフォーム 開発ガイド」の「10.4.8 ビジネスプロセスの再実行を要求するサービスリクエストの作成 (SessionBean)」を参照してください。

4.6 ビジネスプロセスを使用した待機処理

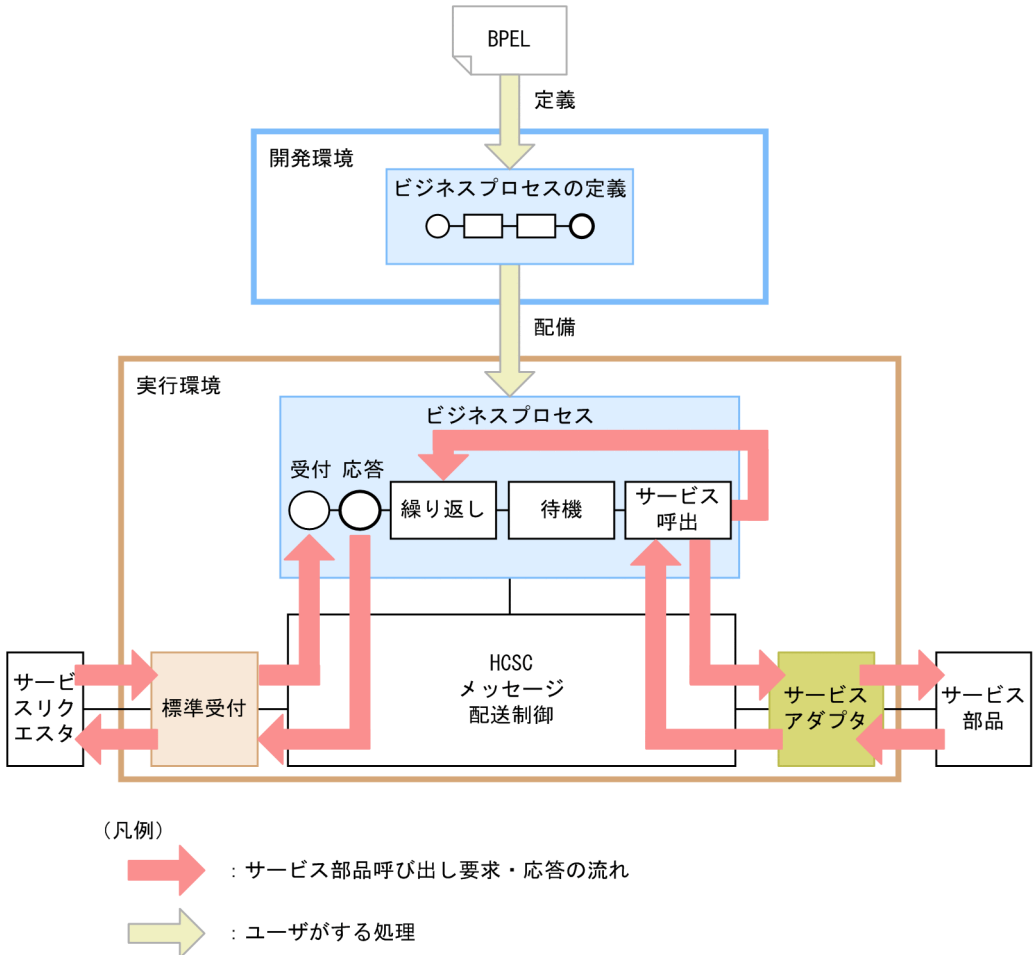
待機アクティビティを定義してビジネスプロセス実行すると、特定の期限に達するまでビジネスプロセスの処理を遅延させることができます。待機アクティビティを定義したビジネスプロセスの待機処理を次の図に示します。

図 4-35 待機アクティビティを定義したビジネスプロセスの待機処理



繰り返しアクティビティの中に待機アクティビティを定義することで、日次処理などの一定間隔での処理を実現することもできます。繰り返しアクティビティの中に待機アクティビティを定義した場合の、ビジネスプロセスの待機処理の図を次に示します。

図 4-36 繰り返しアクティビティの中に待機アクティビティを定義したビジネスプロセスの待機処理



待機アクティビティの定義について次に説明します。

(1) 待機アクティビティでの待機時間の指定

待機アクティビティでは、待機時間として次のどちらかを指定します。

- 待機する間隔
- 待機を解除する期限

どちらの場合も XPath 式を使用して待機時間を指定します。XPath 式を使用すると、ビジネスプロセス内を流れる変数 (XML 電文) 内の時刻に従って動作するため、待機時間を動的に指定できます。また、XPath 式の代わりに、待機する間隔または待機を解除する期限を直接指定することもできます。待機する間隔は XML Schema 型 duration、待機を解除する期限は XML Schema 型 dateTime で指定します。詳細については、マニユ

4. ビジネスプロセスを使った運用

アル「Cosminexus サービスプラットフォーム 開発ガイド」の「6.6.10 待機アクティビティ」を参照してください。

(2) 待機アクティビティを実行するための条件

待機アクティビティは、永続化しないビジネスプロセスでは使用できない、応答アクティビティのあとに設定しなければならないなど、幾つかの条件があります。待機アクティビティを実行するための条件については、マニュアル「Cosminexus サービスプラットフォーム 開発ガイド」の「6.6.10 待機アクティビティ」を参照してください。

(3) 待機アクティビティに関する多重度

待機アクティビティを使用する上で性能を考えた場合、待機アクティビティに関する多重度を見直す必要があります。

待機アクティビティで指定した時間に待機処理を解除するプロセスインスタンスが複数存在している場合で、そのプロセスインスタンスの数より J2EE サーバの TimerService でタイムアウトメソッドをコールバックする最大スレッド数が小さいときは、性能上のボトルネックになることがあります。このような場合、J2EE サーバの環境設定で、タイムアウトメソッドのコールバックの最大スレッド数を設定します。

TimerService でタイムアウトメソッドをコールバックする最大スレッド数は、運用管理ポータルの [論理サーバの環境設定] で設定します。EJB コンテナの設定画面を開き、「タイムアウトメソッドのコールバックの最大スレッド数」の値を、ビジネスプロセスのインスタンス数に合わせて変更してください。設定方法の詳細については、マニュアル「Cosminexus アプリケーションサーバ 運用管理ポータル操作ガイド」の「10.9.3 EJB コンテナの設定」を参照してください。

注 J2EE サーバ用ユーザプロパティファイル (usrconf.properties) の ejbserver.ejb.timerservice.maxCallbackThreads キーに対応します。

5

システム動作を円滑にするための機能

この章では、システム動作を円滑にするための機能について説明します。

5.1 HCSC サーバに関連するタイマの設定

5.2 サービス部品の接続先変更

5.3 SOAP Fault 動作定義ファイルによるフォルトのキャッチ

5.4 アプリケーションからの稼働状態の確認

5.5 XML 電文内の属性値に指定した名前空間接頭辞の補完

5.6 任意形式 (any 形式) の電文フォーマットの定義

5.1 HCSC サーバに関連するタイマの設定

同期のサービス部品呼び出しの場合、応答が返って来ない場合のタイムアウト値を設定することで、無応答でハングアップすることを回避できます。また、サービス部品での実際の処理に掛かる最大時間を考慮して、タイマを設定する必要があります。

5.1.1 HCSC サーバで設定できるタイマ

HCSC サーバで設定できるタイマについて次の図に示します。

図 5-1 設定できるタイマ

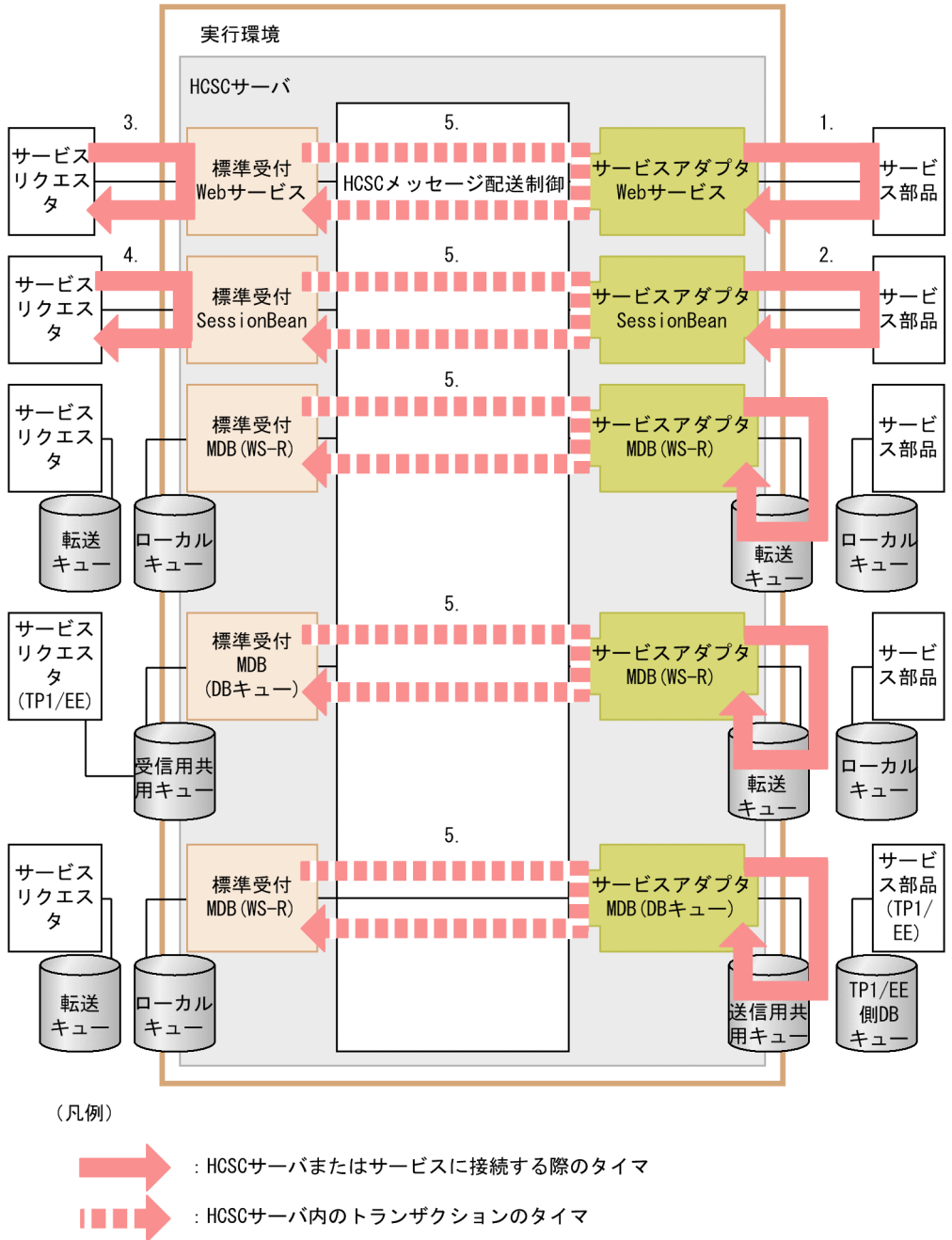


図 5-1 の各項番は次のタイマを示しています。

1. サービス部品に接続するときのタイマ (Web サービス)
2. サービス部品に接続するときのタイマ (SessionBean)

5. システム動作を円滑にするための機能

3. HCSC サーバに接続するときのタイマ (Web サービス)
4. HCSC サーバに接続するときのタイマ (SessionBean)
5. HCSC サーバ内のトランザクションのタイマ

タイマは接続先やプロトコルによって設定方法が異なります。

(1) サービス部品に接続するときのタイマ (Web サービス)

サービス部品 (Web サービス) に接続する場合、書き込み、読み込み、および接続時のタイマを設定できます。

- 書き込み時のタイマ
HTTP および HTTPS (TCP/IP 通信) で使用するソケットに対する write (データ送信) 処理でのタイムアウト値を設定します。接続確立後のデータ送信が完了しない場合などで有効となります。
- 読み込み時のタイマ
HTTP および HTTPS (TCP/IP 通信) で使用するソケットに対する read (データ受信) 処理でのタイムアウト値を設定します。接続確立後のデータ受信が完了しない場合などで有効となります。
- 接続時のタイマ
HTTP および HTTPS (TCP/IP 通信) で使用するソケットに対する connect (ソケット接続) 処理でのタイムアウト値を設定します。接続確立時の要求に対する応答がない場合などで有効となります。

タイムアウト値の設定は、SOAP 通信基盤を利用する場合と、JAX-WS エンジンを利用する場合で異なります。それぞれの場合の設定方法を説明します。

(a) タイムアウト値の設定 (SOAP 通信基盤)

SOAP 通信基盤を利用する場合のタイムアウト値の設定方法について説明します。タイムアウト値は、HCSC サーバ全体またはサービスアダプタ個別に設定できます。

- HCSC サーバ全体の設定
タイムアウト値は、HCSC サーバ稼働マシンの SOAP 通信基盤のサーバ定義ファイルで定義します。サーバ定義ファイルの設定については、マニュアル「Cosminexus アプリケーションサーバ SOAP アプリケーション開発の手引」の「10.2 サーバ定義ファイルの設定」を参照してください。

表 5-1 HCSC サーバ全体のタイマ (Web サービス) の設定 (SOAP 通信基盤)

値	キー名称	デフォルト値 (秒)
サーバ兼クライアントのソケットの書き込みタイムアウト値	c4web.application.<識別子>.socket_write_timeout	60

値	キー名称	デフォルト値 (秒)
サーバ兼クライアントのソケットの読み込みタイムアウト値	c4web.application.<識別子>.socket_read_timeout	300
サーバ兼クライアントのソケットの接続タイムアウト値	c4web.application.<識別子>.socket_connect_timeout	60

- サービスアダプタ個別の設定

開発環境で設定する場合、サービスアダプタ定義画面のクライアント定義ファイルに指定します。クライアント定義ファイルの設定については、マニュアル

「Cosminexus アプリケーションサーバ SOAP アプリケーション開発の手引」の「10.3 クライアント定義ファイルの設定」を参照してください。

表 5-2 サービスアダプタ個別のタイマ (Web サービス) の設定 (SOAP 通信基盤)

値	キー名称	デフォルト値 (秒)
クライアントのソケットの書き込みタイムアウト値	c4web.application.socket_write_timeout	60
クライアントのソケットの読み込みタイムアウト値	c4web.application.socket_read_timeout	300
クライアントのソケットの接続タイムアウト値	c4web.application.socket_connect_timeout	60

(b) タイムアウト値の変更 (SOAP 通信基盤)

すでに HCSC サーバに配備しているサービスアダプタに対して運用環境で定義する場合は、cscsvectl コマンドを使用して、サービス部品呼び出しの通信タイムアウト値を変更します。変更方法の詳細については、マニュアル「Cosminexus サービスプラットフォーム システム構築・運用ガイド」の「5.2.25 サービス部品呼び出しの通信タイムアウト値を変更する」を参照してください。cscsvectl コマンドの使い方については、マニュアル「Cosminexus サービスプラットフォーム リファレンス」の「cscsvectl (サービス情報の管理)」を参照してください。

表 5-3 タイマ (Web サービス) の設定値の変更 (SOAP 通信基盤)

値	キー名称
書き込みタイムアウト値	<クラスタ名>.<サービス ID>.WebService.c4web.application.socket_write_timeout
読み込みタイムアウト値	<クラスタ名>.<サービス ID>.WebService.c4web.application.socket_read_timeout

5. システム動作を円滑にするための機能

値	キー名称
接続タイムアウト値	<クラスタ名>.<サービスID>.WebService.c4web.application.socket_connect_timeout

(c) タイムアウト値の設定 (JAX-WS エンジン)

JAX-WS エンジンを利用する場合のタイムアウト値の設定方法について説明します。タイムアウト値は、HCSC サーバ全体またはサービスアダプタ個別に設定できます。

- HCSC サーバ全体の設定
 HCSC サーバ稼働マシンの JAX-WS エンジンの共通定義ファイルで定義します。共通定義ファイルの設定については、マニュアル「Cosminexus アプリケーションサーバ Web サービス開発の手引」の「10.1.2 共通定義ファイルの設定項目」を参照してください。

表 5-4 HCSC サーバ全体のタイマ (Web サービス) の設定 (JAX-WS エンジン)

値	キー名称	デフォルト値 (ミリ秒)
サーバ兼クライアントのソケットの読み込みタイムアウト値	com.cosminexus.jaxws.request.timeout	300000
サーバ兼クライアントのソケットの接続タイムアウト値	com.cosminexus.jaxws.connect.timeout	60000

- サービスアダプタ個別の設定
 開発環境で設定する場合、サービスアダプタ定義画面のクライアント定義ファイルに指定します。クライアント定義ファイルの設定については、マニュアル「Cosminexus サービスプラットフォーム リファレンス」の「クライアント定義ファイル」を参照してください。

表 5-5 サービスアダプタ個別のタイマ (Web サービス) の設定 (JAX-WS エンジン)

値	キー名称	デフォルト値 (秒)
クライアントのソケットの読み込みタイムアウト値	com.cosminexus.csc.request.timeout	300
クライアントのソケットの接続タイムアウト値	com.cosminexus.csc.connect.timeout	60

(d) タイムアウト値の変更 (JAX-WS エンジン)

すでに HCSC サーバに配備しているサービスアダプタに対して運用環境で定義する場合

は、`cscsvctl` コマンドを使用して、サービス部品呼び出しの通信タイムアウト値を変更します。変更方法の詳細については、マニュアル「Cosminexus サービスプラットフォーム システム構築・運用ガイド」の「5.2.25 サービス部品呼び出しの通信タイムアウト値を変更する」を参照してください。`cscsvctl` コマンドの使い方については、マニュアル「Cosminexus サービスプラットフォーム リファレンス」の「`cscsvctl` (サービス情報の管理)」を参照してください。

表 5-6 タイマ (Web サービス) の設定値の変更 (JAX-WS エンジン)

値	キー名称
読み込みタイムアウト値	<クラスタ名>.<サービスID>.WebService.com.cosminexus.csc.request.timeout
接続タイムアウト値	<クラスタ名>.<サービスID>.WebService.com.cosminexus.csc.connect.timeout

(2) サービス部品に接続するときのタイマ (SessionBean)

(a) タイムアウト値の設定

サービス部品に接続するときのタイマ (SessionBean) は、HCSC サーバ全体に設定したり、サービスアダプタ個別に設定したりできます。設定方法を次に示します。

- HCSC サーバ全体の設定

HCSC サーバ稼働マシンの Cosminexus アプリケーションサーバの EJB コンテナでタイムアウトを設定します。RMI-IIOP 通信のタイムアウトの詳細については、マニュアル「Cosminexus アプリケーションサーバ 機能解説 基本・開発編 (EJB コンテナ)」の「2.11.5 RMI-IIOP 通信のタイムアウト」を参照してください。また、設定するパラメタの詳細については、マニュアル「Cosminexus アプリケーションサーバ リファレンス 定義編 (サーバ定義)」の「15.3 `usrconf.properties` (Java アプリケーション用ユーザプロパティファイル)」を参照してください。

表 5-7 HCSC サーバ全体のタイマ (SessionBean) の設定

値	設定するファイル	キー名称	デフォルト値 (秒)
RMI-IIOP 通信すべてに有効となるタイムアウト値	EJB クライアントアプリケーション用ユーザプロパティファイル (<code>usrconf.properties</code>)	<code>ejbserver.rmi.request.timeout</code>	0 (タイムアウトしません)
CORBA ネーミングサービス部品との通信でのタイムアウト値		<code>ejbserver.jndi.request.timeout</code>	0 (タイムアウトしません)

- サービスアダプタ個別の設定

開発環境で定義する場合、サービスアダプタ定義画面のクライアント定義ファイルに

5. システム動作を円滑にするための機能

指定するファイルで設定します。クライアント定義ファイルの設定については、マニュアル「Cosminexus アプリケーションサーバ SOAP アプリケーション開発の手引」の「10.3 クライアント定義ファイルの設定」を参照してください。

表 5-8 サービスアダプタ個別のタイマ (SessionBean) の設定

値	キー名称	デフォルト値 (秒)
呼び出しタイムアウト値	c4web.application.ejb_timeout	0 (タイムアウトしません)

(b) タイムアウト値の変更

すでに HCSC サーバに配備しているサービスアダプタに対して運用環境で定義する場合は、cscsvccctl コマンドを使用して、サービス部品呼び出しの通信タイムアウト値を変更します。変更方法の詳細については、マニュアル「Cosminexus サービスプラットフォーム システム構築・運用ガイド」の「5.2.25 サービス部品呼び出しの通信タイムアウト値を変更する」を参照してください。cscsvccctl コマンドの詳細は、マニュアル「Cosminexus サービスプラットフォーム リファレンス」の「cscsvccctl (サービス情報の管理)」を参照してください。

表 5-9 タイマ (SessionBean) の設定値の変更

値	キー名称
呼び出しタイムアウト値	<クラスタ名>.<サービス ID>.SessionBean.c4web.application.ejb_timeout

(3) HCSC サーバに接続するときのタイマ (Web サービス)

HCSC サーバに接続するときのタイマ (Web サービス) は、サービスリクエスト稼働マシン側で設定します。サービスリクエストの稼働しているマシン全体に設定したり、サービスリクエスト個別に設定したりできます。

SOAP 通信基盤および JAX-WS エンジンを使用する場合に分けて、設定方法を示します。

(a) タイムアウト値の設定 (SOAP 通信基盤)

SOAP 通信基盤を利用する場合のタイムアウト値の設定方法について説明します。タイムアウト値は、サービスリクエスト稼働マシン全体またはサービスリクエスト個別に設定できます。

- サービスリクエスト稼働マシン全体の設定

サービスリクエスト稼働マシンでの SOAP 通信基盤のサーバ定義ファイルで定義します。設定内容は「5.1.1(1)(a) タイムアウト値の設定 (SOAP 通信基盤)」の場合と同様

です。サーバ定義ファイルの設定については、マニュアル「Cosminexus アプリケーションサーバ SOAP アプリケーション開発の手引」の「10.2 サーバ定義ファイルの設定」を参照してください。

- サービスリクエスト個別の設定

作成するサービスリクエストごとに個別に設定する場合、サービスリクエストを実行するときに指定するクライアント定義ファイルで設定します。設定内容は「5.1.1(1)(a) タイムアウト値の設定 (SOAP 通信基盤)」の場合と同様です。クライアント定義ファイルの設定については、マニュアル「Cosminexus アプリケーションサーバ SOAP アプリケーション開発の手引」の「10.3 クライアント定義ファイルの設定」を参照してください。

(b) タイムアウト値の設定 (JAX-WS エンジン)

JAX-WS エンジンを利用する場合のタイムアウト値の設定方法について説明します。タイムアウト値は、サービスリクエスト稼働マシン全体またはサービスリクエスト個別に設定できます。

- サービスリクエスト稼働マシン全体の設定

サービスリクエスト稼働マシンでの JAX-WS エンジンの共通定義ファイルで定義します。設定内容は「5.1.1(1)(c) タイムアウト値の設定 (JAX-WS エンジン)」の場合と同様です。共通定義ファイルの設定については、マニュアル「Cosminexus アプリケーションサーバ Web サービス開発の手引」の「10.1.2 共通定義ファイルの設定項目」を参照してください。

- サービスリクエスト個別の設定

作成するサービスリクエストごとに個別に設定する場合、プロセス別の定義ファイルで設定します。プロセス別の定義ファイルの設定については、マニュアル「Cosminexus アプリケーションサーバ Web サービス開発の手引」の「10.1.3 プロセス別の定義ファイルの設定」を参照してください。

(4) HCSC サーバに接続するときのタイマ (SessionBean)

HCSC サーバに接続するときのタイマ (SessionBean) は、サービスリクエストの稼働しているマシン全体に設定したり、サービスリクエスト個別に設定したりできます。設定方法を次に示します。

HCSC サーバに接続するときのタイマ (SessionBean) は、サービスリクエスト稼働マシン側で設定します。サービスリクエストの稼働しているマシン全体に設定したり、サービスリクエスト個別に設定したりできます。サービスリクエスト稼働マシンで Cosminexus を使用している場合の設定方法を次に示します。

- サービスリクエスト稼働マシン全体の設定

サービスリクエスト稼働マシンの Cosminexus で、EJB コンテナでタイムアウトを設定します。設定内容は「(2) サービス部品に接続するときのタイマ (SessionBean)」の場合と同様です。RMI-IIOP 通信のタイムアウトの詳細については、マニュアル「Cosminexus アプリケーションサーバ 機能解説 基本・開発編 (EJB コンテナ)」の

5. システム動作を円滑にするための機能

「2.11.5 RMI-IIOP 通信のタイムアウト」を参照してください。設定するパラメタの詳細については、マニュアル「Cosminexus アプリケーションサーバリファレンス 定義編 (サーバ定義)」の「15.3 usrconf.properties (Java アプリケーション用ユーザプロパティファイル)」を参照してください。

- サービスリクエスト個別の設定
作成するサービスリクエストごとに個別に、create から remove までの API 通信のタイムアウト値を設定する場合、サービスリクエストのアプリケーション開発時に API (RequestTimeoutConfigFactory クラスおよび RequestTimeoutConfig クラスのメソッド) で設定します。

(5) HCSC サーバ内のトランザクションのタイマ

HCSC サーバ内のトランザクションのタイマは、HCSC サーバ全体で設定します。HCSC サーバ稼働マシンの Cosminexus アプリケーションサーバの EJB コンテナでタイムアウトを設定します。設定するパラメタの詳細については、マニュアル「Cosminexus アプリケーションサーバリファレンス 定義編 (サーバ定義)」の「15.3 usrconf.properties (Java アプリケーション用ユーザプロパティファイル)」を参照してください。

表 5-10 HCSC サーバ内のトランザクションのタイマの設定

値	設定するファイル	キー名称	デフォルト値 (秒)
EJB クライアントで開始されるトランザクションのトランザクションタイムアウト値	EJB クライアントアプリケーション用ユーザプロパティファイル (usrconf.properties)	ejbserver.jta.TransactionManager.defaultTimeOut	180

5.1.2 サービス部品のタイムアウト値の変更

通信タイムアウト値は開発環境で定義しますが、Web サービス用のサービスアダプタ、および SessionBean 用のサービスアダプタについては、サービス部品に接続するときの通信タイムアウト値を運用環境で変更できます。運用環境でサービス部品のタイムアウト値を変更する場合は、cscsvctl コマンドを使用します。ただし、変更対象のサービスアダプタが未配備状態である必要があります。

変更方法の詳細については、マニュアル「Cosminexus サービスプラットフォーム システム構築・運用ガイド」の「5.2.25 サービス部品呼び出しの通信タイムアウト値を変更する」を参照してください。cscsvctl コマンドの使い方については、マニュアル「Cosminexus サービスプラットフォーム リファレンス」の「cscsvctl (サービス情報の管理)」を参照してください。

5.1.3 タイマ設定時の注意事項

(1) サービス部品に接続するときのタイマと HCSC サーバに接続するときのタイマの関係

HCSC サーバに接続するときのタイマは、サービス部品に接続するときのタイマよりも大きい値を指定してください。

サービス部品呼び出しで時間が掛かり正常応答した場合でも、HCSC サーバに接続するときのタイマが先にタイムアウトし、処理が正常に完了しません。

なお、ビジネスプロセスを呼び出して、サービス部品を呼び出す場合は、複数のサービス部品を呼び出すことを考慮して、HCSC サーバに接続するときのタイマを設定する必要があります。

(2) サービス部品に接続するときのタイマとトランザクションのタイマの関係

トランザクションのタイマは、サービス部品に接続するときのタイマよりも大きい値を指定してください。

サービス部品呼び出しで時間が掛かり正常応答した場合でも、トランザクションがタイムアウトすると、トランザクションがロールバックします。

5.2 サービス部品の接続先変更

サービス部品の接続先は開発環境で定義しますが、次に示すサービスアダプタについては、サービス部品に接続するための接続先を運用環境で変更できます。

- Web サービス用サービスアダプタ
- SessionBean 用サービスアダプタ
- MDB (WS-R) 用サービスアダプタ
- MDB (DB キュー) 用サービスアダプタ

運用環境でサービス部品の接続先を変更する場合は、`cscsvcttl` コマンドを使用します。ただし、変更対象のサービスアダプタが未配備状態である必要があります。

変更方法の詳細については、マニュアル「Cosminexus サービスプラットフォーム システム構築・運用ガイド」の「5.2.23 サービス部品の接続先を変更する」を参照してください。`cscsvcttl` コマンドの使い方については、マニュアル「Cosminexus サービスプラットフォーム リファレンス」の「`cscsvcttl` (サービス情報の管理)」を参照してください。

5.3 SOAP Fault 動作定義ファイルによるフォルトのキャッチ

Web サービス用サービスアダプタでの SOAP Fault の扱いと、フォルトとして扱う場合の注意事項について説明します。

(1) SOAP Fault の扱い

Web サービス用サービスアダプタでは、サービス部品から返された SOAP Fault が次に示す条件をすべて満たす場合、フォルトとして扱われます。

- サービス部品の WSDL に定義されている
- フォルト名が `faultCode` に設定されている

このようなフォルトをユーザ定義例外と呼びます。ユーザ定義例外は、ビジネスプロセスでも認識されるため、フォルト発生後のプロセスの場合分け処理を定義できます。

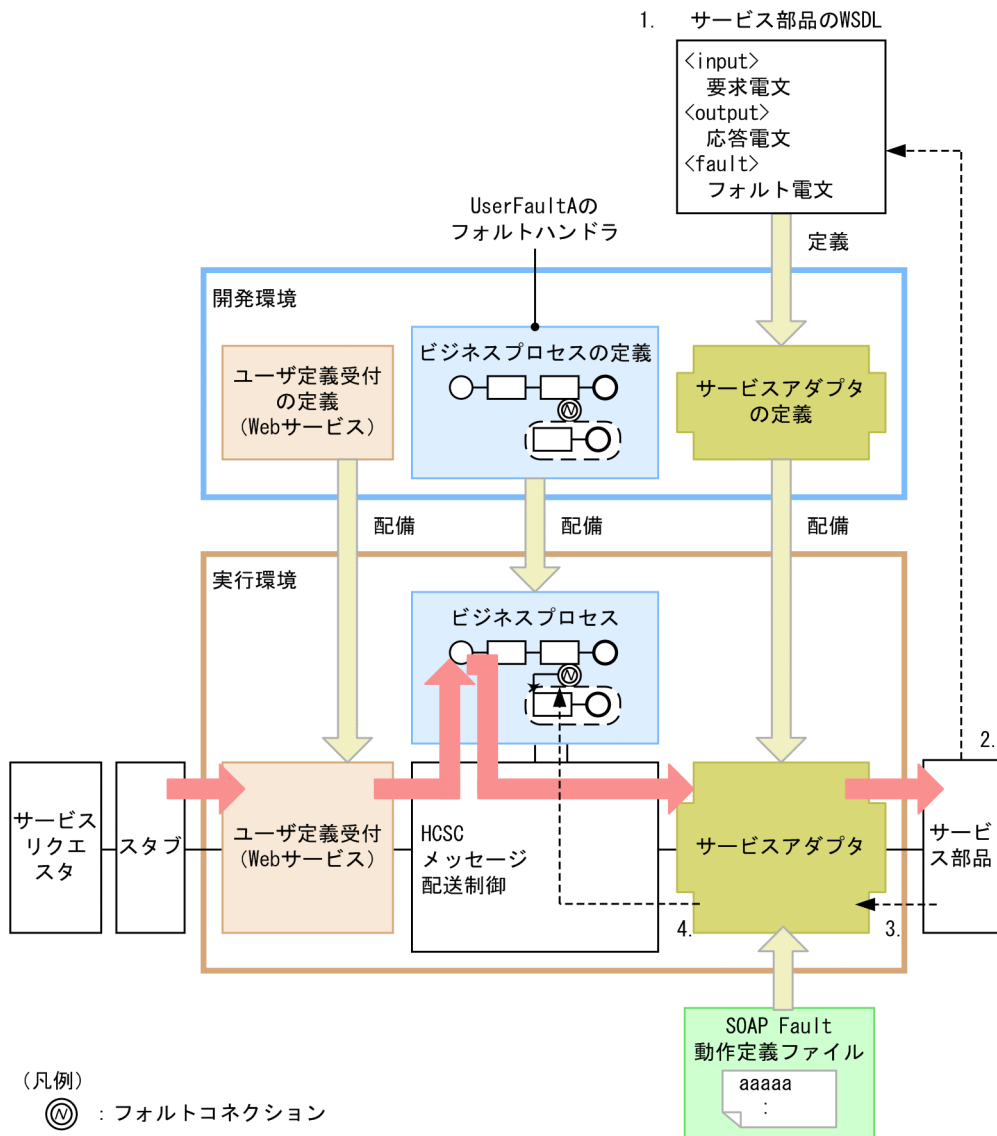
サービス部品の種類によっては、WSDL に定義されていない SOAP Fault が返されたり、`faultCode` に任意の値が設定されて SOAP Fault が返されたりすることがあります。このような SOAP Fault をフォルトとして扱いたいときは、SOAP Fault 動作定義ファイルを使用することで、サービス部品から返されるすべての SOAP Fault をフォルトとして扱うことができるようになります。

SOAP Fault 動作定義ファイルには、サービス部品の WSDL に宣言されている名前空間 (`targetNamespace` の URI) を定義します。SOAP Fault 動作定義ファイルの詳細は、マニュアル「Cosminexus サービスプラットフォーム リファレンス」の「SOAP Fault 動作定義ファイル」を参照してください。

個々のサービスアダプタには、WSDL でサービス部品のインターフェースが定義されており、WSDL に定義されている名前空間が、SOAP Fault 動作定義ファイルに定義されている内容と一致するかどうかで、フォルトとして扱うかどうかを判断します。

SOAP Fault 動作定義ファイルを使用したときの SOAP Fault の扱いを次の図に示します。

図 5-2 SOAP Fault 動作定義ファイルを使用したときの SOAP Fault の扱い



1. サービス部品の WSDL を定義します。
2. サービス部品から返された SOAP Fault が、サービス部品の WSDL に定義されていればフォルトとして扱われます。
3. サービス部品から返された SOAP Fault が、サービス部品の WSDL に定義されていない、または WSDL の定義と異なる名前空間で定義されている場合は、SOAP Fault 動作定義ファイルに定義されている内容と一致するかどうかを照合します。
4. SOAP Fault 動作定義ファイルに定義されている内容と一致している場合は、フォルトとしてビジネスプロセスに戻します。

(2) SOAP Fault をフォルトとして扱う場合の注意事項

サービス部品から返された SOAP Fault では、次の条件をすべて満たす場合にフォルト名が `faultCode` に設定され、フォルトとして扱われます。

- サービス部品側に SOAP 通信基盤を使用していること。
- `fault` メッセージを定義する `wsdl:part` 要素が、必ず `element` 属性で `xsd:schema` 要素内の `element` 要素を参照していること。
- WSDL `fault` 名 (`wsdl:fault` 要素の `name` 属性値) と、`schema` (`wsdl:types` 内) に定義された例外型 `complexType` を指す `element` 名 が、名前空間を含めて一致していること。

これらの条件を満たしていない WSDL を利用する場合で、サービス部品から返された SOAP Fault をフォルトとして扱いたいときは、SOAP Fault 動作定義ファイルを定義してユーザ定義例外となるように設定します。

注

`element` 名とは、「`wsdl:fault` が参照する `wsdl:part` 要素が、`element` 属性を用いて指定する要素」のことです。WSDL を次の順にたどって特定できる要素を指します。

1. `wsdl:fault`
2. `wsdl:message` (この要素をたどらない場合もあります)
3. `wsdl:part`
4. `xsd:element`

なお、Cosminexus の開発支援コマンドである `Java2WSDL` コマンドを使用して、オペレーションの `style` 属性が「`rpc`」の WSDL ファイルを生成した場合、`element` 属性ではなく、`type` 属性で参照する WSDL ファイルが生成されます。そのため、生成された WSDL ファイルをそのまま使用しないで、修正および再生成してから使用します。WSDL ファイルの修正および再生成の方法については、マニュアル「Cosminexus サービスプラットフォーム 開発ガイド」の「4.3.2 サービス部品電文を作成する (Web サービスの場合)」を参照してください。

Cosminexus サービスプラットフォームでフォルトを使用する場合、`document` スタイルの WSDL を使用することを推奨します。

5.4 アプリケーションからの稼働状態の確認

5.4.1 稼働状態の確認の概要

サービスリクエストなどの外部から Web サービス (SOAP 通信) を使用し、サービス名をキーにして稼働状態の確認要求をすると、HCSC サーバに配備されているサービスアダプタの情報を取得できます。

なお、稼働状態の確認は、サービス部品そのものの状態を確認するのではなく、HCSC サーバに配備されているサービスアダプタの状態を確認します。

5.4.2 稼働状態の確認の流れ

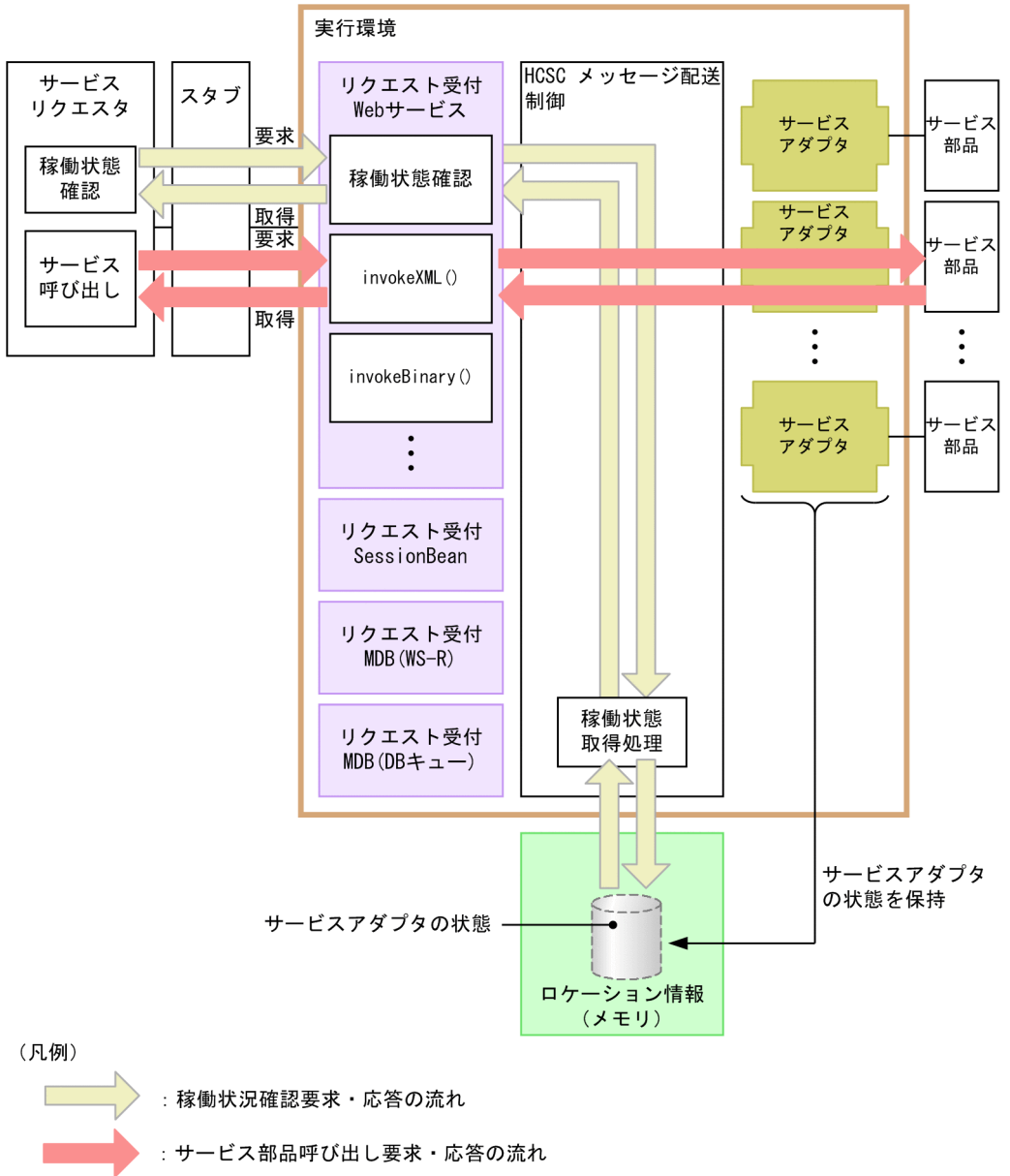
SOAP 用リクエスト受付にはサービスリクエストからのメッセージを配送するためのメソッド (リクエスト受付のオペレーション) として `invokeXML`, `invokeBinary` などがあり、その中のサービスアダプタ稼働状態確認用のオペレーション (`getServiceInfo`) を使用し、稼働状態を確認します。

サービスリクエストからサービスアダプタの稼働状態の確認要求をすると、HCSC サーバは指定されたサービス名に該当するサービスアダプタのロケーション情報 (メモリ上の情報) を検索し、その状態を応答します。

なお、稼働状態確認の要求ができるのは、HCSC サーバのリクエスト受付が起動しているときだけです。

稼働状態の確認の流れを次の図に示します。

図 5-3 稼働状態の確認の流れ



5.4.3 稼働状態の確認時の注意事項

(1) サービス部品の状態とサービスアダプタの状態の同期について

稼働状態確認機能は、サービス部品の状態ではなく、HCSC サーバに配備されているサービスアダプタの状態を確認する機能です。そのため、稼働状態を確認するときは、サービス部品とサービスアダプタが共に同期を取って起動および停止する運用が前提で

5. システム動作を円滑にするための機能

す。サービスアダプタが起動していてもサービス部品が停止している場合は、サービス部品呼び出しの要求はエラーになります。

(2) 稼働状態の確認結果と実状態の差異について

稼働状態の確認でサービスアダプタの起動を確認した場合でも、確認直後にサービスアダプタが停止すると、サービス部品呼び出しの要求をしたときにエラーになります。そのため、サービスリクエストでは、サービスアダプタの稼働状態確認で起動を確認しても、サービス部品呼び出しを要求したときにサービスアダプタが停止状態となっていてエラーリターンする必要があることを考慮する必要があります。

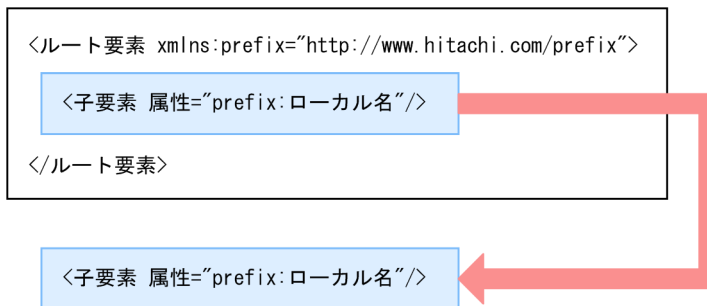
(3) サービス受付の状態について

HCSC サーバのリクエスト受付が起動していないときに、サービスリクエストでサービスアダプタの稼働状態を確認するとエラーになります。そのため、サービスリクエストでは、リクエスト受付が停止状態で、サービスアダプタの稼働状態確認要求がエラーリターンする場合は考慮する必要があります。

5.5 XML 電文内の属性値に指定した名前空間接頭辞の補完

HCSC サーバ内を流れるユーザ電文で、XML スキーマの属性値の型に QName 型を使用する場合、次の図のように XML 文書から「子要素」を部分木として切り出して出力すると、xmlns:prefix の名前空間宣言が欠落します。

図 5-4 部分木取得時の名前空間宣言の欠落



名前空間宣言が欠落した場合、次の問題が発生することがあります。

- ビジネスプロセスでフォルトコネクションを使用してフォルトをキャッチしようとしたが、サービス部品からのフォルトをキャッチできない。
- データ検証機能を使用している場合（HCSC サーバランタイム定義ファイルの `telegram-validation` プロパティに ON を指定）、検証でエラーが発生する。
- 名前空間宣言が欠落した XML 電文をサービスリクエストやサービス部品で受け取った場合、サービスリクエストやサービス部品側でエラーが発生する。

なお、子要素のプリフィックスについては、部分木取得時に名前空間が解決されます。

このような問題を解決するには、HCSC サーバランタイム定義ファイルの `xmltelegram-namespace-complement` プロパティに ON を指定します。ただし、ON を指定すると、OFF を指定した場合よりも電文サイズが大きくなるため注意してください。

HCSC サーバランタイム定義ファイルについては、マニュアル「Cosminexus サービスプラットフォーム リファレンス」の「HCSC サーバランタイム定義ファイル」を参照してください。

5.6 任意形式（any 形式）の電文フォーマットの定義

開発環境で電文のフォーマット定義ファイルに任意形式（any 形式）を設定することで、電文の形式を意識しないサービス部品呼び出し要求・応答ができます。

なお、任意形式のデータを扱うことができる受付およびアダプタは、次のとおりです。

- ユーザ定義受付（カスタム受付）
- 汎用カスタムアダプタ

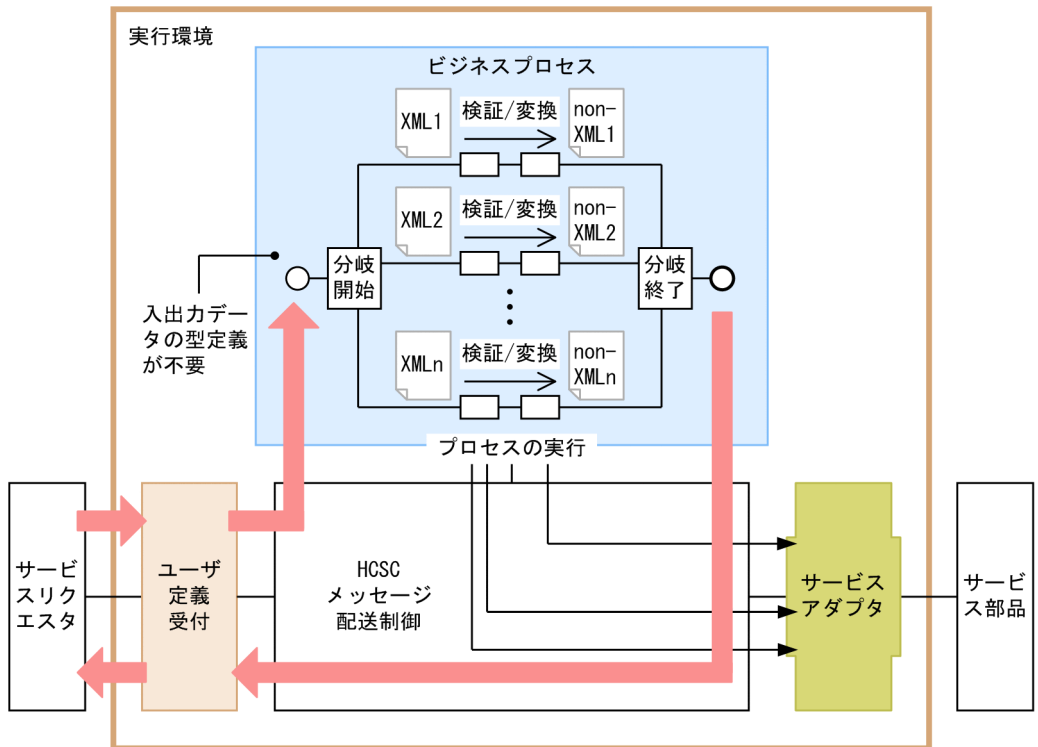
データ変換をしないメッセージ型の組み合わせについては「5.6.2 データ変換をしない場合の変数（メッセージ型）の組み合わせ」を参照してください。

5.6.1 任意形式（any 形式）を設定した場合の処理の流れ

電文のフォーマット定義ファイルに任意形式（any 形式）を設定した場合は、要求電文および応答電文のデータの型を意識する必要がありません。

電文のフォーマット定義ファイルに任意形式（any 形式）を設定した場合の処理の流れを次の図に示します。

図 5-5 電文のフォーマット定義ファイルに任意形式（any 形式）を設定した場合の処理の流れ



(凡例)

➡ : サービス部品呼び出し要求・応答の流れ
 ➡ : アダプタの呼び出し

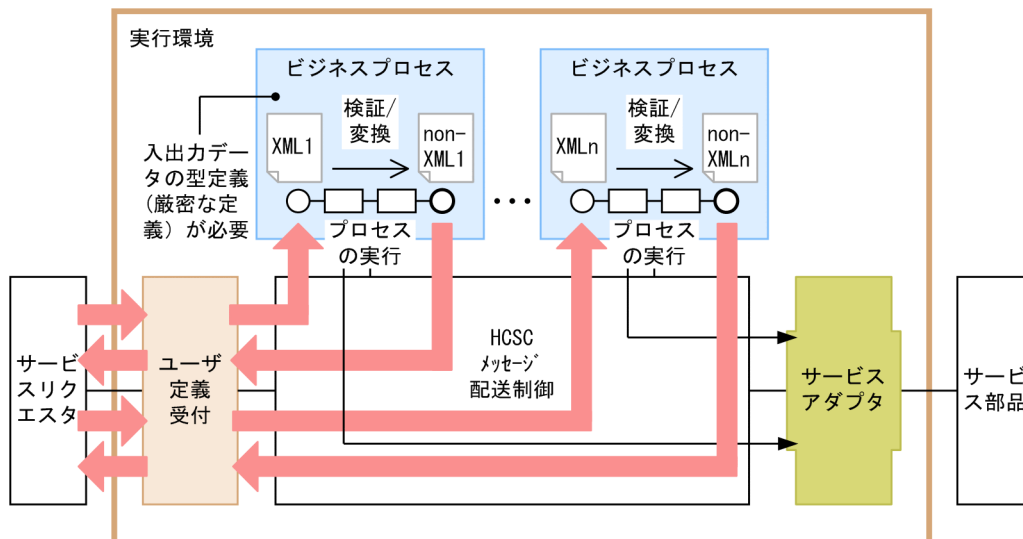
参考

電文のフォーマット定義ファイルに任意形式（any 形式）を設定しない場合は、要求電文および応答電文のデータ型を厳密に決める必要があります。また、開発環境で電文の形式ごとにコンポーネント（受付、ビジネスプロセス、およびアダプタ）を作成する必要があります。

電文のフォーマット定義ファイルに任意形式（any 形式）を設定しない場合の処理の流れを次の図に示します。

5. システム動作を円滑にするための機能

図 5-6 電文のフォーマット定義ファイルに任意形式（any 形式）を設定しない場合の処理の流れ



(凡例)

➡ : サービス部品呼び出し要求・応答の流れ → : アダプタの呼び出し

5.6.2 データ変換をしない場合の変数（メッセージ型）の組み合わせ

任意形式（any 形式）を設定することで、データ変換をしない運用ができます。データ変換をしない場合の前提条件は次のとおりです。

- サービス呼出アクティビティのメッセージ型（XML, non-XML, any）とサービス呼出アクティビティから呼び出すコンポーネント（開発環境で作成する受付、ビジネスプロセス、およびアダプタの総称）で設定したデータの型が一致している。
- ビジネスプロセスのサービス呼出アクティビティで任意形式（any 形式）を設定する場合は、開発環境の定義画面で呼び出し対象のコンポーネントを任意形式（any 形式）で定義している。

受付およびアダプタでデータ変換をしない場合のメッセージ型の組み合わせを次に示します。

表 5-11 受付でデータ変換をしない場合のメッセージ型の組み合わせ

コンポーネント種別	データの型 または メソッド		変数 (メッセージ型)						
			XML		non-XML		any		
			要求	応答	要求	応答	要求	応答	
標準受付 (SOAP)	invokeXML				×	×	×	×	
	invokeBinary	×	×						
標準受付 (EJB)	invokeXML				×	×	×	×	
	invokeBinary	×	×						
標準受付 (MDB(WS-R))	onMessage								
標準受付 (MDB(DB キュー))	onMessage								
ユーザ定義受付 (SOAP 受付)					×	×	×	×	
ユーザ定義受付 (カスタム 受付)	invokeXML	XML				×	×	×	×
		non-XML	×	×	×	×	×	×	×
		any							
	invokeBinary	XML	×	×	×	×	×	×	×
		non-XML	×	×			×	×	
		any							
ユーザ定義受付 (TP1/RPC 受付)	XML	×	×	×	×	×	×	×	
	non-XML	×	×						
	any								
ユーザ定義受付 (FTP 受付)	XML				×	×	×	×	
	non-XML	×	×	×	×	×	×	×	
	any	×	×	×	×	×	×	×	

(凡例)

: 対応しています。

× : 対応していません。

5. システム動作を円滑にするための機能

表 5-12 アダプタでデータ変換をしない場合のメッセージ型の組み合わせ

コンポーネント種別	データの型	変数 (メッセージ型)					
		XML		non-XML		any	
		要求	応答	要求	応答	要求	応答
標準アダプタ (SOAP)	XML			×	×	×	×
	non-XML	-	-	-	-	-	-
	any	-	-	-	-	-	-
標準アダプタ (EJB)	XML			×	×	×	×
	non-XML	-	-	-	-	-	-
	any	-	-	-	-	-	-
標準アダプタ (WSR)	XML		-	×	-	×	-
	non-XML	×	-		-	×	-
	any	-	-	-	-	-	-
標準アダプタ (DBQ)	XML		-	×	-	×	-
	non-XML	×	-		-	×	-
	any	-	-	-	-	-	-
DB アダプタ	XML			×	×	×	×
	non-XML	×	×	×	×	×	×
	any	×	×	×	×	×	×
TP1 アダプタ	XML	×	×	×	×	×	×
	non-XML	×	×			×	×
	any	×	×	×	×		
FTP アダプタ	XML			×	×	×	×
	non-XML	×	×	×	×	×	×
	any	×	×	×	×	×	×
ファイル操作アダプタ	XML			×	×	×	×
	non-XML	×	×	×	×	×	×
	any	×	×	×	×	×	×
ファイルアダプタ	XML			×	×	×	×
	non-XML	×	×			×	×
	any	×	×	×	×	×	×
Object Access アダプタ	XML			×	×	×	×
	non-XML	×	×	×	×	×	×
	any	×	×	×	×	×	×

コンポーネント種別	データの型	変数 (メッセージ型)					
		XML		non-XML		any	
		要求	応答	要求	応答	要求	応答
Message Queue アダプタ	XML			×	×	×	×
	non-XML	×	×			×	×
	any	×	×	×	×	×	×
カスタムアダプタ	XML			×	×	×	×
	non-XML	×	×			×	×
	any	×	×	×	×		

(凡例)

- : 対応しています。
- ×
- : 作成できません。

任意形式 (any 形式) を設定している HCSC コンポーネントを単体サービス実行した場合の対応可否を次の表に示します。

表 5-13 HCSC コンポーネントを単体サービス実行した場合の対応可否

コンポーネント種別	メソッド	フォーマット定義ファイルの設定	
		any	
		要求	応答
標準受付 (SOAP)	invokeXML	×	×
	invokeBinary		
標準受付 (EJB)	invokeXML	×	×
	invokeBinary		
標準受付 (MDB(WS-R))	onMessage		
標準受付 (MDB(DB キュー))	onMessage		

(凡例)

- : 対応しています。
- ×

6

性能向上のための機能

この章では、性能向上のための機能として、サービス部品呼び出し処理の多重度、および XML 電文のキャッシュ機能について説明します。

6.1 サービス部品呼び出し処理の多重度

6.2 XML 電文のキャッシュ機能

6.1 サービス部品呼び出し処理の多重度

サービスリクエストからのサービス部品の呼び出し要求は、同時に実行される場合があります。そのため、サービス部品呼び出し処理の多重度（同時に実行できる数）を設定する必要があります。

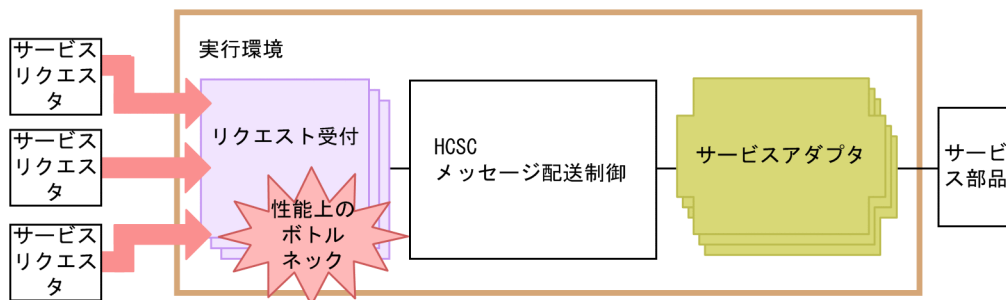
6.1.1 HCSC サーバ内の多重度

多重度の設定が良くない場合、性能上のボトルネックになるおそれがあります。

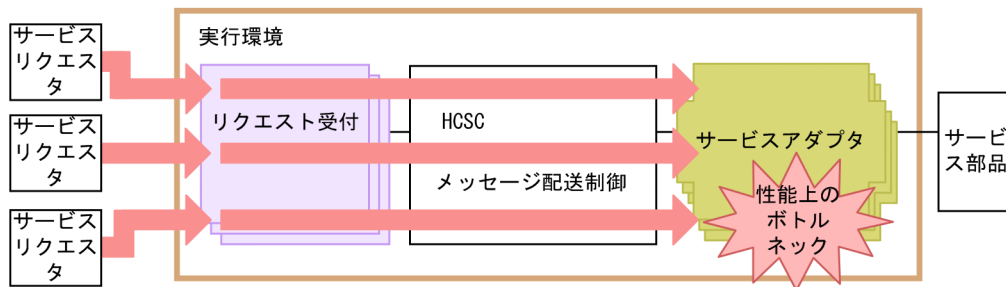
多重度の設定によってボトルネックになる例を次の図に示します。

図 6-1 多重度の設定によってボトルネックになる例

● リクエスト受付の多重度が小さい場合



● サービスアダプタの多重度が小さい場合



(凡例)


 : サービス部品呼び出し要求の流れ

図 6-1 に示すように、リクエスト受付（標準受付またはユーザ定義受付）の多重度が小さいと、一度にたくさん要求が来ても同時に処理ができないため、一度に実行できる処理が少なくなります。また、リクエスト受付の多重度を大きくしても、サービスアダプタで定義している多重度が小さいと、サービスアダプタで一度に実行できる処理が少ないため、サービスアダプタがボトルネックになり、意図した性能を得られなくなります。

(1) HCSC サーバでの多重度

多重度は、標準の同期受付（Web サービス）の場合は、スレッド数および最大同時実行数で設定します。標準の同期受付（SessionBean）の場合は、インスタンス数で設定します。HCSC サーバ内の多重度の設定について、表 6-1、表 6-2、および表 6-3 に示します。

表 6-1 標準受付の多重度を設定するプロパティ

定義ファイル	プロパティ	内容
HCSC サーバランタイム 定義ファイル	request-ejb.instance.minimum	標準の同期受付（SessionBean）のインスタンス最小数
	request-ejb.instance.maximum	標準の同期受付（SessionBean）のインスタンス最大数
	request-ejb.parallel.count	CTM がアプリケーションを呼び出すために用意するスレッド数
	request-soap.instance.minimum	標準の同期受付（Web サービス・SOAP1.1）の最小同時実行数
	request-soap.instance.maximum	標準の同期受付（Web サービス・SOAP1.1）の最大同時実行数
	request-soap1_2.instance.minimum	標準の同期受付（Web サービス・SOAP1.2）の最小同時実行数
	request-soap1_2.instance.maximum	標準の同期受付（Web サービス・SOAP1.2）の最大同時実行数
	request-jms.instance.maximum	標準の非同期受付（MDB（WS-R））のインスタンス最大数

表 6-2 ユーザ定義受付の多重度を設定するプロパティ

定義ファイル	プロパティ	内容
ユーザ定義受付ランタイム定義ファイル	user-defined-reception-soap.threads.maximum	最大同時実行数

表 6-3 サービスアダプタの多重度を設定する項目

画面	内容
開発環境でのサービスアダプタ定義画面	最大インスタンス数

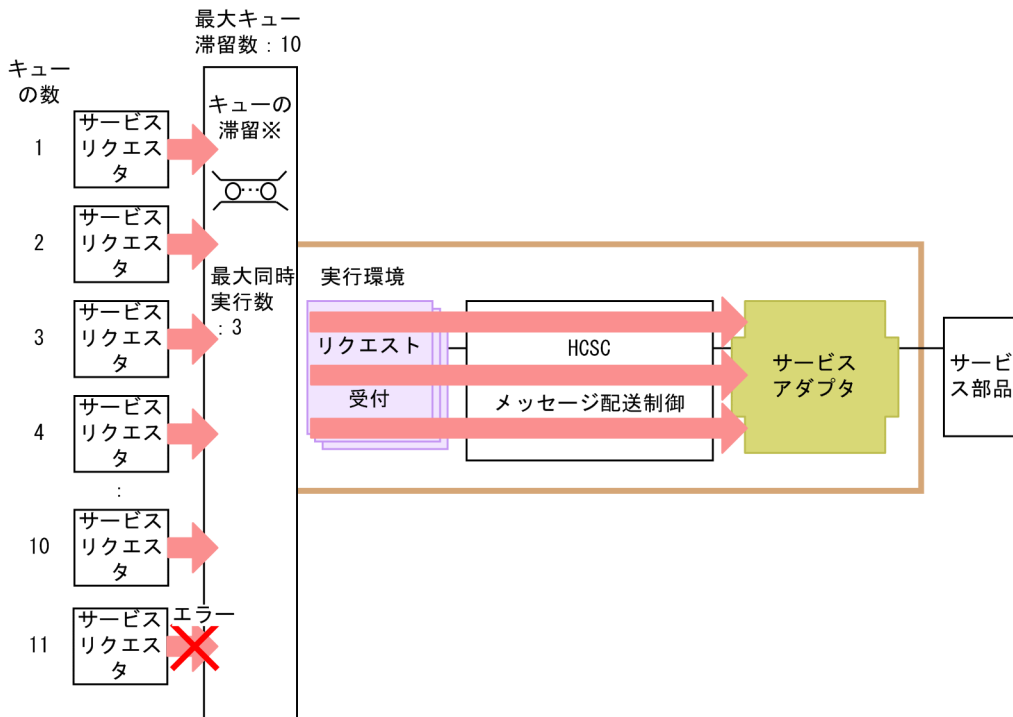
6. 性能向上のための機能

注 最大インスタンス数に 0 を指定した場合は、上限が設定されないで無制限になります。


(2) Web サービス (SOAP 通信) での多重度

Web サービス (SOAP 通信) の場合、同時実行数のほかに Web コンテナレベルでの最大キュー滞留数を設定できます。最大同時実行数と最大キュー滞留数を設定しておくことで、サービスリクエストからの要求を受けたときに、実際の処理をしないで処理を待たせておけるため、流量制御ができます。サービスリクエストからの要求と最大キュー滞留数の関係を次の図に示します。

図 6-2 サービスリクエストからの要求と最大キュー滞留数の関係



(凡例)

 : サービス部品呼び出し要求の流れ

注※ Webサービスの実行待ちキューの滞留を示します。

例えば、図 6-2 に示すように、最大同時実行数を 3 とした場合、最大キュー滞留数を 10 にしておくと、同時にきたサービス部品呼び出し要求のうち、3 件までは同時に処理されます。同時実行数より多くのサービス部品呼び出し要求 (4 件以上) がきた場合は、最大キュー滞留数 (10 件) までは、処理待ちの状態となり、先に実行されている処理が終わり次第、順次処理が実行されます。

最大キュー滞留数の数が上限（10件）まで達している場合に、さらにサービスリクエストからサービス部品呼び出し要求がきた場合は、要求は受け付けられないため、エラーとしてサービスリクエストに応答されます。

HCSC サーバ内の最大キュー滞留数は、表 6-4 および表 6-5 に示す定義ファイルのプロパティで設定します。

表 6-4 標準受付の最大キュー滞留数を設定するプロパティ

定義ファイル	プロパティ	内容
HCSC サーバランタイム 定義ファイル	request-soap.queue-size	標準の同期受付（Web サービス・SOAP1.1）の実行待ちキューのサイズ
	request-soap1_2.queue-size	標準の同期受付（Web サービス・SOAP1.2）の実行待ちキューのサイズ

表 6-5 ユーザ定義受付の最大キュー滞留数を設定するプロパティ

定義ファイル	プロパティ	内容
ユーザ定義受付ランタイム定義ファイル	user-defined-reception-soap.queue-size	実行待ちキューサイズ

6.1.2 データベースへのアクセスに関する多重度

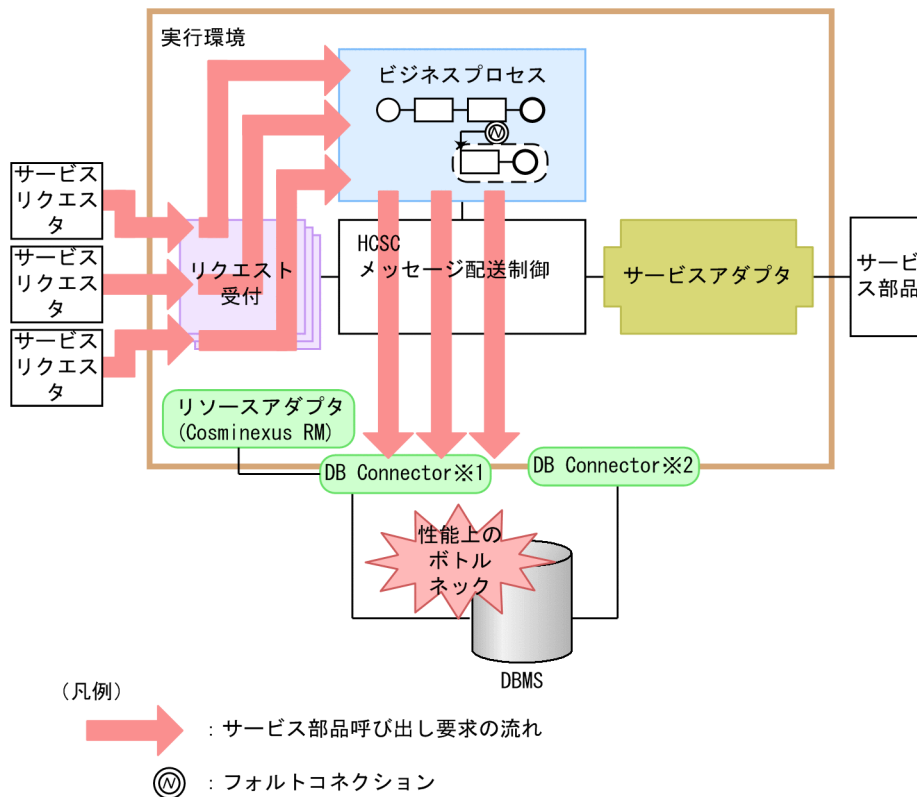
永続化するビジネスプロセスを使用する（プロセスインスタンスの実行履歴を採取する場合、非同期プロトコルを使用する場合、および電文の実行履歴を採取する場合は、HCSC サーバに設定する DB Connector の定義で、コネクションプール数を設定します。

(1) 永続化するビジネスプロセスを使用する場合

HCSC サーバに設定した二つの DB Connector のうち、HCSC サーバセットアップ定義ファイルの dbcon-xadisplayname プロパティに指定した DB Connector を使用します。リクエスト受付からビジネスプロセスを経由して、一度に多くの要求が来ても、DB Connector の部分でデータベースへの出力が一度にできなくなります。すると、性能上のボトルネックが発生して、その結果、ビジネスプロセスの同時実行処理性能が遅くなります。そのため、プロセスインスタンスの実行履歴を採取するときの多重度を設定する必要があります。永続化するビジネスプロセスを使用する場合のボトルネックを次の図に示します。

6. 性能向上のための機能

図 6-3 永続化するビジネスプロセスを使用する場合のボトルネック



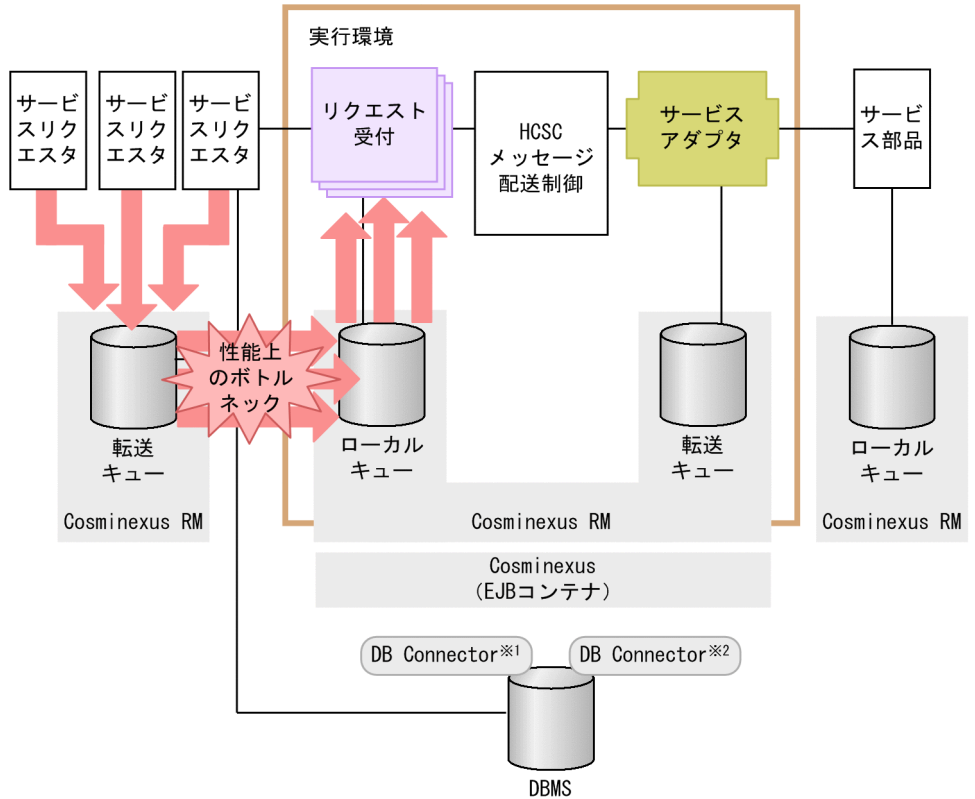
注※1 dbcon-xadisplayname プロパティに指定したLocalTransactionまたはXATransactionの DB Connectorです。
 注※2 dbcon-noisplayname プロパティに指定したNoTransactionまたはXATransactionの DB Connectorです。

(2) 非同期プロトコルを使用する場合

HCSC サーバに設定した二つの DB Connector のうち、HCSC サーバセットアップ定義ファイルの dbcon-xadisplayname プロパティに指定した DB Connector は、非同期 (MDB (WS-R) または MDB (DB キュー)) の標準受付でキューからメッセージを取り出したり、非同期 (MDB (WS-R) または MDB (DB キュー)) のサービスアダプタでメッセージを送信したりするときにも使用します。そのため、プロセスインスタンスの実行履歴を採取するときの多重度に加えて、HCSC サーバがリクエスト受付のキューからメッセージを取り出すときや、サービスアダプタが転送キューにメッセージを送信するときの多重度も含めて、コネクションプール数を検討する必要があります。

キューに対して一度に多くの要求が来ても、Cosminexus RM のリソースアダプタが同時にメッセージの取り出し処理ができないで、性能上のボトルネックが発生します。また、サービスアダプタの多重度を大きくしていても、サービスアダプタがキューにメッセージを送信するときの同時実行数が少ないと、やはり性能上のボトルネックが発生します。非同期プロトコルを使用する場合のボトルネックを次の図に示します。

図 6-4 非同期プロトコルを使用する場合のボトルネック



(凡例)

→ : サービス部品呼び出し要求の流れ

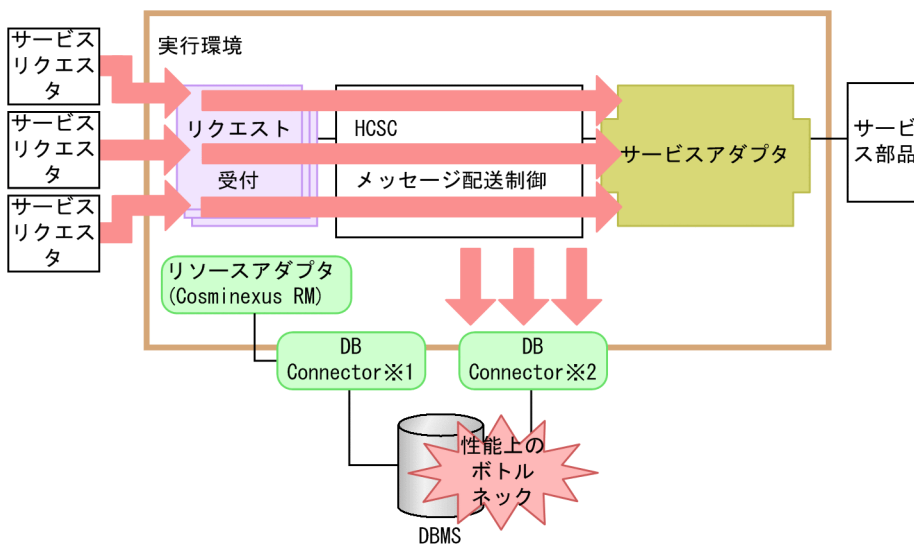
注※1 LocalTransactionまたはXATransaction用のDB Connectorです。
 注※2 NoTransaction用のDB Connectorです。

(3) 電文の実行履歴を採取する場合

HCSC サーバに設定した二つの DB Connector のうち、HCSC サーバセットアップ定義ファイルの dbcon-nodisplayname プロパティに指定した DB Connector を使用します。リクエスト受付に一度に多くの要求が来ても、DB Connector の部分でデータベースへの出力が一度にできなくなるため、性能上のボトルネックが発生して、その結果、サービス部品呼び出しの同時実行処理性能が遅くなります。そのため、電文の実行履歴を採取するときの多重度を設定する必要があります。電文の実行履歴を採取する場合のボトルネックを次の図に示します。

6. 性能向上のための機能

図 6-5 電文の実行履歴を採取する場合のボトルネック



(凡例)

➡ : サービス部品呼び出し要求の流れ

注※1 LocalTransactionまたはXATransaction用のConnectorです。

注※2 NoTransaction用のConnectorです。

データベースへのアクセスに関する多重度は、次の表に示すように設定します。

表 6-6 DB Connector の多重度を設定するプロパティ

定義ファイル	プロパティ	内容
DB Connector (LocalTransaction または XATransaction) のセットアップ時に設定する属性ファイル	MinPoolSize	プールの最小値
	MaxPoolSize	プールの最大値
DB Connector (NoTransaction) のセットアップ時に設定する属性ファイル	MinPoolSize	プールの最小値
	MaxPoolSize	プールの最大値

注 1 プールの最大値を増やした場合、データベース側の同時接続数も変更する必要があるため、注意が必要です (HiRDB の場合、pd_max_users (HiRDB の最大接続数) となります)。

注 2 実行環境が使用する DB コネクション数の最大値については、マニュアル「Cosminexus サービスプラットフォーム システム構築・運用ガイド」の「3.1.2 実行環境に必要なソフトウェアの設定」を参照してください。

表 6-7 Cosminexus RM の多重度を設定するプロパティ

定義ファイル	プロパティ	内容
Cosminexus RM のセットアップ時に 設定する属性ファイル	MinPoolSize	プールの最小値
	MaxPoolSize	プールの最大値

6.1.3 Web サービス (SOAP 通信) に関する多重度

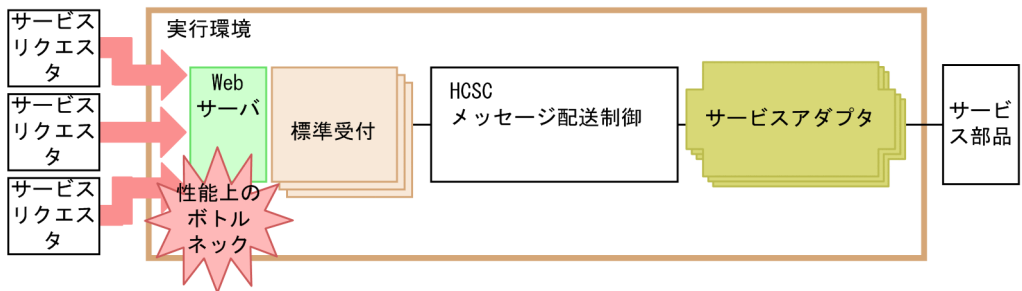
(1) 同時実行スレッド数

Web サービス (SOAP 通信) を使用する場合は、実行環境で設定している Web サーバ (HTTP サーバ) の同時実行スレッド数も関係します。


Web サービス (SOAP 通信) の標準受付やユーザ定義受付が複数ある場合、各受付に設定した同時実行スレッド数の合計より、Web サーバに設定した全体の同時実行スレッド数が優先されます。そのため、Web サーバに設定した全体の同時実行スレッド数が、各受付に設定した同時実行スレッド数の合計より小さいと、一度に多くの要求が来ても同時に処理ができないため、性能上のボトルネックになることがあります。

Web サービス (SOAP 通信) を使用する場合のボトルネックを次の図に示します。

図 6-6 Web サービス (SOAP 通信) を使用する場合のボトルネック



(凡例)

 : サービス部品呼び出し要求の流れ

Web サービス (SOAP 通信) に関する多重度は、次のように設定します。

Hitachi Web Server の場合

コンフィグファイルに指定できるディレクティブでのサーバの性能についての定義をします。

詳細については、マニュアル「Hitachi Web Server」を参照してください。

インプロセス HTTP サーバ機能の場合

運用管理ポータル の [論理サーバの環境設定] - [J2EE サーバ] - [J2EE サーバの論理サーバ名] - [HTTP サーバ] で次の内容を設定します。

6. 性能向上のための機能

- 同時実行スレッド数
- 初期スレッド数

詳細については、マニュアル「Cosminexus アプリケーションサーバ 機能解説 基本・開発編 (Web コンテナ)」の「4. インプロセス HTTP サーバ」を参照してください。

(2) 占有スレッド数

Web サーバ (HTTP サーバ) の同時実行スレッド数が十分な数を確保できない状況であっても、個々のリクエスト受付で必ず最低限の実行をする場合は、占有スレッド数を定義します。

占有スレッド数を定義した場合は、Web サーバ (HTTP サーバ) の同時実行スレッド数以上の要求が個別のリクエスト受付に来た場合でも、必ず指定した数だけは同時に実行されます。

ほかのリクエスト受付に要求が来ていない場合は、最大同時実行数まで同時に実行できます。

HCSC サーバ内の占有スレッド数は、表 6-8 および表 6-9 に示すように設定します。

表 6-8 標準受付の占有スレッド数を設定するプロパティ

定義ファイル	プロパティ	内容
HCSC サーバランタイム 定義ファイル	request-soap.exclusive.threads	標準の同期受付 (Web サービス・SOAP1.1) の占有スレッド数
	request-soap1_2.exclusive.threads	標準の同期受付 (Web サービス・SOAP1.2) の占有スレッド数

表 6-9 ユーザ定義受付の占有スレッド数を設定するプロパティ

定義ファイル	プロパティ	内容
ユーザ定義受付ランタイム 定義ファイル	user-defined-reception-soap.exclusive.threads	占有スレッド数

最大同時実行数と占有スレッド数の関係を次の図に示します。

図 6-7 最大同時実行数と占有スレッド数の関係

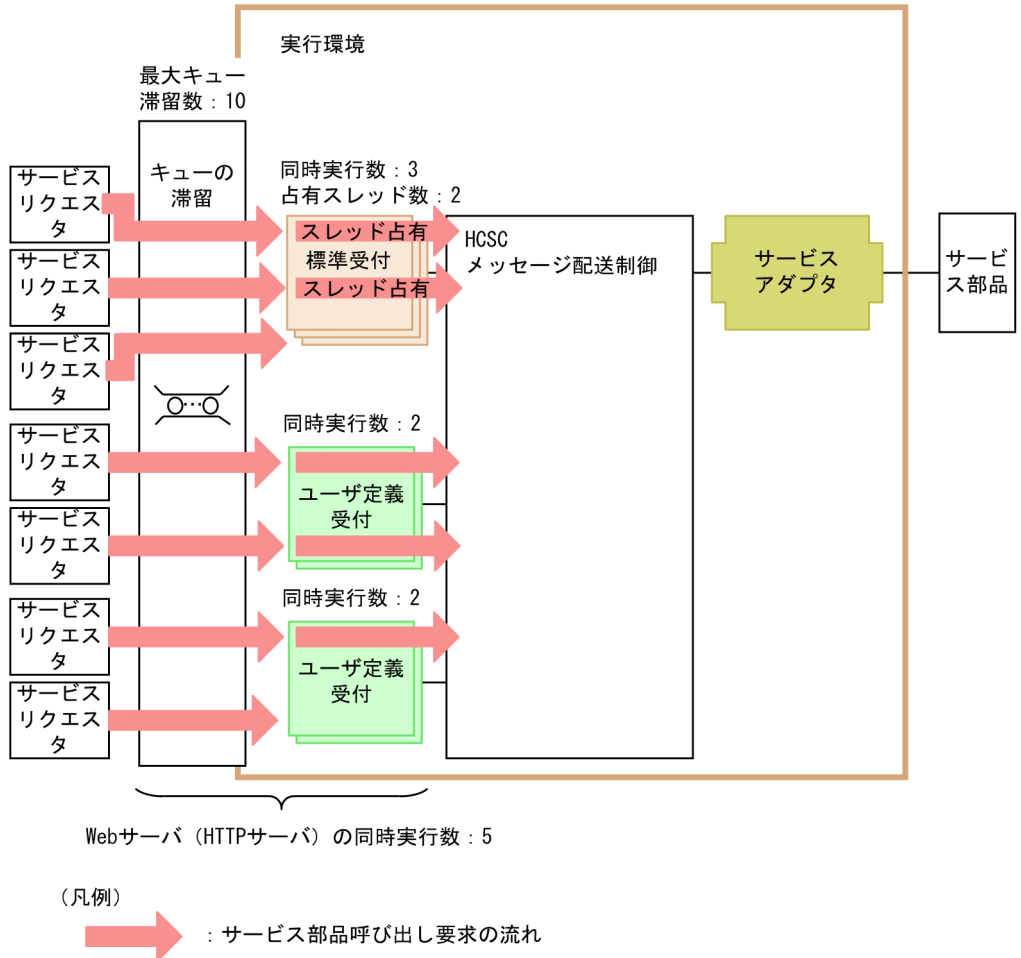


図 6-7 に示すように、標準受付、二つのユーザ定義受付、および Web サーバ (HTTP サーバ) を次のように設定した場合、標準受付は必ず二つのスレッドを占有します。

- 標準受付：最大キュー滞留数 10，同時実行数 3，占有スレッド数 2
- ユーザ定義受付：同時実行数 2
- Web サーバ (HTTP サーバ)：同時実行数 5

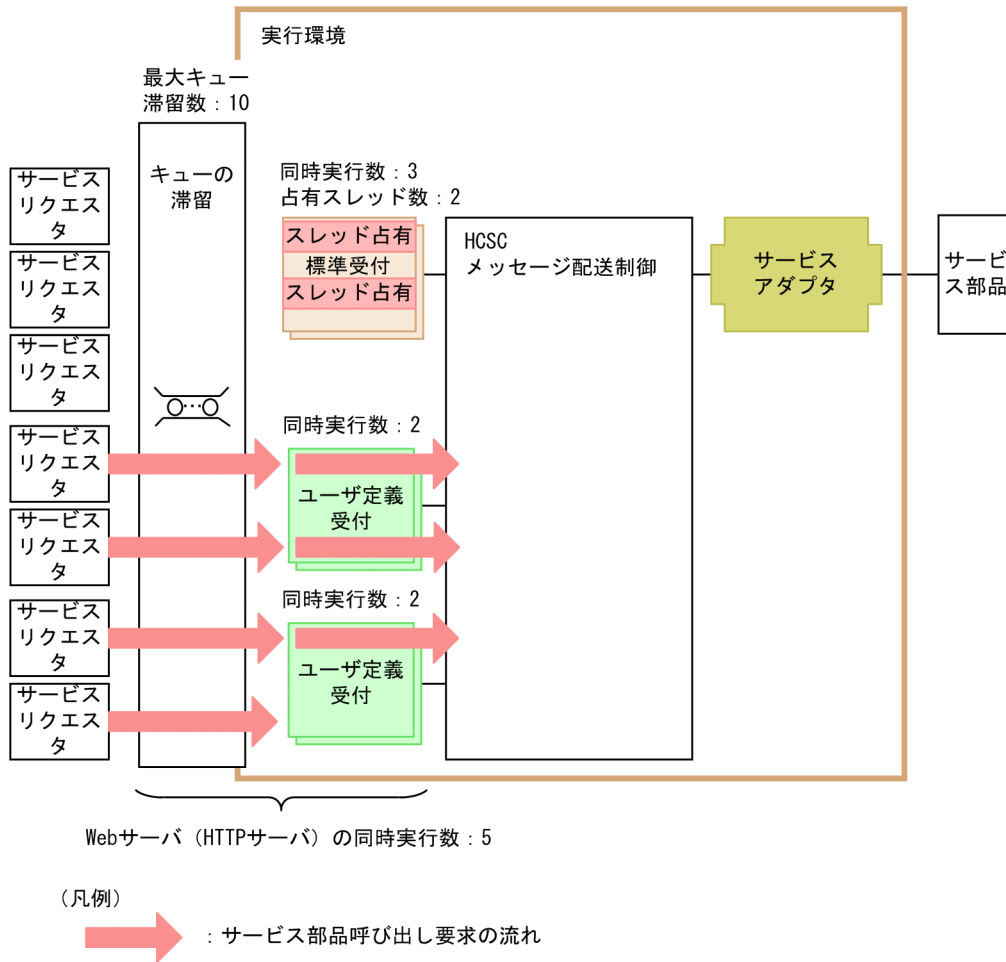
そのため、Web サーバ (HTTP サーバ) の同時実行数に指定した 5 件以上の要求が来ると、二つのスレッドは標準受付の実行のために使用され、残りの三つのスレッドを使ってユーザ定義受付が動作します。

ただし、最大キュー滞留数、標準受付の同時実行数、占有スレッド数、および Web サーバ (HTTP サーバ) の同時実行数が図 6-7 と同じ場合で、図 6-8 に示すように、占有スレッド数を指定した標準受付に要求が来ていないときでも、必ず二つはスレッドを占有して実行する状態になります。また、ほかのユーザ定義受付では、それぞれ同時実行数

6. 性能向上のための機能

が2件で合計の同時実行数は4件になるはずですが、Webサーバ（HTTPサーバ）の同時実行数5件であるため、残り三つしか処理できなくなります。ほかのリクエスト受付の同時実行数が増えることはありません。

図 6-8 占有スレッド数を指定した受付に要求が来ない場合



6.2 XML 電文のキャッシュ機能

(1) XML 電文のキャッシュ機能の概要

XML 電文のキャッシュ機能を使用すると、次に示す条件が同一の XML 電文が流れる場合に、処理速度が向上することがあります。

- 要素の並び順が同じ
- 属性の並び順が同じ
- 要素の繰り返しと同じ

これは、一度解析した XML 文書の特徴を保持しているため、類似文書の場合にパース処理速度の向上が図れます。

(2) XML 電文のキャッシュ機能の設定

XML 電文のキャッシュ機能を使用するには、HCSC サーバランタイム定義ファイルの `xmltelegram-maxcache-num` プロパティで値を設定します。

`xmltelegram-maxcache-num` プロパティの詳細は、マニュアル「Cosminexus サービスプラットフォーム リファレンス」の「HCSC サーバランタイム定義ファイル」を参照してください。

7

障害調査のための機能

この章では、Cosminexus サービスプラットフォームで発生した障害を調査する機能について説明します。

7.1 障害情報の取得・障害回復・障害解析機能

7.2 データ検証機能

7.3 ユーザ電文トレース機能

7.1 障害情報の取得・障害回復・障害解析機能

7.1.1 障害情報の取得機能

Cosminexus サービスプラットフォームの運用環境には、運用・実行環境での障害の発生に備えて、トレース情報を保存する機能があります。トレース情報は、次に示すファイルに格納されます。

- メッセージログファイル
- リクエストトレースファイル
- 性能解析トレースファイル
- 保守情報（例外ログ、メンテナンスログ、メソッドトレースなど）

これらのトレース情報を確認することで、障害の発生時刻、発生場所、原因などを把握できます。

また、Cosminexus サービスプラットフォームの開発環境および運用環境の画面では、画面操作時に自動的に実行されるチェックによって出力されたメッセージは、コンソールビューに表示されます。コンソールビューに表示されるメッセージの内容を確認し、各種定義や設定を変更してから再度操作できます。

7.1.2 障害回復機能

次に示す運用時に発生した障害についても、コマンドで情報の確認、ファイルのバックアップ、環境の復旧などができます。

- システムのセットアップまたはアンセットアップ時
- オペレーション実行時
- サービス部品呼び出し実行時

7.1.3 障害解析機能

Cosminexus サービスプラットフォームでは、性能解析トレースファイルを利用して障害解析を補助するメッセージトラッキング機能を提供しています。メッセージトラッキングとは、障害発生箇所を特定し、障害の切り分けや障害調査を補助する障害解析ツールです。メッセージトラッキングを起動すると、専用の対話型プロンプトが表示されます。表示された対話型プロンプトでコマンドを実行し、性能解析トレースファイルを解析します。

メッセージトラッキング機能は、障害発生時の性能解析トレースファイルがあれば、uCosminexus Service Architect、uCosminexus Service Platform、またはuCosminexus Operator for Service Platform がインストールされているマシンならどこでも利用できます。そのため、障害が発生した HCSC サーバ稼働マシンだけでなく、障害が発生した HCSC サーバとは別のリモート環境でも障害を解析できます。

7.2 データ検証機能

7.2.1 データ検証機能の概要

サービスアダプタでは、要求電文や応答電文が、定義したフォーマット定義に対して妥当かどうかを検証できます。また、データ変換機能を利用している場合は、変換前後の電文が、定義したフォーマット定義に対して妥当かどうかを検証できます。

サービスアダプタに指定したフォーマット定義の形式と、要求時および応答時に送られてくる電文の形式が妥当であるかどうか検証し、電文がフォーマットの形式に対して妥当でない場合は、サービスを実行する前にエラーを返します。

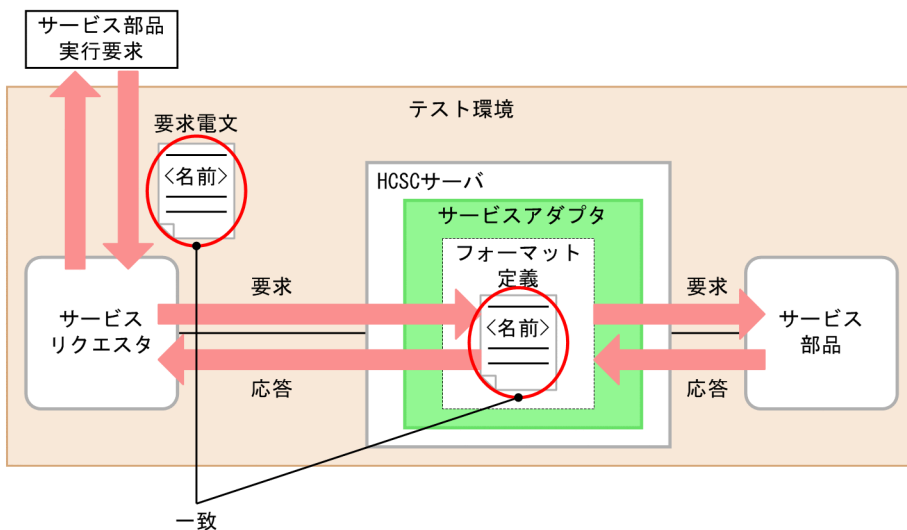
テスト環境などで、サービスを実行する前に妥当な電文かどうかを検証できるため、サービスの実行の失敗を事前に回避できます。

要求電文とフォーマット定義が一致する場合と一致しない場合の電文の流れを次の図に示します。

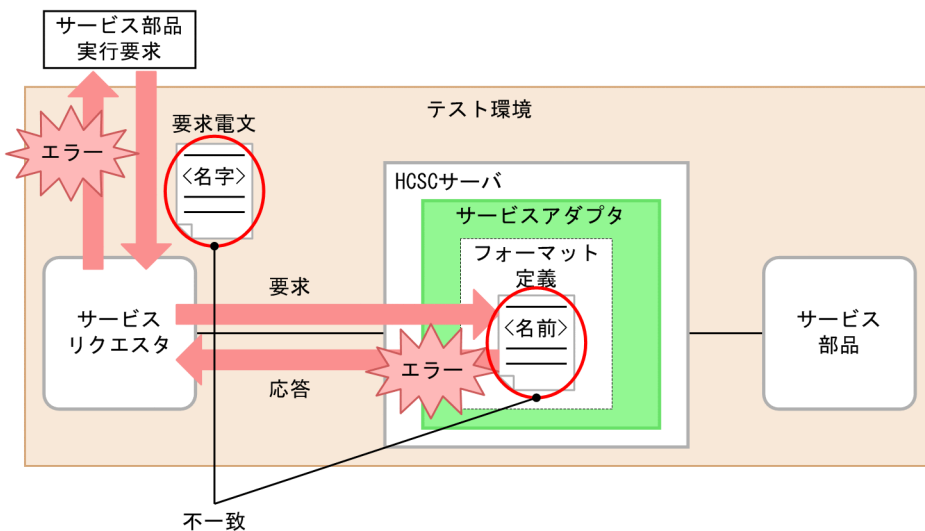
7. 障害調査のための機能

図 7-1 要求電文とフォーマット定義が一致する場合と一致しない場合の電文の流れ

●要求電文がフォーマット定義と一致する場合



●要求電文がフォーマット定義と一致しない場合



(凡例)

➡ : サービス部品呼び出し要求の流れ

7.2.2 データ検証機能の設定

データ検証機能を使用するには、HCSC サーバランタイム定義ファイルの telegram-validation プロパティで値を設定します。

telegram-validation プロパティの詳細は、マニュアル「Cosminexus サービスプラットフォーム」

フォームリファレンスの「HCSC サーバランタイム定義ファイル」を参照してください。

7.2.3 データ検証機能の有効範囲

データ検証機能の有効範囲を次の表に示します。なお、データ検証機能が有効な場合、検証の対象はHCSCサーバ全体になります。

表 7-1 データ検証機能の有効範囲

アダプタ種別	電文種別	要求 / 応答	アダプタでのデータ変換有無	検証機能有効 / 無効	検証される電文
サービスアダプタ	XML	要求	データ変換なし	有効	サービス部品電文
			標準電文 サービス部品電文へのデータ変換あり	有効	標準電文, サービス部品電文
		応答	データ変換なし	有効	サービス部品電文
			サービス部品電文 標準電文へのデータ変換あり	有効	標準電文, サービス部品電文
		フォルト	-	無効	-
	バイナリ	要求	データ変換なし	無効	-
			標準電文 サービス部品電文へのデータ変換あり	無効	-
		応答	データ変換なし	無効	-
			サービス部品電文 標準電文へのデータ変換あり	無効	-
	DB アダプタ	XML	要求	データ変換なし	無効
標準電文 サービス部品電文へのデータ変換あり				無効	-
応答			データ変換なし	無効	-
			サービス部品電文 標準電文へのデータ変換あり	無効	-

(凡例) - : 該当しない

電文の検証機能が有効になるアダプタの詳細については、マニュアル「Cosminexus サービスプラットフォーム システム構築・運用ガイド」の「7.7.5 ユーザ電文に誤りがある場合の障害対策」を参照してください。

7.3 ユーザ電文トレース機能

7.3.1 ユーザ電文トレース機能の概要

ユーザ電文トレース機能は、HCSC サーバ上を流れる電文の内容をユーザ電文トレースファイルに出力する機能です。

次の内容をユーザ電文トレースとして取得できます。

- リクエスト受付で受け付けたサービス部品呼び出し要求または応答の電文
- ビジネスプロセスを呼び出した時の要求または応答の電文
- サービスアダプタからサービス部品を呼び出したときの要求または応答の電文
- データ変換を行ったときの変換前または変換後の電文

ユーザ電文トレース機能を使用することによって、設計したとおりの電文が流れているかを確認できます。また、各ポイント（リクエスト側とサービス側、BP のアクティビティ、データ変換など）も確認できるため、開発の容易性が増します。

7.3.2 ユーザ電文トレース機能の設定

ユーザ電文トレース機能を使用するには、次に示す HCSC サーバランタイム定義ファイルのプロパティで値を設定します。

- telegramtrace プロパティ
- telegramtrace-filepath プロパティ
- telegramtrace-filesize プロパティ
- telegramtrace-filename プロパティ

各プロパティの詳細は、マニュアル「Cosminexus サービスプラットフォーム リファレンス」の「HCSC サーバランタイム定義ファイル」を参照してください。

ユーザ電文トレースを取得する設定にすると、HCSC サーバ全体に対して有効となります。

7.3.3 ユーザ電文トレース機能使用時の注意事項

(1) セキュリティ面の考慮について

ユーザ電文トレース機能は、電文の内容をファイルに出力します。そのため、電文の内容が情報漏洩する懸念があり、セキュリティ上の問題が発生します。したがって、本番運用を含む通常の運用では使用しないでください。

また、出力したユーザ電文トレースファイルの取り扱いに注意してください。

(2) DB アダプタでの出力について

Cosminexus Service Coordinator 01-60 以前の「<Cosminexus のインストールディレクトリ>¥CSC¥lib¥cscdba.ear」を使用して作成した DB アダプタでは、取得位置 SVC、および取得位置 CNVST の電文トレースは出力されません。

8

他製品と連携して利用する機能

この章では、JP1 と連携した運用管理，および接続できるデータベースについて説明します。

8.1 JP1 と連携した運用管理

8.2 接続できるデータベース

8.1 JP1 と連携した運用管理

JP1 と連携して Cosminexus サービスプラットフォームを運用すると、業務システム全体の監視、問題の検知などの運用を効率良くできるようになります。また、サーバやアプリケーションの起動・停止を自動化することで、システムの日常運用の効率化を図れます。

JP1 と連携したシステムの運用については、マニュアル「Cosminexus アプリケーションサーバ 機能解説 運用 / 監視 / 連携編」の「12.2 JP1 との連携」も参照してください。

8.1.1 JP1/IM との連携による障害検知

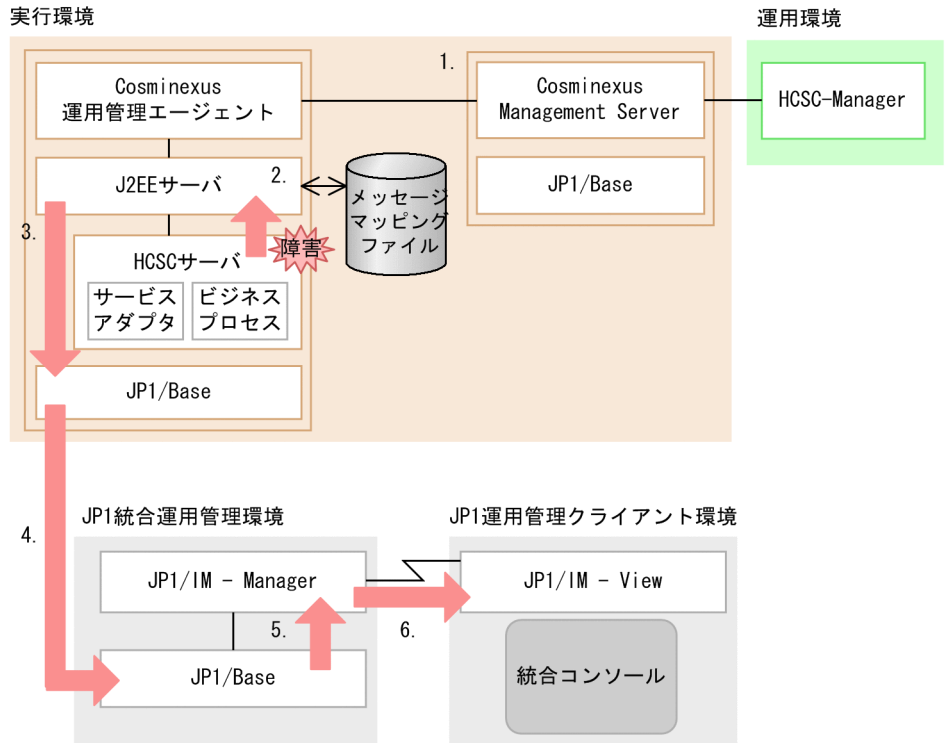
システムで発生した障害を JP1 イベントによって集中管理することができます。

Cosminexus サービスプラットフォームでは JP1/IM - Manager と連携し、システムで発生した障害を検知して、障害メッセージを JP1 イベントとして JP1 へ通知できます。また、JP1/IM - View では、JP1/IM - Manager のホストに接続して、実行環境から発行された JP1 イベントを表示できます。JP1 イベントとして発行されるメッセージは、ユーザの環境に合わせてカスタマイズできます。



どの障害が発生した場合に、どのメッセージが通知されるかをあらかじめ設定しておくことで、JP1/IM - View で表示されるメッセージログを監視して、障害発生時の障害の種類や原因などの調査を簡略化できます。

JP1/IM と連携した障害監視の概要を次の図に示します。

図 8-1 JP1/IM と連携した障害監視の概要



(凡例)

-  : JP1/IMが制御する画面
-  : 障害情報の流れ

1. Cosminexus Management Server で、JP1/IM で監視する JP1 イベント発行を設定します。
2. 実行環境で発生した障害のメッセージが JP1 イベントに変換されます。
3. JP1/Base を使用して、システムで発生した障害のメッセージが JP1 イベントとして発行されます。
4. JP1 イベントが JP1 統合運用管理環境に転送されます。
5. JP1 統合運用管理環境に転送されてきた JP1 イベントが、JP1/IM - Manager によって収集されます。
6. 統合コンソールから JP1/IM - View を使用し、JP1/IM - Manager に接続して、JP1 イベントの詳細を確認できます。

8.1.2 JP1 によるプロセス監視

Cosminexus サービスプラットフォームのシステムを JP1 で監視する場合の推奨プロセス

8. 他製品と連携して利用する機能

スについて説明します。

(1) 監視推奨プロセス

HCSC サーバ起動時の各論理サーバのプロセスは、Cosminexus Manager で監視されていて、プロセスダウンやハングアップなどの障害が発生すると、論理サーバを自動で再起動します。そのため、Cosminexus サービスプラットフォームのシステムを JP1 で監視する場合は、次のプロセスを監視の対象とすることを推奨します。

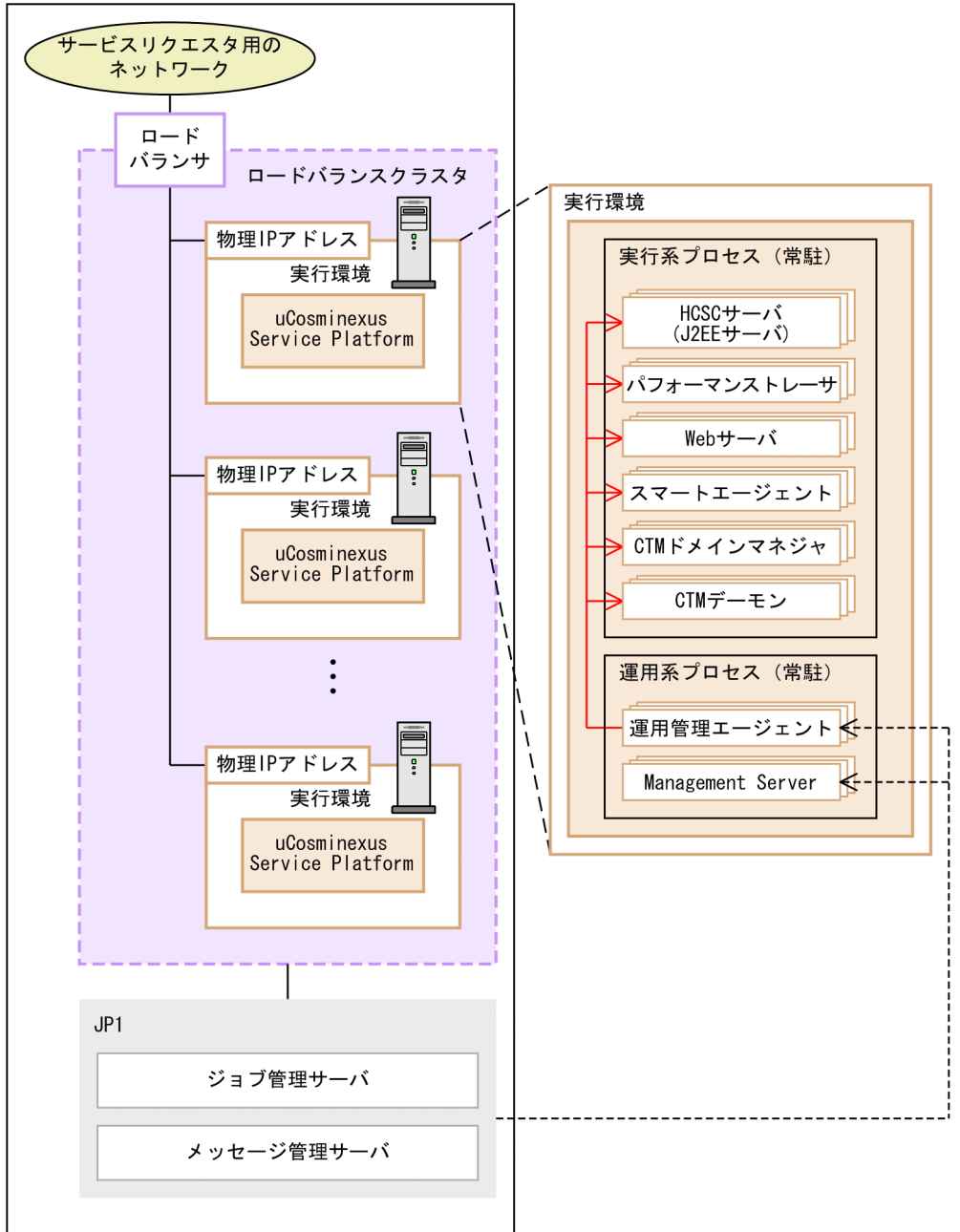
- Management Server
- 運用管理エージェント

ポイント

各論理サーバのプロセスのうち、上記推奨プロセス以外のプロセスを JP1 で監視すると、プロセスの二重監視になります。

HCSC サーバ起動時のプロセス構成を次の図に示します。

図 8-2 HCSC サーバ起動時のプロセス構成



(凡例)

→ : 運用管理エージェントからの監視

----> : JP1からの監視

☐ : プロセス

8. 他製品と連携して利用する機能

JP1 の監視推奨となるプロセス名を次の表に示します。

表 8-1 JP1 からの監視推奨となるプロセス名

監視推奨プロセス	プロセス名	
	Windows の場合	UNIX の場合
Management Server	cjstartweb.exe	cjstartweb
運用管理エージェント	adminagent.exe	adminagent

また、データベースを使用している場合は、使用しているデータベースのプロセスも JP1 の監視推奨プロセスとなります。それらのプロセス名については、各データベースのマニュアルを参照してください。

(2) プロセスを監視するときの注意事項

プロセス数の上限と下限で監視する場合、上限を 1、下限を 1 で監視すると、一時的に cjstartweb や adminagent のプロセスが 1 より大きくなり、障害を検知してしまうことがあります。Management Server、運用管理エージェントについては複数起動を抑止する機能があるため、Cosminexus Manager の推奨としては上限なしで監視してください。

8.2 接続できるデータベース

ここでは、Cosminexus サービスプラットフォームから接続できるデータベースおよび接続に使用する JDBC ドライバについて説明します。データベースごとに接続に使用する JDBC ドライバが異なります。なお、使用する機能によって接続できるデータベースおよび使用する JDBC ドライバが異なります。

Cosminexus サービスプラットフォームから接続できるデータベース、JDBC ドライバおよび接続に使用する機能の対応を、次の表に示します。

表 8-2 Cosminexus サービスプラットフォームから接続できるデータベース

データ ベース	JDBC ドライバ	メッセージング機能およびビジ ネスプロセス機能		DB アダプタ機能	
		ローカルト ランザクシ ョン	グローバ ルトラン ザクシ ョン	ローカルト ランザクシ ョン	グローバ ルトラン ザクシ ョン
HiRDB Version 9	HiRDB Type4 JDBC Driver				
	Cosminexus DABroker Library	-	-	-	-
HiRDB Version 8	HiRDB Type4 JDBC Driver				
	Cosminexus DABroker Library				
Oracle 11g R2 11.2	Oracle JDBC Thin Driver				
	Cosminexus DABroker Library	-	-	-	-
Oracle 11g R1 11.1	Oracle JDBC Thin Driver				
	Cosminexus DABroker Library	-	-	-	-
Oracle 10g R2 10.2.0	Oracle JDBC Thin Driver				
	Cosminexus DABroker Library	-	-		

(凡例)

- : 使用できます。かつ、使用を推奨します。
- : 使用できます。
- : 使用できません。

付録

付録 A このマニュアルの参考情報

付録 B 用語解説

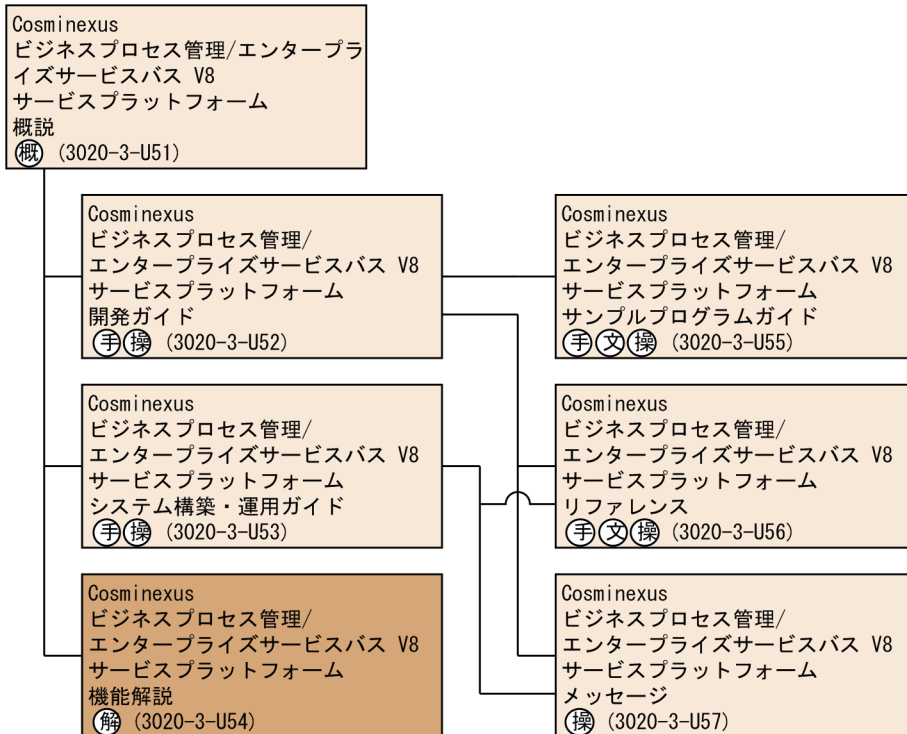
付録 A このマニュアルの参考情報

このマニュアルを読むに当たっての参考情報を示します。

付録 A.1 関連マニュアル

このマニュアルの関連マニュアルを、次の図に示します。必要に応じてお読みください。

図 A-1 このマニュアルの関連マニュアル



<記号>

- (概) : 概説書
- (解) : 解説書
- (手) : 手引書
- (文) : 文法書
- (操) : 操作書

また、図に示したマニュアル以外で、このマニュアルと関連するマニュアルを次に示します。必要に応じてお読みください。

Cosminexus アプリケーションサーバ関連

- Cosminexus アプリケーションサーバ V8 概説 (3020-3-U01)

- Cosminexus アプリケーションサーバ V8 ファーストステップガイド (3020-3-U02)
- Cosminexus アプリケーションサーバ V8 システム設計ガイド (3020-3-U03)
- Cosminexus アプリケーションサーバ V8 システム構築・運用ガイド (3020-3-U04)
- Cosminexus アプリケーションサーバ V8 機能解説 基本・開発編 (Web コンテナ)
(3020-3-U05)
- Cosminexus アプリケーションサーバ V8 機能解説 基本・開発編 (EJB コンテナ)
(3020-3-U06)
- Cosminexus アプリケーションサーバ V8 機能解説 基本・開発編 (コンテナ共通機能)
(3020-3-U07)
- Cosminexus アプリケーションサーバ V8 機能解説 拡張編 (3020-3-U08)
- Cosminexus アプリケーションサーバ V8 機能解説 運用 / 監視 / 連携編 (3020-3-U09)
- Cosminexus アプリケーションサーバ V8 機能解説 保守 / 移行 / 互換編 (3020-3-U10)
- Cosminexus アプリケーションサーバ V8 アプリケーション設定操作ガイド
(3020-3-U12)
- Cosminexus アプリケーションサーバ V8 運用管理ポータル操作ガイド (3020-3-U13)
- Cosminexus アプリケーションサーバ V8 リファレンス コマンド編 (3020-3-U14)
- Cosminexus アプリケーションサーバ V8 リファレンス 定義編 (サーバ定義)
(3020-3-U15)
- Cosminexus アプリケーションサーバ V8 リファレンス 定義編 (アプリケーション / リ
ソース定義) (3020-3-U16)
- Hitachi Web Server (3020-3-U17)
- Cosminexus アプリケーションサーバ V8 仮想化システム構築・運用ガイド
(3020-3-U18)
- TPBroker Version 5 トランザクショナル分散オブジェクト基盤 TPBroker ユーザーズ
ガイド (3020-3-U19)
- TPBroker Version 5 トランザクショナル分散オブジェクト基盤 TPBroker 運用ガイド
(3020-3-U20)
- Cosminexus アプリケーションサーバ V8 Cosminexus Reliable Messaging
(3020-3-U21)
- Cosminexus アプリケーションサーバ V8 アプリケーション開発ガイド (3020-3-U25)
- Cosminexus アプリケーションサーバ V8 リファレンス API 編 (3020-3-U26)
- Cosminexus アプリケーションサーバ V8 Cosminexus XML Processor ユーザーズガイ
ド (3020-3-U27)
- VisiBroker Version 5 Borland(R) Enterprise Server VisiBroker(R) デベロッパーズガ
イド (3020-3-U28)
- VisiBroker Version 5 Borland(R) Enterprise Server VisiBroker(R) プログラマーズリ
ファレンス (3020-3-U29)
- Cosminexus アプリケーションサーバ V8 SOAP アプリケーション開発の手引
(3020-3-U30)
- Cosminexus アプリケーションサーバ V8 Web サービス開発の手引 (3020-3-U31)

- Cosminexus アプリケーションサーバ V8 Web サービスセキュリティ 使用の手引 (3020-3-U32)
- Cosminexus アプリケーションサーバ V8 Cosminexus XML Security - Core ユーザーズガイド (3020-3-U33)
- Cosminexus アプリケーションサーバ V8 メッセージ 1 KDAL-KDCG および Hitachi Web Server 編 (3020-3-U41)
- Cosminexus アプリケーションサーバ V8 メッセージ 2 KDJE-KDJW 編 (3020-3-U42)
- Cosminexus アプリケーションサーバ V8 メッセージ 3 KECX-KEDT / KEOS02000-29999 / KEUC-KFRM 編 (3020-3-U43)
- Cosminexus アプリケーションサーバ V8 メッセージ 4 監査ログ編 (3020-3-U44)
- Cosminexus ビジネスプロセス管理 / エンタープライズサービスバス V8 サービスプラットフォーム TP1 アダプタ ユーザーズガイド (3020-3-U58)

その他

- 分散トランザクション処理機能 TP1/Server Base Enterprise Option 使用の手引 (3000-3-982)
- uCosminexus TP1 Gateway ユーザーズガイド (3000-3-989)
- Cosminexus サービスプラットフォーム ファイルアダプタ ユーザーズガイド (3020-3-P06)

なお、このマニュアルでは、次のマニュアルについて、名称を省略して表記しています。マニュアルの正式名称とこのマニュアルでの表記を次の表に示します。

正式名称	このマニュアルでの表記
Cosminexus ビジネスプロセス管理 / エンタープライズサービスバス V8 サービスプラットフォーム 開発ガイド	Cosminexus サービスプラットフォーム 開発ガイド
Cosminexus ビジネスプロセス管理 / エンタープライズサービスバス V8 サービスプラットフォーム システム構築・運用ガイド	Cosminexus サービスプラットフォーム システム構築・運用ガイド
Cosminexus ビジネスプロセス管理 / エンタープライズサービスバス V8 サービスプラットフォーム リファレンス	Cosminexus サービスプラットフォーム リファレンス
Cosminexus アプリケーションサーバ V8 機能解説 基本・開発編 (Web コンテナ)	Cosminexus アプリケーションサーバ 機能解説 基本・開発編 (Web コンテナ)
Cosminexus アプリケーションサーバ V8 機能解説 基本・開発編 (EJB コンテナ)	Cosminexus アプリケーションサーバ 機能解説 基本・開発編 (EJB コンテナ)
Cosminexus アプリケーションサーバ V8 機能解説 運用 / 監視 / 連携編	Cosminexus アプリケーションサーバ 機能解説 運用 / 監視 / 連携編
Cosminexus アプリケーションサーバ V8 アプリケーション設定操作ガイド	Cosminexus アプリケーションサーバ アプリケーション設定操作ガイド

正式名称	このマニュアルでの表記
Cosminexus アプリケーションサーバ V8 運用管理ポータル操作ガイド	Cosminexus アプリケーションサーバ 運用管理ポータル操作ガイド
Cosminexus アプリケーションサーバ V8 リファレンス コマンド編	Cosminexus アプリケーションサーバ リファレンス コマンド編
Cosminexus アプリケーションサーバ V8 リファレンス 定義編 (サーバ定義)	Cosminexus アプリケーションサーバ リファレンス 定義編 (サーバ定義)
Cosminexus アプリケーションサーバ V8 リファレンス 定義編 (アプリケーション/リソース定義)	Cosminexus アプリケーションサーバ リファレンス 定義編 (アプリケーション/リソース定義)
Cosminexus アプリケーションサーバ V8 SOAP アプリケーション開発の手引	Cosminexus アプリケーションサーバ SOAP アプリケーション開発の手引
Cosminexus アプリケーションサーバ V8 Web サービス開発の手引	Cosminexus アプリケーションサーバ Web サービス開発の手引

付録 A.2 このマニュアルでの表記

(1) 製品名

このマニュアルで使用する表記と、対応する製品名を次に示します。

表記	製品名	
Eclipse	Eclipse Classic 3.6.1	
HiRDB または HiRDB サーバ	HiRDB/Parallel Server	HiRDB/Parallel Server Version 8
	HiRDB/Single Server	HiRDB/Single Server Version 8
	HiRDB Server	HiRDB Server Version 9
IPF	Itanium(R) Processor Family	
JP1	JP1 関連製品	
JP1/AJS2	JP1/AJS2 - Agent	JP1/Automatic Job Management System 2 - Agent
	JP1/AJS2 - Light Edition	JP1/Automatic Job Management System 2 - Light Edition
	JP1/AJS2 - Manager	JP1/Automatic Job Management System 2 - Manager
	JP1/AJS2 - View	JP1/Automatic Job Management System 2 - View
JP1/IM	JP1/IM - Manager	JP1/Integrated Management - Manager
	JP1/IM - View	JP1/Integrated Management - View
Oracle	Oracle 10g	Oracle 10gR2

表記		製品名
	Oracle 11g	Oracle Database 11g
SAP R/3		SAP R/3(R)
TP1/EE		TP1/Server Base Enterprise Option
UNIX	AIX	AIX 5L V5.3
		AIX V6.1
		AIX V7.1
	HP-UX	HP-UX 11i V2 (IPF)
		HP-UX 11i V3 (IPF)
	Linux (IPF)	Red Hat Enterprise Linux(R) AS 4 (IPF)
		Red Hat Enterprise Linux(R) 5 (Intel Itanium(R))
		Red Hat Enterprise Linux(R) 5 Advanced Platform (Intel Itanium(R))
	Linux または Linux (x86 / AMD64 & Intel EM64T)	Red Hat Enterprise Linux(R) AS 4 (x86)
		Red Hat Enterprise Linux(R) 5 Advanced Platform (x86)
		Red Hat Enterprise Linux(R) ES 4 (x86)
		Red Hat Enterprise Linux(R) 5 (x86)
		Red Hat Enterprise Linux(R) AS 4 (AMD64 & Intel EM64T)
		Red Hat Enterprise Linux(R) 5 Advanced Platform (AMD/Intel 64)
		Red Hat Enterprise Linux(R) ES 4 (AMD64 & Intel EM64T)
Red Hat Enterprise Linux(R) 5 (AMD/Intel 64)		
Red Hat Enterprise Linux(R) Server 6 (32-bit x86)		
Red Hat Enterprise Linux(R) Server 6 (64-bit x86_64)		

(2) Cosminexus の機能名

このマニュアルで使用する表記と、対応する Cosminexus の機能名を次に示します。

表記	Cosminexus の機能名
Cosminexus RM	Cosminexus Reliable Messaging
CTM	Cosminexus Component Transaction Monitor
HCSC	Hitachi Cosminexus Service Coordinator
HCSC-Business Process , ビジネスプロセス基盤 , または BP 基盤	Hitachi Cosminexus Service Coordinator - Business Process
HCSC-Data Transform またはデータ変換基盤	Hitachi Cosminexus Service Coordinator - Data Transform
HCSC-DB Adapter または DB アダプタ	Hitachi Cosminexus Service Coordinator - Database Adapter
HCSC-Manager または HCSC-MNG	Hitachi Cosminexus Service Coordinator - Manager
HCSC-Messaging , HCSC-MSG , またはメッセージング基盤	Hitachi Cosminexus Service Coordinator - Messaging
HCSC-TE	Hitachi Cosminexus Service Coordinator Tools for Eclipse
Management Server	Cosminexus Management Server
MyEclipse	MyEclipse for Cosminexus
PRF	Cosminexus Performance Tracer
TPBroker	Cosminexus TPBroker

(3) Java 関連用語

このマニュアルで使用する表記と、対応する Java 関連用語を次に示します。

表記	Java 関連用語
Connector 1.5	J2EE™ Connector Architecture 1.5
EAR	Enterprise ARchive
EJB または Enterprise JavaBeans	Enterprise JavaBeans
J2EE	J2EE™
	Java™ 2 Platform, Enterprise Edition
J2SE	Java™ 2 Platform, Standard Edition
Java	Java™
Java 2 SDK	Java™ 2 Software Development Kit, Standard Edition
JAXP	Java™ API for XML Processing
JDBC	Java™ Database Connectivity
JMS	Java™ Message Service
JSP	JavaServer Pages™

表記	Java 関連用語
Servlet またはサーブレット	Java™ Servlet

付録 A.3 英略語

このマニュアルで使用する英略語を次に示します。

英略語	英字での表記
API	Application Programming Interface
BP	Business Process
BPEL	Business Process Execution Language
BPM	Business Process Management
BPMN	Business Process Modeling Notation
DB	Database
ERP	Enterprise Resource Planning
ESB	Enterprise Service Bus
ETL	Extract Transform Loading
HTML	Hyper Text Markup Language
HTTP	Hyper Text Transfer Protocol
IIOB	Internet Inter-Orb Protocol
IT	Information Technology
JSF	JavaServer Faces
JSP	JavaServer Pages
MBean	Managed Bean
MDB	Message-Driven Bean
OS	Operating System
RDB	Relational Database
RMI	Remote Method Invocation
RPC	Remote Procedure Call
SaaS	Software as a Service
SOA	Service Oriented Architecture
SOAP	Simple Object Access Protocol
SSL	Secure Sockets Layer
W3C	World Wide Web Consortium
WFA	Work-Flow Architecture
WS-R	Web Services Reliability
WSDL	Web Service Description Language

英略語	英字での表記
WST	Web Standard Tools
XML	Extensible Markup Language

付録 A.4 KB (キロバイト) などの単位表記について

1KB (キロバイト), 1MB (メガバイト), 1GB (ギガバイト), 1TB (テラバイト) はそれぞれ $1,024$ バイト, $1,024^2$ バイト, $1,024^3$ バイト, $1,024^4$ バイトです。

付録 B 用語解説

(英字)

BPEL (Business Process Execution Language)

XMLをベースにしたワークフロー記述言語です。Cosminexus サービスプラットフォームでは、アクティビティとコネクションで定義したビジネスプロセスを BPEL 形式で保存できます。

BPMN (Business Process Modeling Notation)

業務手順をわかりやすく図示して可視化するための表記ルールを定めたものです。ビジネスプロセス・モデリング表記法ともいいます。

CTM (Cosminexus Component Transaction Monitor)

構成ソフトウェアの一つです。サービス部品の実行要求が集中しないようにスケジューリングをするための機能です。

DB Connector

データベースに接続するためのリソースアダプタです。

DB アダプタ

データベースの操作をサービス部品として利用する場合に、操作するデータベースや実行する SQL を定義した HCSC コンポーネントです。

DB キュー

データベースを介してシステム間で連携する、TP1/EE (TP1/Server Base Enterprise Option) の機能です。DB キュー機能を使用すると、データベースをキューとして、TP1/EE と J2EE との間で、非同期型のメッセージ送受信ができます。

EAR (Enterprise ARchive) ファイル

J2EE アプリケーションを構成する複数のファイルを EAR ファイル形式でパッケージ化したものです。

Cosminexus サービスプラットフォームでは、HCSC コンポーネントをパッケージングして、EAR ファイルを作成します。

Eclipse

Eclipse プロジェクト (eclipse.org) が提供するオープンソースの統合開発環境です。ソースコードの編集支援機能やデバッグ機能など、アプリケーションの開発効率を向上させる各種機能を備えています。

Cosminexus サービスプラットフォームでは、Eclipse に必要な機能をプラグインして利用します。

EJB (Enterprise JavaBeans)

業務ロジックをプログラムとして記述したビジネスロジック、および永続的データを格納するオブジェクトであるエンティティを Java コンポーネント化したものです。Oracle Corporation から EJB 仕様が公開されています。

FTP アダプタ

Cosminexus サービスプラットフォームと FTP サーバとの間でファイルの送受信をするために利用するアダプタです。

FTP インバウンドアダプタ

FTP 連携で使用されるリソースアダプタです。Connector 1.5 仕様準拠しています。

FTP 受付

FTP インバウンドアダプタを経由して FTP クライアントからの実行要求を受け付ける機能（インターフェース）です。

ファイル操作アダプタや FTP アダプタと連携することで、Cosminexus サービスプラットフォーム上で FTP クライアントと FTP サーバの間のファイル転送ができるようになります。

FTP クライアント

FTP を使用してファイルの送受信を行うクライアント用ソフトウェアです。

FTP サーバ

FTP クライアントからの要求によってファイルを送受信するサーバ用ソフトウェアです。

FTP 連携

Cosminexus サービスプラットフォーム上で FTP クライアントと FTP サーバの間のファイル転送を実現するための機能です。

HCSC (Hitachi Cosminexus Service Coordinator)

uCosminexus Service Platform を構成する構成ソフトウェアの一つです。Cosminexus サービスプラットフォームで SOA を適用したシステムを構築、運用するための機能として、ビジネスプロセス実行機能、データ変換機能、DB アダプタ機能、実行環境の運用管理機能、およびメッセージング制御機能を提供しています。

HCSC-Business Process (ビジネスプロセス基盤)

ビジネスプロセスの定義に従って、適切なサービス部品を呼び出すための機能です。

HCSC-Data Transform (データ変換基盤)

データ変換定義に従って、要求電文・応答電文の電文フォーマットを変換する機能です。

HCSC-DB Adapter (DB アダプタ)

DB アダプタの定義に従って適切なサービス部品（データベースの操作）を呼び出す機能です。

HCSC-Manager

Cosminexus サービスプラットフォームの運用環境で利用する運用管理機能です。

HCSC サーバのセットアップ機能、HCSC コンポーネントの配備機能、プロセスインスタンスの実行履歴管理機能などが含まれます。

HCSC-Manager コマンド共通定義ファイル (csccmd.properties)

運用環境で使用するコマンドの必須項目の省略値を定義するファイルです。J2SE のプロパティ形式のファイルとして作成します。

このファイルに省略値を定義しておくと、コマンド入力時に該当するオプションの入力を省略でき

ます。

HCSC-Manager 定義ファイル (cscmng.properties)

運用環境のカスタマイズに必要な情報を定義するファイルです。J2SE のプロパティ形式のファイルとして作成します。

リポジトリのインポート先となるディレクトリや、ログファイルの大きさ・面数などを定義できます。

HCSC-Messaging (メッセージング基盤)

サービスアダプタの定義に従って適切なサービス部品を呼び出す機能です。

HCSCTE

Cosminexus サービスプラットフォームの開発環境で利用する開発支援機能です。HCSCTE で提供される機能は、Eclipse にプラグインして利用します。

アダプタやビジネスプロセスの作成・定義機能などが含まれます。

HCSC コンポーネント

開発環境で作成するサービスアダプタ、DB アダプタ、ビジネスプロセス、およびユーザ定義受付の総称です。

HCSC コンポーネントの検証

定義したアダプタ、ビジネスプロセス、データ変換定義の内容が妥当かどうかを検証する機能です。必要な定義があるか、また定義の関係が正しいかを、必要に応じて任意のタイミングで検証できます。

HCSC サーバ

実行環境でサービス部品の実行を管理するサーバ機能です。メッセージング基盤、ビジネスプロセス基盤、データ変換基盤など、実行環境に必要な機能が含まれます。また、標準受付、ユーザ定義受付などのリクエスト受付機能や、サービスアダプタ (Web サービス, SessionBean, MDB (WS-R), MDB (DB キュー), DB アダプタ) の各種アダプタ機能も含まれます。

HCSC サーバ構成定義ファイル (cscsvsetup.xml)

HCSC サーバの構成を定義するファイルです。XML 形式のファイルとして作成します。

HCSC サーバ、クラスタ、Cosminexus Manager などの情報を定義できます。

HCSC サーバセットアップ定義ファイル (cscsvsetup.properties)

HCSC サーバのセットアップに必要な情報を定義するファイルです。J2SE のプロパティ形式のファイルとして作成します。

HCSC サーバに設定するクラスタ名や使用できる標準受付の種類などを定義できます。

HCSC サーバランタイム定義ファイル (cscsvconfig.properties)

HCSC サーバの起動時に必要なランタイム情報を定義するファイルです。J2SE のプロパティ形式のファイルとして作成します。

実行履歴を採取するかどうかや、トレースファイルのレベル・出力先などを定義できます。

HTTP

インターネットで、WWW サーバと WWW クライアントの間で HTML 文書を送受信するための通

信プロトコルです。

iWay アダプタ

Cosminexus サービスプラットフォームの実行環境から他社 ERP パッケージのサービス部品を呼び出す場合に利用するアダプタです。

J2EE サーバ

アプリケーションを実行するためのサーバ基盤である J2EE コンテナを生成、実行する環境です。Cosminexus サービスプラットフォームは、J2EE サーバとして Cosminexus Component Container を使用します。

JAX-WS エンジン

Cosminexus が提供する Web サービスの通信基盤です。SOAP1.1 仕様および SOAP1.2 仕様に対応しています。

JDBC (Java Database Connectivity)

Java のプログラムでデータベースを利用するための API です。

JavaSoft が 1996 年前半に仕様を公開しました。Java プログラムから、JDBC API を経由して「JDBC ドライバ」を呼び出し、ドライバがデータベース・サーバへの接続機能を提供します。

JP1

日立の統合システム運用管理ソフトウェアの総称です。

MBean (Managed Bean)

管理対象となるリソースを表す Java オブジェクトです。JMX 仕様に基づいて実装されます。マネージャに対するリソースのインターフェースとして機能し、リソースに関する情報や操作をマネージャに提供します。

MDB (Message-Driven Bean)

JMS と連携するメッセージ駆動タイプの Bean です。EJB コンテナは JMS の Destination からの JMS メッセージ受信を契機に Bean を起動します。

Message Queue アダプタ

Cosminexus サービスプラットフォームの実行環境から既存のメッセージキュー (IBM WebSphere MQ システム) に対してメッセージの送受信をする場合に利用するアダプタです。

MyEclipse

Java アプリケーションの開発環境です。MyEclipse には、Eclipse でアプリケーションを開発するときに使用するプラグインがまとめられています。MyEclipse を使用すると、アプリケーションの作成、J2EE サーバへのアプリケーションのデプロイ、J2EE サーバの起動・停止、データベースの操作などができます。

Cosminexus では、構成ソフトウェアとして、MyEclipse の日本語版環境である MyEclipse for Cosminexus を提供しています。

Object Access アダプタ

Cosminexus サービスプラットフォームの実行環境から既存の TPBroker システム (Object

Wrapper システム) のサービス部品を呼び出す場合に利用するアダプタです。

RMI-IIOP

JavaRMI と JavaIDL を統合した API です。

SessionBean

セッション単位での処理を行う機能がある、EJB で規定されている Bean の一つです。

SOAP (Simple Object Access Protocol)

ネットワーク上のサービスを呼び出す際のメッセージを規定したものです。HTTP と XML がベースとなっています。

SOAP1.1/1.2 併用モード

利用する Web サービスの標準仕様および通信基盤を表すモードです。SOAP1.1 仕様および SOAP1.2 仕様に対応したサービス部品 / サービスリクエストを利用するときに指定します。SOAP1.1/1.2 併用モードでは、通信基盤として Cosminexus が提供する JAX-WS エンジンを利用します。

SOAP1.1 モード

利用する Web サービスの標準仕様および通信基盤を表すモードです。SOAP1.1 仕様に対応したサービス部品 / サービスリクエストを利用するときに指定します。SOAP1.1 モードでは、通信基盤として Cosminexus が提供する SOAP 通信基盤を利用します。

SOAP 通信基盤

Cosminexus が提供する Web サービス (SOAP アプリケーション) の通信基盤です。SOAP1.1 仕様に対応しています。

SQL オペレーション定義ファイル

DB アダプタを利用してデータベースにアクセスする場合に、実行する SQL を定義するファイルです。DB アダプタ用の XML フォーマット定義ファイルを作成するときの基となるファイルです。

TP1/RPC 受付

既存の OpenTP1 システム内にあるサービスリクエストからのサービス部品呼び出し要求を受け付けるための機能 (インターフェース) です。

TP1 アダプタ

Cosminexus サービスプラットフォームの実行環境から既存の OpenTP1 システム内にあるサービス部品を呼び出す場合に利用するアダプタです。

TP1 インバウンドアダプタ

TP1 インバウンド連携機能で使用するリソースアダプタです。Connector 1.5 仕様に準拠していません。

TP1 連携システム

OpenTP1 システム内にあるサービスリクエストおよびサービス部品と接続するためのシステム構成です。

uCosminexus Service Architect

Cosminexus サービスプラットフォームの開発環境を構築・運用するための製品です。

uCosminexus Service Platform

Cosminexus サービスプラットフォームの実行環境・運用環境を構築・運用するための製品です。

W3C (World Wide Web Consortium)

XML や DOM を含む Web 関連技術の標準化を推進する非営利団体です。

Web サービス

インターネット上に分散するアプリケーションが提供するサービスを、クライアントに API として提供するサービスです。

WFA (Work-Flow Architecture)

経済産業省がエンタープライズアーキテクチャで提案している業務フローの記法です。業務にかかわる組織や担当者、業務の流れ、および流れる情報を明示できます。

WS-R (Web Services Reliability)

標準化団体 OASIS で標準化された、インターネット向けのオープンな非同期通信プロトコルです。

WSDL (Web Service Description Language)

Web サービスを記述するための言語仕様です。XML がベースとなっています。

XML スキーマ

W3C 勧告で定められた仕様で、XML 文書内の論理的なデータ構造を規定するための仕組みです。

XML フォーマット定義ファイル

XML 形式のデータを扱う場合に利用する電文フォーマット定義ファイルです。XML スキーマファイルとして作成します。

(ア行)

アクティビティ

ビジネスプロセスの処理フローを定義する部品です。基本アクティビティと構造アクティビティに分類されます。複数のアクティビティを画面上に配置し、コネクションで連結して、ビジネスプロセスの処理フローを定義します。

アダプタ

サービスリクエストからの要求電文を受け取り、アダプタ内で定義されているサービス部品へ要求電文を送信して、サービス部品を呼び出すアプリケーションです。
アダプタには、標準で使用するサービスアダプタ（標準アダプタ）と、ユーザが必要に応じて使用するカスタムアダプタ（DB アダプタ、TP1 アダプタなど）があります。サービスアダプタは、Web サービス、SessionBean、MDB（WS-R）、および MDB（DB キュー）のサービス部品を呼び出すアダプタです。カスタムアダプタは、それ以外のサービス部品を呼び出すアダプタです。

受付処理

サービス部品の呼び出し要求を受け付けたり、電文を転送したりする処理を記述した API です。ユーザが作成した受付処理をカスタム受付フレームワーク上で動作させることで、任意のプロトコルに対応した受付を開発できます。

運用環境

開発環境で作成した EAR ファイルをリポジトリから読み込み、実行環境に配備する環境です。また、実行環境で利用する HCSC サーバをセットアップします。実行環境での業務の運用を開始したあとは、システムの起動・停止、および状態を監視します。

永続化

ビジネスプロセスのプロセスインスタンスの実行状況を履歴として、データベースに記録することです。

応答電文

サービス部品の実行結果を返すための電文です。応答電文には、サービス部品から HCSC サーバに送信される電文と、HCSC サーバからサービスリクエストに送信される電文があります。

(力行)

開発環境

実行環境でサービス部品を呼び出すために必要な HCSC コンポーネントを作成したり、作成した HCSC コンポーネントを HCSC サーバおよびクラスタにどのように配備するか定義（配備定義）したりする環境です。

カスタムアダプタ

呼び出すサービス部品の種類が、Web サービス、SessionBean、MDB (WS-R)、MDB (DB キュー) 以外のアダプタです。

カスタム受付

サービスリクエストからの実行要求を受け付けるための機能（インターフェース）です。ユーザが作成した受付処理をカスタム受付フレームワーク上で動作させることで、任意のプロトコルからの実行要求を受け付けることができます。

カスタム受付フレームワーク

カスタム受付を作成するために Cosminexus サービスプラットフォームで提供している開発用フレームワークです。

共通フォルダ

FTP 連携で、複数のビジネスプロセスからアクセスできるファイルを格納するために HCSC サーバに作成する任意のフォルダです。

クラスタ

ある共通の機能を提供するサーバの集合です。Cosminexus サービスプラットフォームでは、二つの HCSC サーバで実行系と待機系を構成する HA クラスタと、複数の HCSC サーバでクラスタを構成するロードバランスクラスタがあります。

コネクション

アクティビティを連結して、アクティビティの実行順序を定義するための部品です。通常のコネクションに加えて、フォルト処理を連結するフォルトコネクションおよび並列処理時のリンクを設定するリンクコネクションがあります。

(サ行)

サービス

業務と一対一に対応し、業務に必要な機能を再利用できるソフトウェア部品のことです。SOA では、サービスを組み合わせて業務システムを構築します。これによって、業務の変化に対して、システムを迅速に変更できます。

サービスアダプタ

どのサービス部品を実行するかを定義した HCSC コンポーネントです。呼び出すサービス部品の種類 (Web サービス, SessionBean, MDB (WS-R), MDB (DB キュー)) ごとに定義する内容は異なります。

サービス型ソフトウェア

ソフトウェアの機能のうち、ユーザが必要とするものだけをサービスとして配布し利用できるようにしたソフトウェアの配布形態です。SaaS (Software as a Service) ともいいます。

サービス部品

サービスアダプタおよび DB アダプタから実行できる業務処理のことです。Web サービス, SessionBean, MDB (WS-R), MDB (DB キュー), およびデータベースの操作をサービス部品として利用できます。

サービスリクエスト

HCSC サーバへサービス部品の実行要求 (要求電文) を送信するアプリケーションです。HCSC サーバの標準受付が同期標準受付 (Web サービス / SessionBean) か、非同期標準受付 (MDB (WS-R) / MDB (DB キュー)) か、またはユーザ定義受付 (Web サービス) かによって、作成するサービスリクエストの形式が異なります。

標準受付にサービス部品の実行を要求するサービスリクエストを作成するときは、HCSC サーバに送信される要求電文の電文フォーマットを、標準電文に合わせる必要があります。

ユーザ定義受付にサービス部品の実行を要求するサービスリクエストを作成するときは、サービス部品またはビジネスプロセスに定義した電文フォーマット (WSDL) に合わせる必要があります。

作業フォルダ

FTP 連携で、FTP クライアントや FTP サーバなどから受信したファイルを扱うために HCSC サーバに作成されるフォルダです。

システム構成定義

HCSC サーバおよびクラスタのセットアップ情報と、HCSC コンポーネントを HCSC サーバおよびクラスタにどのように配備するかの情報が含まれる定義情報です。

HCSC サーバおよびクラスタのセットアップ情報は、運用環境で更新されます。開発環境では、HCSC サーバおよびクラスタに HCSC コンポーネントをどのように配備するかの情報を定義します。

実行環境

サービスリクエストで受け付けた実行要求に応じて、HCSC サーバを介して適切なサービス部品を呼び出し、業務を実行する環境です。

実行履歴

ビジネスプロセスのプロセスインスタンスの実行履歴と電文の実行履歴とがあります。ビジネスプロセスの実行状況や応答状況を確認したり、サービス部品の情報や電文の配送経路や配送状態を確認したりできます。

関連セット

サービスリクエストから送信された要求電文を一意に識別するために利用する文字列です。HCSC サーバには、サービスリクエストから関連セットを含む要求電文が送信されます。サービス部品の HCSC サーバは、関連セットを含んだ応答電文をサービスリクエストに返信します。その結果、サービスリクエストは受信した応答電文が、どの要求電文に対する応答であるのかを一意に識別できます。

(夕行)

データ変換

標準電文とサービス部品電文の電文フォーマットが異なる場合に、データ変換定義に従って、標準電文をサービス部品電文の電文フォーマットに変換することです。

データ変換定義

標準電文とサービス部品電文の電文フォーマットの相違を解消するためのデータ変換方法を定義したものです。標準電文とサービス部品電文の電文フォーマット定義ファイルの各要素をマッピングし、必要に応じてファンクションを設定して定義します。

電文フォーマット

要求電文および応答電文のデータ構造です。

サービスリクエストから HCSC サーバに送信される要求電文の電文フォーマットは、標準電文の電文フォーマットに合わせる必要があります。標準電文とサービス部品に実行要求するための電文 (サービス部品電文) の電文フォーマットが異なる場合は、データ変換定義を作成して、電文フォーマットの相違を解消します。

電文フォーマット定義ファイル

電文の構造 (電文フォーマット) を定義したファイルです。XML フォーマット定義ファイルとバイナリフォーマット定義ファイルとがあります。サービス部品の実行に利用する電文で XML 形式のデータを扱う場合は、XML フォーマット定義ファイルを、バイナリ形式 (XML 形式以外) のデータを扱う場合は、バイナリフォーマット定義ファイルを利用します。

(八行)

バイナリフォーマット定義ファイル

バイナリ形式 (XML 形式以外) のデータを扱う場合に利用する電文フォーマット定義ファイルです。XML スキーマファイルとして作成します。バイナリ形式のデータ内の値の格納形式を定義し、

定義した情報を XML スキーマファイルに付加したファイル（拡張子：.fdx）です。

配備

HCSC コンポーネントを適切な HCSC サーバに配置することです。開発環境で作成した HCSC コンポーネントは、運用環境を利用して実行環境（HCSC サーバ）に配備します。

配備定義

開発環境で、HCSC コンポーネントを HCSC サーバおよびクラスタにどのように配備するか定義し、システム構成定義を更新することです。運用環境から HCSC コンポーネントを配備するときには、開発環境で配備定義したシステム構成定義の内容に従って配備します。

パッケージング

開発環境で作成した HCSC コンポーネントを、実行環境に配備できるように EAR ファイルに組み立てることです。

ビジネスプロセス

複数のサービス部品の処理の順番や条件などを定義して、一連の業務の流れとして定義したものです。Cosminexus サービスプラットフォームの開発環境では、複数のアクティビティを画面上に配置し、コネクションで連結して定義します。

標準受付

サービスリクエストからの要求電文を受け付けるための機能（インターフェース）です。HCSC サーバに含まれる機能です。標準受付を開始すると、サービスリクエストからの要求電文を受け付けられる状態になります。

同期受付（Web サービス / SessionBean）および非同期受付（MDB（WS-R） / MDB（DB キュー））の要求を受け付けることができます。

標準電文（標準電文フォーマット）

Cosminexus サービスプラットフォームで提供する標準的な電文フォーマットです。

ファイルアダプタ

Cosminexus サービスプラットフォームの実行環境から HCSC サーバ上のローカルディスクに対して直接ファイルの入出力をする場合に利用するアダプタです。

ファイル操作アダプタ

ファイルのレイアウト変換、文字コード変換、複製、および削除をするためのアダプタです。

ファンクション

データ変換を定義する場合に、変換元ノードの値を加工して変換先ノードにマッピングする場合に使用するときの加工の方法を定義するものです。

文字列や数値を変換するファンクション、ノードの名称を取得するファンクション、条件による処理の選択や繰り返しを定義するファンクションなどがあります。

プロセスインスタンス

ビジネスプロセスで実行している業務の一つの実体です。ワークフローの案件に相当します。

(マ行)

マッピング

データ変換を定義する場合に、変換元と変換先のノードを連結することです。マッピングの始点をマッピング元、終点をマッピング先といいます。

(ヤ行)

ユーザ定義受付

サービスリクエストからの実行要求を受け付けるための機能（インターフェース）です。HCSC サーバに含まれる標準受付とは異なり、ユーザが任意のインターフェースを定義できます。定義したユーザ定義受付を HCSC サーバに配備して開始すると、サービスリクエストからの要求電文を受け付けられる状態になります。

要求電文

サービス部品の実行を要求するための電文です。要求電文には、サービスリクエストから HCSC サーバに送信される電文と HCSC サーバ内のアダプタからサービス部品に送信される電文がありません。

(ラ行)

リポジトリ

定義した HCSC コンポーネントやシステム構成定義を格納するディレクトリです。格納した情報は、リポジトリの管理機能（リポジトリのインポート/エクスポート機能）を利用して、開発環境と運用環境との間で受け渡します。

索引

数字

1:1 系切り替えシステム 27

B

BPEL〔用語解説〕254

BPMN〔用語解説〕254

C

Cookie 情報〔Web サービス (SOAP 通信)〕
63

Cosminexus (EJB コンテナ) との関係 73

Cosminexus RM との関係〔MDB (DB
キュー)〕94

Cosminexus RM との関係〔MDB (WS-R)〕
78

Cosminexus サービスプラットフォームの機
能〔機能概要〕1

csccmd.properties〔用語解説〕255

cscmng.properties〔用語解説〕256

cscsvconfig.properties〔用語解説〕256

cscsvsetup.properties〔用語解説〕256

cscsvsetup.xml〔用語解説〕256

CTM〔HCSC サーバの冗長構成〕25

CTM〔用語解説〕254

D

DB Connector〔用語解説〕254

DB アダプタ〔DB アダプタの作成機能〕4

DB アダプタ〔用語解説〕254

DB アダプタの作成〔DB アダプタの作成機
能〕4

DB キュー〔用語解説〕254

E

EAR ファイル〔用語解説〕254

Eclipse〔用語解説〕254

EJB〔用語解説〕254

F

FTP アダプタ〔用語解説〕255

FTP インバウンドアダプタ〔用語解説〕255

FTP 受付〔用語解説〕255

FTP クライアント〔用語解説〕255

FTP クライアントおよび FTP サーバとの接
続 128

FTP サーバ〔用語解説〕255

FTP 連携〔用語解説〕255

H

HA モニタ〔HCSC サーバの冗長構成〕26

HCSC〔用語解説〕255

HCSC-Business Process〔用語解説〕255

HCSC-Data Transform〔用語解説〕255

HCSC-DB Adapter〔用語解説〕255

HCSC-Manager〔用語解説〕255

HCSC-Manager コマンド共通定義ファイル
〔用語解説〕255

HCSC-Manager 定義ファイル〔用語解説〕
256

HCSC-Messaging〔用語解説〕256

HCSC-TE〔用語解説〕256

HCSC コンポーネント〔用語解説〕256

HCSC コンポーネントの検証 2

HCSC コンポーネントの配備 11

HCSC サーバ〔用語解説〕256

HCSC サーバ構成定義ファイル〔用語解説〕
256

HCSC サーバセットアップ定義ファイル〔用
語解説〕256

HCSC サーバで設定できるタイマ 190

HCSC サーバに関連するタイマの設定 190

HCSC サーバのクラスタ構成 25

HCSC サーバの冗長構成〔クラスタソフト
ウェア〕26

HCSC サーバの冗長構成〔ロードバランス機
能〕25

HCSC サーバのセットアップ〔実行環境の
セットアップ機能〕 19
 HCSC サーバランタイム定義ファイル〔用語
解説〕 256
 HTTP〔用語解説〕 256

I

iWay アダプタ〔用語解説〕 257

J

J2EE サーバ〔用語解説〕 257
 JAX-WS エンジン〔用語解説〕 257
 JDBC〔用語解説〕 257
 JP1〔用語解説〕 257
 JP1/IM との連携による障害検知 238
 JP1 との連携 238
 JP1 によるプロセス監視 239

M

MBean〔用語解説〕 257
 MDB〔用語解説〕 257
 Message Queue アダプタ〔用語解説〕 257
 Microsoft Cluster Service〔HCSC サーバの
冗長構成〕 26
 MyEclipse〔用語解説〕 257

O

Object Access アダプタ〔用語解説〕 257
 Object Wrapper システムとの接続 122
 OpenTP1 との接続 115

R

RMI-IIOP〔用語解説〕 258

S

SessionBean〔用語解説〕 258
 SOAP〔用語解説〕 258
 SOAP1.1/1.2 併用モード〔用語解説〕 258
 SOAP1.1 モード〔用語解説〕 258

SOAP Fault 動作定義ファイルによるフォルト
のキャッチ 201
 SOAP 通信基盤〔用語解説〕 258
 SOAP メッセージの構成〔Web サービス
(SOAP 通信)〕 67
 SQL オペレーション定義ファイル〔用語解
説〕 258

T

TP1/RPC 受付〔用語解説〕 258
 TP1 アダプタ〔用語解説〕 258
 TP1 インバウンドアダプタ〔用語解説〕 258
 TP1 連携システム〔用語解説〕 258

U

uCosminexus Service Architect〔用語解説〕
259
 uCosminexus Service Platform〔用語解説〕
259

W

W3C〔用語解説〕 259
 Web サービス〔用語解説〕 259
 Web サービス (SOAP 通信) の場合の接続
先動的変更 69
 Web サービスの通信基盤との関係 53
 WFA (Work-Flow Architecture)〔用語解
説〕 259
 WS-R〔用語解説〕 259
 WSDL〔用語解説〕 259

X

XML スキーマ〔用語解説〕 259
 XML 電文内の属性値に指定した名前空間接
頭辞の補完 207
 XML 電文のキャッシュ機能 227
 XML フォーマット定義ファイル〔用語解説〕
259

あ

アクティビティ 157

アクティビティ〔ビジネスプロセスの作成機能〕 6
 アクティビティ〔用語解説〕 259
 アダプタ〔用語解説〕 259
 アプリケーションからの稼働状態の確認 204

う

受付処理〔用語解説〕 260
 運用環境〔用語解説〕 260
 運用を管理する機能 21

え

永続化〔用語解説〕 260
 永続化しないビジネスプロセス 161
 永続化するビジネスプロセス 161

お

応答電文〔用語解説〕 260

か

開発環境〔用語解説〕 260
 開発環境の機能概要 2
 カスタムアダプタ〔用語解説〕 260
 カスタムアダプタを使用した場合のシステムとの接続 134
 カスタム受付〔用語解説〕 260
 カスタム受付フレームワーク〔用語解説〕 260
 カスタム受付を使用した場合のシステムとの接続 131

き

キューの名称と個数〔MDB (DB キュー)〕 106
 キューの名称と個数〔MDB (WS-R)〕 93
 共通フォルダ〔用語解説〕 260

く

クラスタ〔用語解説〕 260

クラスタ構成における運用環境の構成パターン 30
 クラスタソフトウェア 26
 クラスタのセットアップ 19

け

検証〔用語解説〕 256

こ

コネクション〔用語解説〕 261
 コンポーネントインポート機能を利用した分散開発の流れ 33

さ

サービス〔用語解説〕 261
 サービスアダプタ〔サービスアダプタの作成機能〕 3
 サービスアダプタ〔用語解説〕 261
 サービスアダプタの作成〔サービスアダプタの作成機能〕 3
 サービス型ソフトウェア〔用語解説〕 261
 サービス部品〔用語解説〕 261
 サービス部品の種類〔サービスアダプタの作成機能〕 3
 サービス部品の接続先変更 200
 サービス部品のタイムアウト値の変更 198
 サービス部品呼び出し 37
 サービス部品呼び出し〔MDB (DB キュー)〕 94
 サービス部品呼び出し〔MDB (WS-R)〕 78
 サービス部品呼び出し〔SessionBean〕 73
 サービス部品呼び出し〔Web サービス (SOAP 通信)〕 53
 サービス部品呼び出し〔基本的な流れ〕 38
 サービス部品呼び出し〔メッセージの流れ (MDB (DB キュー))〕 96
 サービス部品呼び出し〔メッセージの流れ (MDB (WS-R))〕 78
 サービス部品呼び出し〔ユーザ電文と WSDL の関係〕 55
 サービス部品呼び出し〔ユーザ電文とサービス部品のスタブとの関係〕 73

サービス部品を呼び出す〔応答電文の扱い
(MDB (DB キュー))] 103
 サービス部品を呼び出す〔応答電文の扱い
(MDB (WS-R))] 85
 サービス部品を呼び出す流れ 41
 サービスリクエスト〔サービスリクエストの
作成機能〕14
 サービスリクエスト〔用語解説〕261
 サービスリクエストの作成〔サービスリクエ
スタの作成機能〕14
 作業フォルダ〔用語解説〕261

し

識別情報 50
 システム構成定義〔用語解説〕261
 実行環境〔用語解説〕262
 実行環境のセットアップ 19
 実行履歴〔用語解説〕262
 実行履歴の管理 23
 実行履歴の検索〔プロセスインスタンス〕23
 実行履歴の検索と詳細情報の参照〔電文〕24
 実行履歴の削除〔電文〕24
 実行履歴の削除〔プロセスインスタンス〕23
 実行履歴の詳細情報の参照〔プロセスインス
タンス〕23
 障害解析機能 230
 障害回復機能 230
 障害情報の取得機能 230
 障害調査のための機能 229
 障害発生時の系の切り替え 28

せ

性能解析トレースファイル 230
 性能向上のための機能 215
 接続できるデータベース 243

そ

関連セット〔ビジネスプロセスの作成機能〕
6
 関連セット〔用語解説〕262
 関連セットの役割 160

た

タイマの設定 190
 他社 ERP パッケージとの接続 123
 多重度 216
 他製品と連携して利用する機能 237

て

定義例外 201
 データ検証機能 231
 データベースとの接続 108
 データ変換〔用語解説〕262
 データ変換基盤〔用語解説〕255
 データ変換定義〔データ変換定義の作成機
能〕8
 データ変換定義〔用語解説〕262
 データ変換定義の作成〔データ変換定義の作
成機能〕8
 データ変換をしない場合の変数(メッセージ
型)の組み合わせ 210
 デッドメッセージキュー〔MDB (DB
キュー)] 106
 デッドメッセージキュー〔MDB (WS-R)]
93
 電文の実行履歴の管理 24
 電文フォーマット〔電文フォーマットの作成
機能〕2
 電文フォーマット〔用語解説〕262
 電文フォーマット定義ファイル〔用語解説〕
262
 電文フォーマットの作成〔電文フォーマット
の作成機能〕2

と

トランザクション〔MDB (DB キュー)]
105
 トランザクション〔MDB (WS-R)] 87
 トランザクション〔SessionBean] 75
 トランザクション〔Web サービス (SOAP
通信)] 63

に

- 任意形式 (any 形式) の電文フォーマットの定義 208
- 任意形式 (any 形式) を設定した場合の処理の流れ 208

は

- バイナリフォーマット定義ファイル〔用語解説〕 262
- 配備〔用語解説〕 263
- 配備から開始または停止から削除の一括実行機能 13
- 配備定義〔HCSC コンポーネントの配備機能〕 12
- 配備定義〔用語解説〕 263
- 配備定義機能 11
- パッケージング〔パッケージング機能〕 13
- パッケージング〔用語解説〕 263
- パッケージング機能 13
- 汎用ロードバランサ 25

ひ

- ビジネスプロセス〔ビジネスプロセスの作成機能〕 5
- ビジネスプロセス〔用語解説〕 263
- ビジネスプロセス基盤〔用語解説〕 255
- ビジネスプロセスの階層化 7
- ビジネスプロセスの再実行 183
- ビジネスプロセスの再実行〔運用環境画面〕 183
- ビジネスプロセスの再実行〔コマンド〕 183
- ビジネスプロセスの再実行〔サービスリクエスト (SessionBean)〕 184
- ビジネスプロセスの再実行〔サービスリクエスト (SOAP 通信)〕 183
- ビジネスプロセスの作成〔ビジネスプロセスの作成機能〕 5
- ビジネスプロセスの仕組み 152
- ビジネスプロセスの実行状態の管理 7
- ビジネスプロセスの状態の互換性の指定を OFF にした場合のトランザクション 174

- ビジネスプロセスの状態の互換性の指定を ON にした場合のトランザクション 165
- ビジネスプロセスの処理の流れ 152
- ビジネスプロセスの通信モデル 153
- ビジネスプロセスのデバッグ機能 15
- ビジネスプロセスのトランザクション 164
- ビジネスプロセスを使用した待機処理 186
- ビジネスプロセスを使った運用 139
- 標準受付〔用語解説〕 263
- 標準電文〔用語解説〕 263
- 標準電文フォーマット〔用語解説〕 263
- 標準プロトコル以外のシステムとの接続 107

ふ

- ファイルアダプタ〔用語解説〕 263
- ファイル操作アダプタ〔用語解説〕 263
- ファイルを扱うシステムとの接続 125
- ファンクション〔データ変換定義の作成機能〕 9
- ファンクション〔用語解説〕 263
- プロセスインスタンス〔ビジネスプロセスを使った運用〕 154
- プロセスインスタンス〔用語解説〕 263
- プロセスインスタンスの再実行 24
- プロセスインスタンスの実行履歴の管理 23
- 分散開発時の注意事項 36
- 分散開発時のリポジトリのインポート 33

へ

- 変数 6
- 変数の種類 160
- 変数の役割 160

ま

- マッピング〔データ変換定義の作成機能〕 9
- マッピング〔用語解説〕 264

め

- メッセージキューとの接続 127
- メッセージトラッキング機能 230
- メッセージログファイル 230

メッセージング基盤〔用語解説〕 256

ゆ

ユーザ定義受付〔ユーザ定義受付の作成機能〕 10

ユーザ定義受付〔用語解説〕 264

ユーザ定義受付の作成〔ユーザ定義受付の作成機能〕 10

ユーザ電文トレース機能 234

よ

要求電文〔用語解説〕 264

要素 3

り

リクエストトレースファイル 230

リポジトリ〔用語解説〕 264

ろ

ロードバランス機能〔HCSC サーバの冗長構成〕 25