
RT システム仕様記述方式

2008.11.XX
Version 0.2



独立行政法人産業技術総合研究所
知能システム研究部門

Contents

1	概要	6
1.1	目的.....	6
1.2	仕様概要.....	7
1.2.1	要求仕様.....	7
1.2.2	プラットフォーム非依存モデル.....	8
1.2.3	プラットフォーム依存モデル.....	8
1.3	提案に関する連絡先.....	8
1.4	準拠性.....	9
1.5	既存の標準仕様との関連.....	9
1.5.1	Platform-Independent Model.....	9
1.5.2	Platform-Specific Models.....	9
2	Platform-Independent Model	10
2.1	書式および凡例.....	10
2.2	RTS プロファイル.....	12
2.3	RTS 基本プロファイル.....	14
2.3.1	rts_profile.....	14
2.3.2	component.....	17
2.3.3	CompositeType.....	19
2.3.4	dataPort.....	20
2.3.5	serviceport.....	21
2.3.6	configuration_set.....	21
2.3.7	configurationData.....	22
2.3.8	execution_context.....	22
2.3.9	participant.....	23
2.3.10	component_group.....	24
2.3.11	dataport_connector.....	25
2.3.12	serviceport_connector.....	26
2.3.13	target_component.....	27
2.3.14	target_port.....	28
2.3.15	message_Sending.....	28
2.3.16	startup.....	29
2.3.17	shutdown.....	29

2.3.18	activation	29
2.3.19	deactivation	29
2.3.20	resetting	30
2.3.21	initialize	30
2.3.22	finalize	30
2.3.23	condition	31
2.3.24	wait_time	31
2.3.25	preceding	31
2.3.26	target_executioncontext	32
2.4	RTS 拡張プロファイル	34
2.4.1	rts_profile_ext	34
2.4.2	component_ext	35
2.4.3	location	36
2.4.4	Direction	37
2.4.5	dataport_ext	37
2.4.6	serviceport_ext	38
2.4.7	execution_context_ext	39
2.4.8	dataport_connector_ext	39
2.4.9	serviceport_connector_ext	40
2.4.10	target_component_ext	41
2.4.11	target_port_ext	41
2.4.12	condition_ext	41
2.4.13	property	42
3	プラットフォーム依存モデル	43
3.1	UML-XML 変換	43
3.1.1	基本型	43
3.1.2	パッケージと名前空間	44
3.1.3	クラス	44
3.2	UML-YAML 変換	46
3.2.1	基本型	46
3.2.2	パッケージと名前空間	46
3.2.3	クラス	46
4	Appendix: 各種定義	48
4.1	kind (execution_context)	48
5	Appendix: XML スキーマ	49
5.1	RTS 基本 Profile(RtsProfile.xsd)	49
5.2	RTS 拡張 Profile(RtsProfile_ext.xsd)	53
6	Appendix: YAML スキーマ	56
7	Appendix: サンプル	61
7.1	RT システム仕様	61
7.2	XML サンプル	64
7.3	YAML サンプル	66
8	Appendix: 仕様記述利用例	70
8.1	RTS 仕様記述の利用例	70
8.2	システム構築・検証での利用	71
8.3	システム運用時に利用	72
8.4	既存仕様の再利用	73
9	参考文献	74
9.1	標準仕様	74

序論

本仕様書について

本仕様書は次世代ロボット知能化技術開発プロジェクト (以降「知能化プロジェクト」と呼ぶ。) における、各種仕様記述方式のうち、RTシステム仕様記述方式について記述したものである。本仕様記述方式はプロジェクト内で共通的に使用されることを意図したものである。

知能化プロジェクト仕様書

次世代ロボット知能化技術開発プロジェクトに関連する各種仕様は以下のWebサイトにおいて公開する。

<http://openrtp.jp>

凡例

プログラミング言語等の記述は以下の凡例に従って一般文書と区別される。ただし、テーブルおよびセクションのタイトル等はこの限りではない。

Helvetica bold - IDLまたはXMLの要素

Courier bold - プログラミング言語の要素

Helvetica - 例外

イタリックは文書名、仕様名およびその他の文書をさす場合に用いられるものとする。

1 概要

目次

Section Title	Page
目的	6
仕様概要	7
提案に関する連絡先	8
準拠性	9
既存の標準仕様との関連	9

1.1 目的

本仕様書は、コンポーネント指向ロボット開発のための、システム仕様記述方式を定めるものである。コンポーネント指向ロボット開発において、ロボットシステムを構成するコンポーネントの仕様を共通化し、システムインテグレーションの際に開発者に対して十分な情報を提供し、システム構築を円滑に行うためにロボットシステムの情報の標準的な記述方式を定める。

本仕様では、OMG RTC (Robotic Technology Component) Specification に置いて定められているモデルに準拠したコンポーネントモデル (以下RTCとする) を前提とする。RTCはロボットのハードウェアやソフトウェア等の機能要素の論理表現である。UML 2.1.2において記述されている一般的なコンポーネントモデルを拡張し、ロボットドメインに特化したデータ構造や振舞いのデザインパターンを提供することで、RTシステムを構成する柔軟な構成要素 (ビルディングブロック) を提供する。RTシステム開発者は、複数のベンダから提供される多くのRTCを用いて、柔軟なシステムをより迅速に構築することが出来る。

1.2 仕様概要

本仕様は、システム開発者が個々のRTCをブラックボックスとして扱いながら、システム構築を行う際に十分な情報を提供するためのRTシステムの仕様記述方式を定めたものである。

1.2.1 要求仕様

本仕様記述方式により記述された情報は、ロボットシステム開発の以下のフェーズにおいて使用されることを目的とする。

- コンポーネント設計時：ある機能要素をコンポーネント化する際に、そのコンポーネントが受け持つべき機能、提供する機能、入出力データ、インターフェース等、どのような特性を持つかを記述することで、コンポーネント設計時に利用することができる。
- コンポーネント実装時：仕様記述方式に基づきミドルウェアなどが提供するフレームワークを利用して、ある程度のソースコードを生成することが可能である。本仕様記述方式に基づき、コンポーネント仕様を記述したファイルをソースコードジェネレータ等に与えることで、ソースコードの自動生成に利用することができる。
- システム設計時：システムを実コンポーネントを動作させることなく、コンポーネントの仕様のみを使用し、接続の整合性やコンポーネントの配置などを検討する際に利用することができる。
- コンポーネント検索：作成済みのコンポーネントをリポジトリサーバ等に蓄積し、必要な時に適切なコンポーネントを検索、配置するための情報を提供するために使用することなどが想定される。
- 仕様の再利用：機能やインターフェースが同等であるが、実装が異なるコンポーネントを作成する際に、仕様を再利用することができる。

1.2.2 プラットフォーム非依存モデル

プラットフォーム非依存モデル (Platform Independent Model : 以後PIMとする) は特定の言語や記述フォーマットに依存しないソフトウェアまたはデータのモデルであり、その実体はUMLで記述されたモデル図と詳細を記述した文書から構成される。

PIM は以下の2つのパッケージから構成される:

1. **RTS Basic Profile.** RTシステムの基本的なモデルを記述する際に必要な情報を定義する。RTシステムの基本構成、Port間接続情報などがこれに含まれる。
2. **RTS Extended Profile.** RTS Basic Profile に含まれないオプションな雑多な情報や各種ツールでの利用を想定した情報を記述するためのデータ構造を定義する。

以下に、上記2つのパッケージの関係を示す。

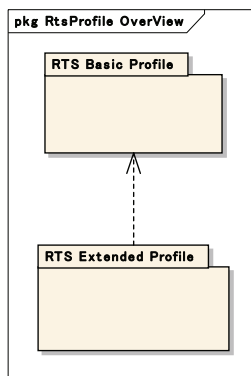


Figure 1 RTC Description Format

1.2.3 プラットフォーム依存モデル

プラットフォーム依存モデル (Platform Specific Model: 以後PSMとする) は、PIMで定義したモデルを利用するプラットフォームに応じて変換したモデルである。PSMは、ツール間のデータ交換や実装する際のデータ構造として利用される。

本仕様書では以下のPSMを提示している。

XML. RTS仕様記述方式をXMLで記述するためのXMLスキーマ。

YAML. RTS仕様記述方式をYAMLで記述するためのYAMLスキーマ。

1.3 提案に関する連絡先

本仕様書に関する質問・意見等の宛先は以下のとおり。

安藤 慶昭
独立行政法人産業技術総合研究所 研究員
知能システム研究部門 タスクインテリジェンス研究グループ
〒305-8568 茨城県つくば市梅園1-1-1 中央第2
TEL: 029-861-5981 FAX: 029-861-5971
e-mail: n-ando@aist.go.jp

1.4 準拠性

本仕様に記述される全てのPIMに準拠したデータ構造を実装することが実装者には求められる。ただし、オプションな属性に関してはこの限りでない。

1.5 既存の標準仕様との関連

1.5.1 Platform-Independent Model

PIM は UML 2.1.2 [UML] を用いて定義されている。

コンポーネントモデルは OMG RTC Specification [RTC] および OMG SDO (Super Distributed Object) [SDO] において定義されている。

1.5.2 Platform-Specific Models

PSMはXMLスキーマ[XML Schema]およびYAMLスキーマを用いて定義されている。

2 Platform-Independent Model

目次

Section Title	Page
書式および凡例	10
RTSプロファイル	12
RTS基本プロファイル	14
RTS拡張プロファイル	34

2.1 書式および凡例

PIMにおいて、クラスは以下に示すような表を用いて記述される。各メンバの詳細は表に続く本文で記述される。

<class name>		
attributes		
<attribute name>	<multiplicity>	<attribute type>
...
relationships		
<association end name>	<multiplicity>	<property type>
...

上記の表において、attributesの欄は対象クラスが直接所有する属性を表す。また、relationshipsの欄は関連により対象クラスが間接的に所有する属性を表す。また、列挙型定数は列挙型として定義される。

以下に、属性として **long** 型の **my_attribute**(多重度0..1)、関連として **SomeClass**型の **my_relational_attribute1**(多重度0..*)、**string**

型の **my_relational_attribute2**(多重度1) を持つ**MyClass** の例を示す。

MyClass		
attributes		
my_attribute	0..1	long
relationships		
my_relational_attribute1	0..*	SomeClass
my_relational_attribute2	1	string

2.2 RTS プロファイル

RTSプロファイルは、RTシステムの各種属性、接続情報を定義するデータモデルである。RTSプロファイルのPIMモデル全体図を図2に示す。また、プロパティ情報に関する部分のモデルを図3に示す。

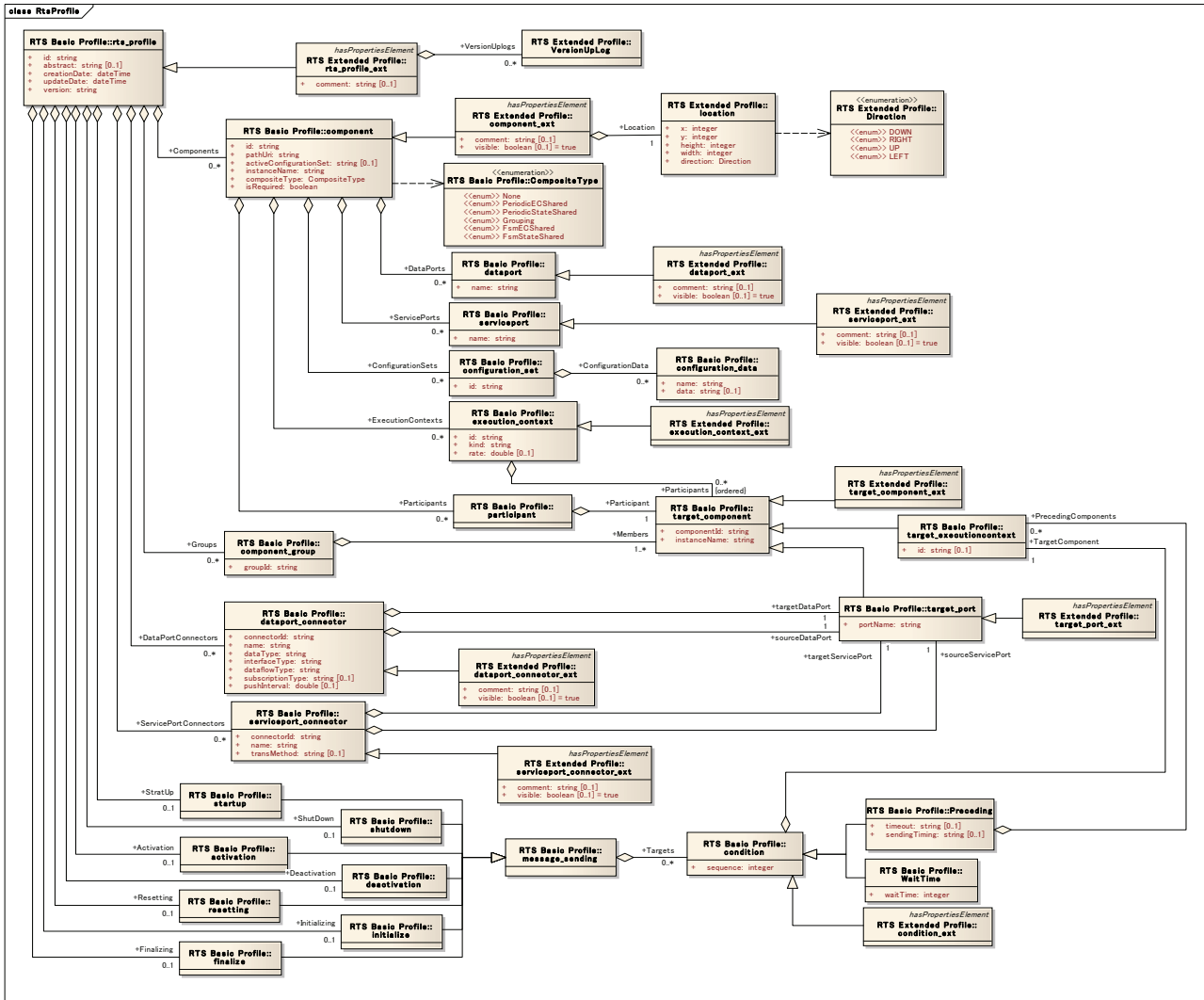


Figure 2 RTS プロファイル(全体図)

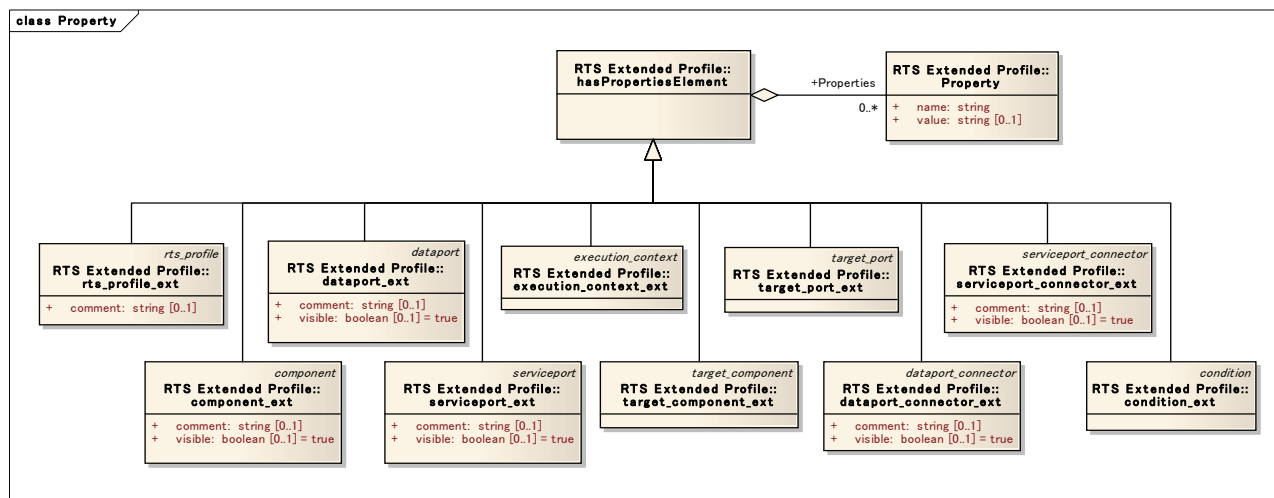


Figure3 RTS プロファイル(制約部分)

RTS基本プロファイル (RTS Basic Profile)

RTS基本プロファイルは、RTシステムの基本メタ情報を含む。

RTS拡張プロファイル (RTS Extended Profile)

RTS拡張プロファイルは、RTシステムの本質的な機能にかかわらない付加的な情報や各種ツールでの利用を想定した情報を記述するために用意されているプロファイルである。

2.3 RTS 基本プロファイル

2.3.1 rts_profile

概要

rts_profileは、RTシステム仕様記述のルート要素である。また、仕様記述対象のRTシステムを識別するID情報やRTC仕様記述のバージョン情報を保持する。

RTSプロファイル(PIM)の基本部分の拡大図を図4に示す。

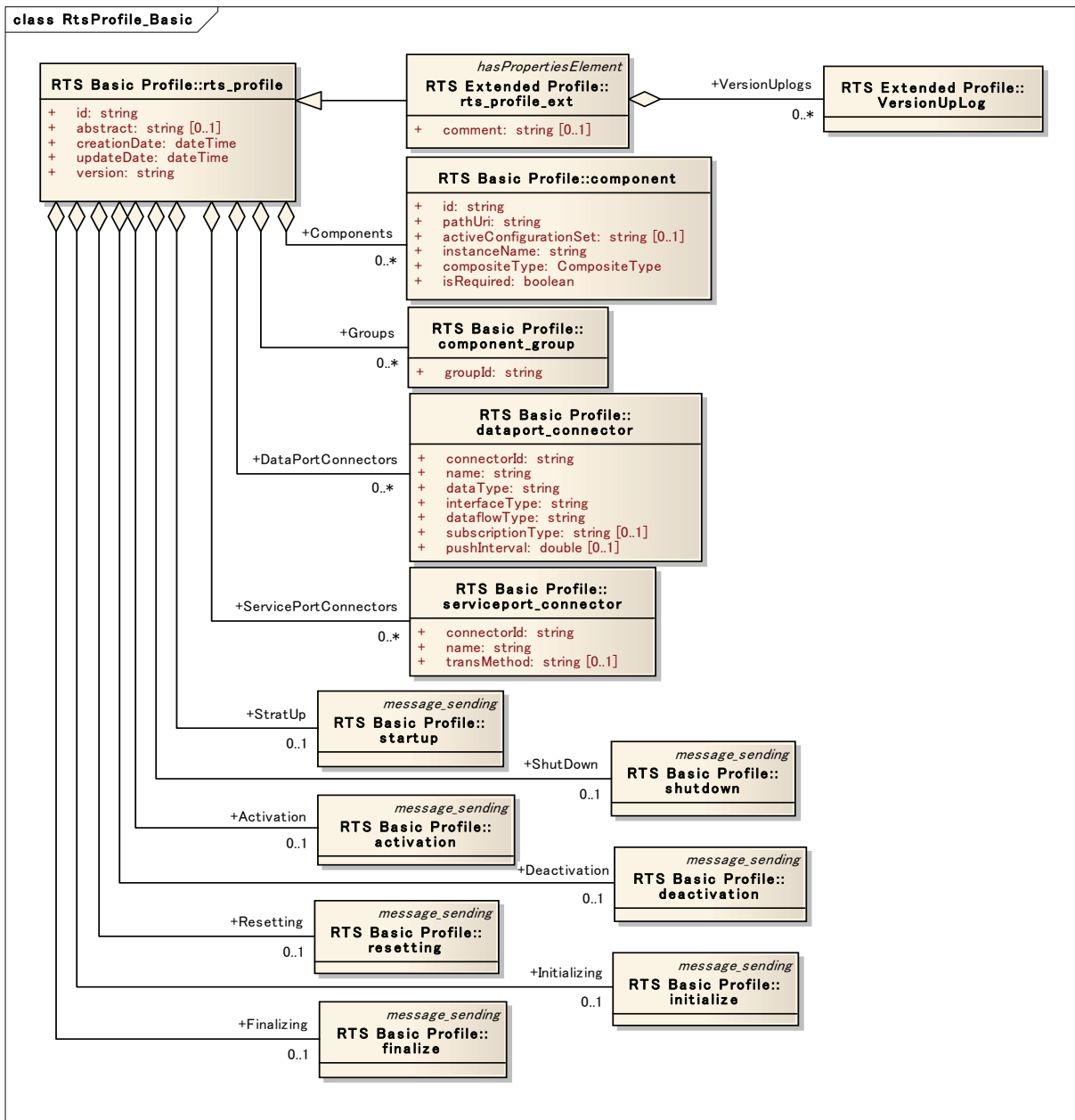


Figure 4 RTS プロファイル(基本部分)

意味

RTシステム仕様記述の各種詳細情報は、`rts_profile`要素以下の各要素に記述される。

属性

<i>rts_profile</i>		
attributes		
id	1	string
abstract	0..1	string
creationDate	1	dateTime
updateDate	1	dateTime
version	1	string
relationships		
Components	0..*	component
Groups	0..*	component_group
DataPortConnectors	0..*	dataport_connector
ServicePortConnectors	0..*	Serviceport_connector
StartUp	0..1	startup
ShutDown	0..1	shutdown
Activation	0..1	activation
Deactivation	0..1	deactivation
Resetting	0..1	resetting
Initializing	0..1	initialize
Finalizing	0..1	finalize

2.3.1.1 *id*

概要

仕様記述対象のRTシステムを一意に識別するための識別子を指定する。

意味

RTシステムの識別子を指定する。

識別子の書式は、以下の構成とする。

RTSystem : [ベンダ名] : [システム名] : [バージョン番号]

2.3.1.2 *abstract*

概要

RTシステムに関する説明を指定する。

意味

この属性値はRTシステムの利用者の利便性のために設けられている。省略可能属性。

2.3.1.3 *creationDate*

概要

RTシステム仕様の作成日時を指定する。

意味

この属性値は当該RTシステム仕様が最初に作成された日時を記述する。基本的にはツールにて自動設定されることを意図している。必須属性。

2.3.1.4 *updateDate*

概要

RTシステム仕様の最終更新日時を指定する。

意味

この属性値は当該RTシステム仕様が最後に更新された日時を記述する。基本的にはツールにて自動設定されることを意図している。必須属性。

2.3.1.5 *version*

概要

RTシステム仕様記述自体のバージョン番号を指定する。

意味

利用しているRTシステム仕様記述の整合性を確保するためにバージョン番号を指定する。

2.3.2 component

概要

componentはRTシステムを構成するRTCの情報を記述する要素メタ情報である。

RTSプロファイル(PIM)のコンポーネント部分の拡大図を図5に示す。

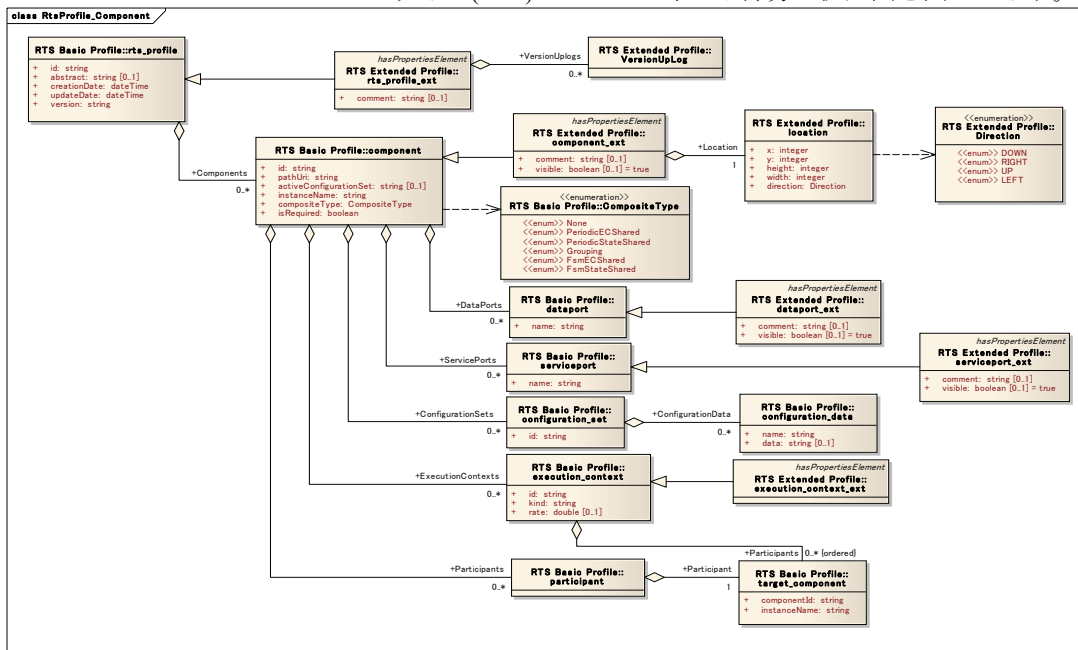


Figure 5 RTS プロファイル(コンポーネント部分)

意味

componentはRTシステムを構成するRTCに関する情報を記述する。

属性

<i>component</i>			
attributes			
id	1		string
pathUri	1		string
activeConfigurationSet	0..1		string
instanceName	1		string
compositeType	1		CompositeType
isRequired	1		boolean
relationships			
DataPorts	0..*		dataport
ServicePorts	0..*		serviceport
ConfigurationSets	0..*		configuration_set
ExecutionContexts	0..*		execution_context
Participants	0..*		participant

2.3.2.1 *id*

概要

RTシステムを構成するRTCのidを指定する。

意味

RTシステムを構成するRTCのidを指定する。同じ仕様のRTCから複数のインスタンスを生成する場合もあるため、それぞれのRTCは、id + instanceNameにて一意に識別される。必須属性。

2.3.2.2 *pathUri*

概要

RTシステムを構成するRTC仕様の登録場所をURI形式で指定する。

意味

RTC仕様の登録場所をURI形式で指定する。必須属性。

2.3.2.3 *activeConfigurationSet*

概要

RTCのactiveConfigurationSetのIDを指定する。

意味

当該RTCにactiveConfigurationSetが設定されている場合に、そのIDを指定する。省略可能属性。

2.3.2.4 *instanceName*

概要

RTシステムを構成するRTCのインスタンス名を指定する。

意味

RTシステムを構成するRTCのインスタンス名を指定する。同じ仕様のRTCから複数のインスタンスを生成する場合もあるため、それぞれのRTCは、id + instanceNameにて一意に識別される。必須属性。

2.3.2.5 *compositeType*

概要

コンポジットコンポーネントの形式を指定する。

意味

RTシステムを構成するRTCのコンポジットタイプを指定する。コンポジットタイプは、CompositeType列挙型で定義されている中から選択可能である。必須属性。

2.3.2.6 *isRequired*

概要

当該RTCが必須であるかを指定する。

意味

RTシステムを構成するRTCが必須であるかを指定する。定義したRTシス

テムを動作させる際に、本RTCが必ず必要であるかを指定する。必須属性。

2.3.3 CompositeType

CompositeTypeは、RTCのコンポジットタイプを指定するための列挙型である。

属性

CompositeType	
attributes	
None	CompositeType
PeriodicECShared	CompositeType
PeriodicStateShared	CompositeType
Grouping	CompositeType
FsmECShared	CompositeType
FsmStateShared	CompositeType

2.3.3.1 None

概要

コンポジットコンポーネントではないRTCを表す型。

意味

対象RTCがコンポジットコンポーネントではなく、通常のRTCであることを表現する型。

2.3.3.2 PeriodicECShared

概要

周期実行型のExecutionContextを持ち、全ての属性を共有する形のコンポジットコンポーネントを表す型。

意味

対象RTCが全ての属性を共有する形のコンポジットコンポーネントであることを表現する型。本タイプのコンポジットコンポーネントでは、各RTCが同一の周期実行型ExecutionContextを共有するとともに、状態も共有する形で動作する。RTC間の結合度が一番強いタイプのコンポジットコンポーネント。

2.3.3.3 PeriodicStateShared

概要

周期実行型のExecutionContextのみを共有する形のコンポジットコンポーネントを表す型。

意味

対象RTCが周期実行型のExecutionContextのみを共有する形のコンポジットコンポーネントであることを表現する型。本タイプのコンポジットコンポーネントでは、各RTCがExecutionContextのみを共有する形で動作する。

各RTCの状態は独立であるため、あるコンポジットコンポーネント内で複数の状態が存在する場合もありえる。

2.3.3.4 Grouping

概要

何も共有しない形のコンポジットコンポーネントを表す型。

意味

対象RTCが何も共有しない形のコンポジットコンポーネントであることを表現する型。本タイプのコンポジットコンポーネントでは、各RTCがそれぞれExecutionContextや状態を保持する。RTC間の結合度が一番弱いタイプのコンポジットコンポーネント。

2.3.3.5 FsmECShared

概要

イベント駆動型のExecutionContextを持ち、全ての属性を共有する形のコンポジットコンポーネントを表す型。

意味

対象RTCが全ての属性を共有する形のコンポジットコンポーネントであることを表現する型。本タイプのコンポジットコンポーネントでは、各RTCが同一のイベント駆動型ExecutionContextを共有するとともに、状態も共有する形で動作する。RTC間の結合度が一番強いタイプのコンポジットコンポーネント。

2.3.3.6 FsmStateShared

概要

イベント駆動型のExecutionContextのみを共有する形のコンポジットコンポーネントを表す型。

意味

対象RTCがイベント駆動型のExecutionContextのみを共有する形のコンポジットコンポーネントであることを表現する型。本タイプのコンポジットコンポーネントでは、各RTCがExecutionContextのみを共有する形で動作する。各RTCの状態は独立であるため、あるコンポジットコンポーネント内で複数の状態が存在する場合もありえる。

2.3.4 dataPort

RTシステムを構成する各RTCについて、他のデータポートとの接続情報(ConnectorProfile)を保持するデータポートを表す。

属性

<i>dataPort</i>		
attributes		
name	1	string
no relationships		

2.3.4.1 name

概要

DataPortの名称を指定する。

意味

接続情報(ConnectorProfile)を保持するDataPortの名称を指定する。必須属性。

2.3.5 serviceport

RTシステムを構成する各RTCについて、他のサービスポートとの接続情報(ConnectorProfile)を保持するサービスポートを表す。

属性

<i>serviceport</i>		
attributes		
name	1	string
no relationships		

2.3.5.1 name

概要

ServicePortの名称を指定する。

意味

接続情報(ConnectorProfile)を保持するServicePortの名称を指定する。必須属性。

2.3.6 configuration_set

概要

configuration_setは、各RTCに設定されたコンフィギュレーションパラメータに関するメタ情報の集合である。複数のコンフィギュレーションパラメータをまとめたものをConfigurationSetと呼び、RTCはConfigurationSetを複数持つことができる。ConfigurationSetのパラメータはRTCのConfigurationインターフェースから取得・変更することができる。

属性

<i>configuration_set</i>		
attributes		
id	1	string
relationships		
ConfigurationData	0..*	configurationData

2.3.6.1 id

概要

各ConfigurationSetを識別するid情報を指定する。

意味

本属性は各RTCに設定されたConfigurationSetを識別するためのid情報を指定する。必須属性。

2.3.7 configurationData

configurationDataは各RTCに設定された個々のコンフィギュレーションパラメータに関するメタ情報を記述する。

属性

configurationData		
attributes		
name	1	string
data	0..1	string
no relationships		

2.3.7.1 name

概要

コンフィギュレーションパラメータの名称を記述する。

意味

各RTCに設定したコンフィギュレーションパラメータの名称を記述する。この名称はRTCのconfigurationパラメータのキー値として使用される。必須属性。

2.3.7.2 data

概要

コンフィギュレーションパラメータの設定値を記述する。

意味

各RTCに設定したコンフィギュレーションパラメータの設定値を記述する。省略可能属性。

2.3.8 execution_context

execution_contextは、当該RTシステムにて使用するExecutionContextに関するメタ情報を定義する。

属性

execution_context		
attributes		
id	1	string
kind	1	string
rate	0..1	double
relationships		

Participants	0..* {ordered}	target_component
--------------	----------------	------------------

2.3.8.1 *id*

概要

ExecutionContextのidを指定する。

意味

ExecutionContextを識別するためのid情報を設定する。必須属性。

2.3.8.2 *kind*

概要

使用しているExecutionContextのアクションの実行タイプを指定する。

意味

アクションの実行タイプとしては、Appendixに指定されている種類の中から選択可能である。必須属性。

2.3.8.3 *rate*

概要

使用しているExecutionContextの実行周期を指定する。

意味

本属性は使用しているExecutionContextの実行周期を記述する。記述単位は[Hz]である。この属性値はPeriodic Sampled Data Processing型のExecutionContext(データフロー型のRTC)のみにおいて有効であり、他のタイプの場合には無視される。省略可能属性。

2.3.9 *participant*

participantはココンジットコンポーネントを構成するコンポーネントを表す。実際の参照RTCは、子要素のtarget_component要素にて指定する。

属性

<i>Participant</i>		
no attributes		
relationships		
Participant	0..*	target_component

2.3.10 component_group

当該RTシステム内のコンポーネントのグループ情報に関するメタ情報を定義する。

RTSプロファイル(PIM)のグループ情報部分および接続情報部分の拡大図を図6に示す。

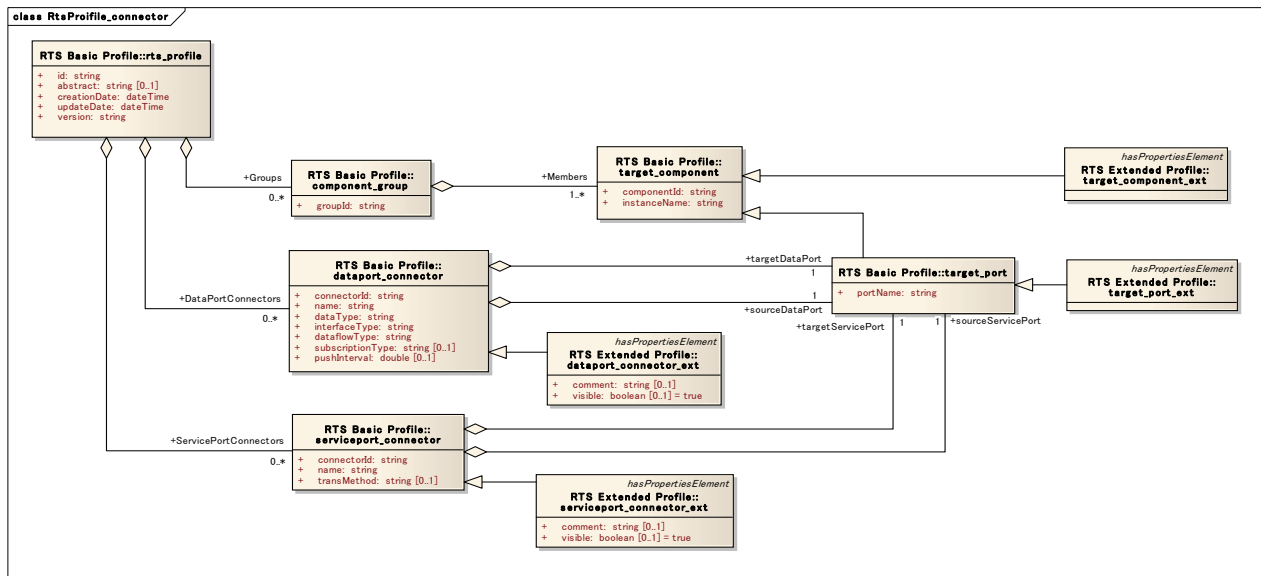


Figure 6 RTS プロファイル(グループ情報部分および接続情報部分)

属性

<i>component_group</i>		
attributes		
groupId	1	string
relationships		
Members	1..*	target_component

2.3.10.1 groupId

概要

RTシステム内のコンポーネントグループを一意に識別する情報を指定する。

意味

当該RTシステム内のコンポーネントグループを一意に識別するID情報を指定する。必須属性。

2.3.11 *dataport_connector*

DataPort間の接続情報に関するメタ情報を定義する。

属性

<i>Dataport_connector</i>		
attributes		
connectorId	1	string
name	1	string
dataType	1	string
interfaceType	1	string
dataflowType	1	string
subscriptionType	0..1	string
pushInterval	0..1	double
relationships		
sourceDataPort	1	target_port
targetDataPort	1	target_port

2.3.11.1 *connectorId*

概要

接続情報を一意に識別する情報を指定する。

意味

接続情報(ConnectorProfile)を一意に識別するID情報を指定する。必須属性。

2.3.11.2 *name*

概要

接続情報に関する名称を指定する。

意味

接続情報(ConnectorProfile)の名称を指定する。必須属性。

2.3.11.3 *dataType*

概要

dataPort間でやり取りするデータの型を指定する。

意味

当該Connectorにて接続されているdataPort間で送受信するデータ型を指定する。必須属性。

2.3.11.4 *interfaceType*

概要

当該Connectorにて使用するinterfaceTypeを指定する。

意味

RTシステム設計者が設計時に想定しているinterfaceTypeを指定する。実際の動作時には、各RTCが動作するRTミドルウェアでサポートしている型に

も依存する。必須属性。

2.3.11.5 *dataflowType*

概要

当該Connectorにて使用するdataflowTypeを指定する。

意味

RTシステム設計者が設計時に想定しているdataflowTypeを指定する。実際の動作時には、各RTCが動作するRTミドルウェアでサポートしている型にも依存する。必須属性。

2.3.11.6 *subscriptionType*

概要

当該Connectorにて使用するsubscriptionTypeを指定する。

意味

RTシステム設計者が設計時に想定しているsubscriptionTypeを指定する。本属性はDataFlowTypeとしてPUSHが設定されている場合のみ有効となる。また、実際の動作時には、各RTCが動作するRTミドルウェアでサポートしている型にも依存する。省略可能属性。

2.3.11.7 *pushInterval*

概要

当該Connectorにて使用するデータ送信周期を指定する。

意味

本属性は、RTシステム設計者が設計時に想定しているデータ送信周期を指定する。省略可能属性。

2.3.12 *serviceport_connector*

ServicePort間の接続情報に関するメタ情報を定義する。

属性

<i>serviceport_connector</i>		
attributes		
connectorId	1	string
name	1	string
transMethod	0..1	string
relationships		
sourceServicePort	1	target_port
targetServicePort	1	target_port

2.3.12.1 *connectorId*

概要

接続情報を一意に識別する情報を指定する。

意味

サービスポート間の接続情報(ConnectorProfile)を一意に識別するID情報を指定する。必須属性。

2.3.12.2 name

概要

接続情報の名称を指定する。

意味

接続情報(ConnectorProfile)の名称を指定する。必須属性。

2.3.12.3 transMethod

概要

ServicePortが利用する通信方式を指定する。

意味

RTシステム設計者が設計時に想定している通信方式を指定する。実際の動作時には、各RTCが動作するRTミドルウェアでサポートしている方式にも依存する。省略可能属性。

2.3.13 target_component

target_componentは、参照先RTCを識別するための情報を記述する。本要素は、コンポーネントグループ指定、ExecutionContext参加要素指定、起動順などの操作起動順指定、コンポジットコンポーネントの子要素指定時などに要素を特定するために使用される。

属性

target_component			
attributes			
componentId	1	string	
instanceName	1	string	
no relationships			

2.3.13.1 componentId

概要

対象RTCのidを指定する。

意味

参照先RTCのidを指定する。各RTCは、componentId + instanceNameによって一意に識別される。必須属性。

2.3.13.2 instanceName

概要

対象RTCのインスタンス名を指定する。

意味

参照先RTCのインスタンス名を指定する。各RTCは、componentId + instanceNameによって一意に識別される。必須属性。

2.3.14 target_port

target_portは、各コネクタの接続先ポートの情報を記述する。dataport_connector, serviceport_connectorの接続先要素を特定するために使用される。

属性など

target_port		
Derived from	target_component	
attributes		
portName	1	string
no relationships		

2.3.14.1 portName

概要

対象Portの名称を指定する。

意味

接続対象Portの名前を指定する。必須属性。

2.3.15 message_Sending

概要

各RTCの起動順、終了順などシステム内での振る舞いを指定する要素。RTSプロファイル(PIM)の振る舞い情報部分の拡大図を図7に示す。

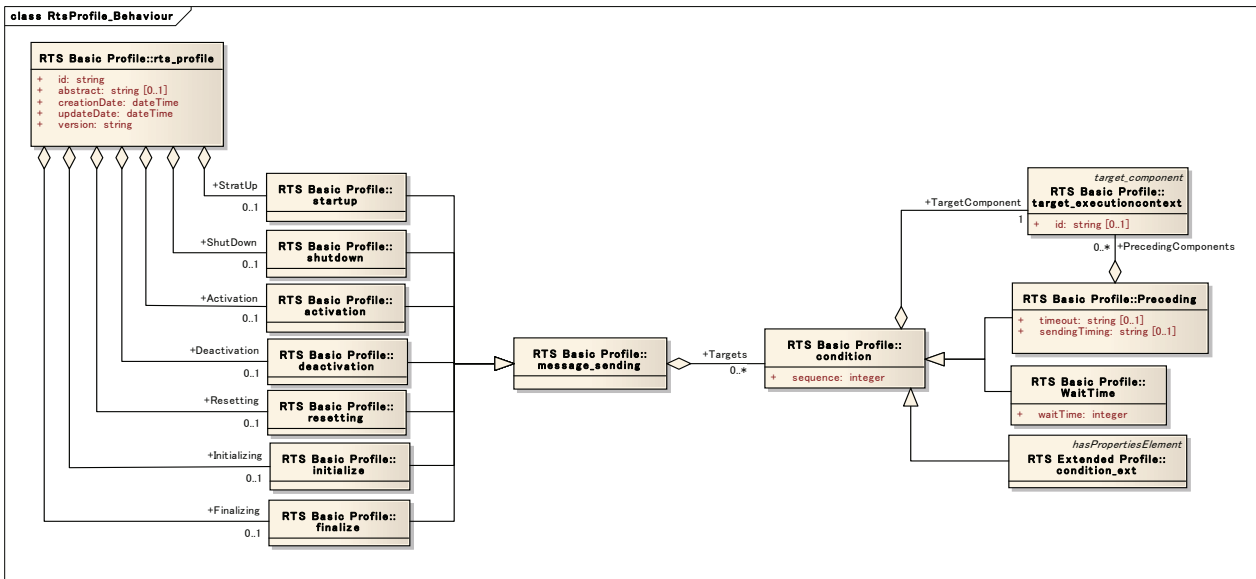


Figure7 RTS プロファイル(振る舞い情報部分)

属性など

message_sending		
no attributes		
relationships		
Targets	0..*	condition

2.3.16 *startup*

概要

RTシステムを起動する際の各RTCの起動順、起動条件を指定するための要素である。

属性など

startup	
Derived from	message_sending
no attributes	
no relationships	

2.3.17 *shutdown*

概要

RTシステムを停止する際の各RTCの停止順、停止条件を指定するための要素である。

属性など

shutdown	
Derived from	message_sending
no attributes	
no relationships	

2.3.18 *activation*

概要

RTシステムを活性化(Active化)する際の各RTCの活性化順、活性化条件を指定するための要素である。

属性など

activation	
Derived from	message_sending
no attributes	
no relationships	

2.3.19 *deactivation*

概要

RTシステムを非活性化(Deactive化)する際の各RTCの非活性化順、非活性

化条件を指定するための要素である。

属性など

<i>deactivation</i>	
Derived from	message_sending
no attributes	
no relationships	

2.3.20 *resetting*

概要

RTシステムをリセットする際の各RTCのリセット順、リセット処理実行条件を指定するための要素である。

属性など

<i>resetting</i>	
Derived from	message_sending
no attributes	
no relationships	

2.3.21 *initialize*

概要

RTシステムを初期化する際の各RTCの初期化順、初期化処理実行条件を指定するための要素である。

属性など

<i>initialize</i>	
Derived from	message_sending
no attributes	
no relationships	

2.3.22 *finalize*

概要

RTシステムを終了する際の各RTCの終了順、終了処理実行条件を指定するための要素である。

属性など

<i>finalize</i>	
Derived from	message_sending
no attributes	
no relationships	

2.3.23 *condition*

概要

*condition*は、RTCの各種操作の実行順、実行条件を指定する要素である。なお、実行条件については、実行までの待機時間、先行して実行すべきRTCを指定可能である。また、実行順、実行条件の指定対象RTCは、関連属性であるTargetComponentにて指定する。

属性

<i>conditions</i>		
attributes		
sequence	1	integer
relationships		
TargetComponent	1	target_executioncontext

2.3.23.1 *sequence*

概要

各種操作の実行順を指定する。

意味

TargetComponentで指定したRTCの実行順を指定する。必須属性。

2.3.24 *wait_time*

概要

*wait_time*は、各RTCが指定された処理を実行するまでの待機時間を記述する。各RTCは指定された処理の実行指令を受け取った後、本要素で指定された時間だけ待った後に実際の処理を開始する。待機時間はミリ秒単位で指定。

属性

<i>wait_time</i>		
attributes		
waitTime	1	integer
no relationships		

2.3.24.1 *waitTime*

概要

各種操作の待機時間を指定する。

意味

指定された各種操作を実行するまでの待機時間をミリ秒単位で指定する。必須属性。

2.3.25 *preceding*

概要

*preceding*は、対象RTCが指定された処理を実行する際に、同じ処理を先行

して実行すべきRTCを指定する。対象RTCが実行指令を受け取った場合、本要素で指定されているRTCの状態を確認し、全要素が処理実行済みの場合に実際の処理を実行する。

属性

<i>preceding</i>		
attributes		
timeout	0..1	string
sendingTiming	0..1	string
relationships		
PrecedingComponent	0..*	target_executioncontext

2.3.25.1 timeout

概要

先行RTCのタイムアウト時間を指定する。

意味

先行RTCとして指定された要素が正常に動作しているかどうかを判定するためのタイムアウト時間をミリ秒単位で指定する。省略可能属性。

2.3.25.2 sendingTiming

概要

各操作の実行タイミングを指定する。

意味

指定された各操作を実行するタイミングを指定する。先行RTCの処理が完了後実行(同期処理)、先行RTCの処理完了を待たずに実行(非同期処理)を指定可能。指定が省略されている場合には同期実行とみなす。省略可能属性。

2.3.26 target_executioncontext

target_executioncontextは、実行順、実行条件の指定対象RTCおよび先行RTCを指定する際に利用する要素である。1つのRTCは複数のExecutionContextを持つことが可能であるため、設定対象操作がstartup、shutdown、activation、deactivation、resettingの場合には、RTCに加えてExecutionContextを指定する必要がある。

属性など

<i>target_executioncontext</i>		
Derived from	target_component	
attributes		
id	0..1	string
no relationships		

2.3.26.1 *id*

概要

対象ExecutionContextのIDを指定する。

意味

指定対象ExecutionContextのIDを指定する。設定対象操作がstartup、shutdown、activation、deactivation、resettingの場合には指定が必要。省略可能属性。

2.4 RTS 拡張プロファイル

2.4.1 rts_profile_ext

rts_profile_extはRTシステムの付加情報に関するメタ情報を記述する。RTSプロファイル(PIM)の基本部分の拡大図を図8に再掲する。

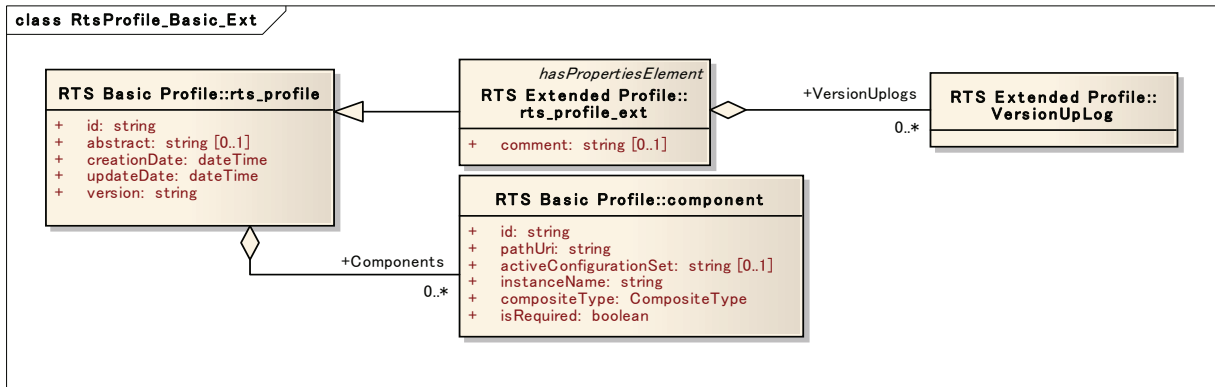


Figure 8 RTS プロファイル(基本部分)

属性

<i>rts_profile_ext</i>		
attributes		
comment	0..1	string
VersionUpLog	0..*	string
relationships		
Properties	0..*	property

2.4.1.1 comment

概要

RTシステムに対する各種コメントを記述する。

意味

RTシステムに対する簡単なコメントを記述する。本属性は各種ツールで表示することを意図している(本属性の表示/非表示は各種ツール毎の設定に依存する)。省略可能属性。

2.4.1.2 VersionUpLog

概要

RTシステムの各バージョンに関する補足説明を記述する。

意味

RTシステムのバージョンアップを行う際に、各バージョンに関する説明(修正点、機能アップ点など)を記述する。省略可能属性。

2.4.2 component_ext

component_extはcomponentの付加情報に関するメタ情報を記述するための親要素である。

RTSプロファイル(PIM)のコンポーネント部分の拡大図を図9に再掲する。

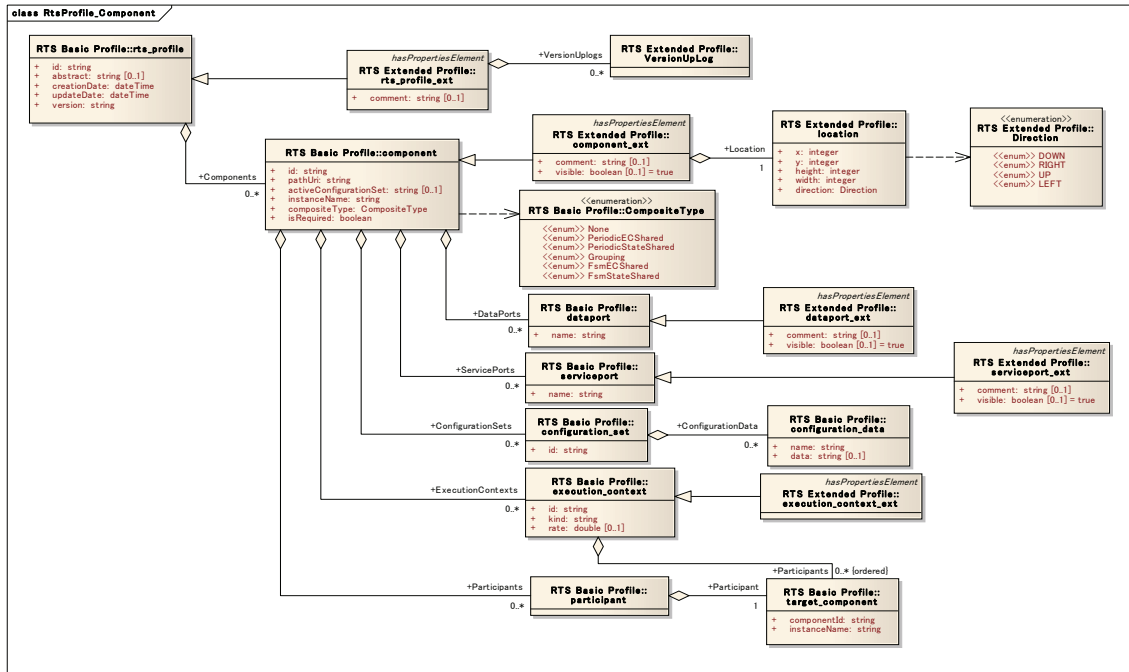


Figure 9 RTS プロファイル(コンポーネント部分)

属性

component_ext		
attributes		
comment	0..1	string
visible	0..1	boolean
relationships		
Location	1	Location
Properties	0..*	property

2.4.2.1 comment

概要

RTCに対する各種コメントを記述する。

意味

RTコンポーネントに対する簡単なコメントを記述する。本属性は各種ツールで表示することを意図している(本属性の表示/非表示は各種ツール毎の設定に依存する)。省略可能属性。

2.4.2.2 *visible*

概要

RTCの表示/非表示を設定する。

意味

当該RTコンポーネントをツール上で表示するかしないかを設定する。本属性が省略されている場合にはTrue(表示)が設定されているとみなす。省略可能属性。

2.4.3 *location*

*location*は当該componentのツール上での表示情報に関するメタ情報を記述する。

属性

<i>location</i>		
attributes		
x	1	integer
y	1	integer
height	1	integer
width	1	integer
direction	1	Direction
no relationships		

2.4.3.1 *x*

概要

当該コンポーネントのツール上での表示位置 x 座標を指定する。

意味

ツール上でコンポーネントを表示する際のx座標を指定する。必須属性。

2.4.3.2 *y*

概要

当該コンポーネントのツール上での表示位置y座標を指定する。

意味

ツール上でコンポーネントを表示する際のy座標を指定する。必須属性。

2.4.3.3 *height*

概要

当該コンポーネントのツール上での表示高さを指定する。

意味

ツール上でコンポーネントを表示する際の高さを指定する。本属性に「-1」が設定されている場合は、ツール側にて表示高さの自動調整を行う。必須属性。

2.4.3.4 width

概要

当該コンポーネントのツール上での表示幅を指定する。

意味

ツール上でコンポーネントを表示する際の幅を指定する。本属性に「-1」が設定されている場合は、ツール側にて表示幅の自動調整を行う。必須属性。

2.4.3.5 direction

概要

当該コンポーネントのツール上での表示向きを指定する。

意味

ツール上でコンポーネントを表示する際の向きを指定する。必須属性。

2.4.4 Direction

Directionは、ツール上にてRTCを表示する向きを指定するための列挙型である。各項目はそれぞれ「下向き」「右向き」「上向き」「左向き」にてRTCを表示することを表す

属性

<i>Direction</i>	
attributes	
DOWN	Direction
RIGHT	Direction
UP	Direction
LEFT	Direction

2.4.5 dataport_ext

dataport_extはdataPortの付加情報に関するメタ情報を記述するための親要素である。

属性

<i>dataport_ext</i>		
attributes		
comment	0..1	string
visible	0..1	boolean
relationships		
Properties	0..*	property

2.4.5.1 comment

概要

dataPortに対する各種コメントを記述する。

意味

dataPortに対する簡単なコメントを記述する。本属性は各種ツールで表示することを意図している(本属性の表示/非表示は各種ツール毎の設定に依存する)。省略可能属性。

2.4.5.2 *visible*

概要

dataPortの表示/非表示を設定する。

意味

当該dataPortをツール上で表示するかしないかを設定する。本属性が省略されている場合にはTrue(表示)が設定されているとみなす。省略可能属性。

2.4.6 *serviceport_ext*

serviceport_extはservicePortの付加情報に関するメタ情報を記述するための親要素である。

属性

<i>serviceport_ext</i>		
attributes		
comment	0..1	string
visible	0..1	boolean
relationships		
Properties	0..*	property

2.4.6.1 *comment*

概要

servicePortに対する各種コメントを記述する。

意味

servicePortに対する簡単なコメントを記述する。本属性は各種ツールで表示することを意図している(本属性の表示/非表示は各種ツール毎の設定に依存する)。省略可能属性。

2.4.6.2 *visible*

概要

servicePortの表示/非表示を設定する。

意味

当該servicePortをツール上で表示するかしないかを設定する。本属性が省略されている場合にはTrue(表示)が設定されているとみなす。省略可能属性。

2.4.7 execution_context_ext

execution_context_extはexecutionContextの付加情報に関するメタ情報を記述するための親要素である。

属性

execution_context_ext		
no attributes		
relationships		
Properties	0..*	property

2.4.8 dataport_connector_ext

dataport_connector_extはdataPort間の接続に関する付加情報のメタ情報を記述するための親要素である。

RTSプロファイル(PIM)の接続情報部分の拡大図を図10に再掲する。

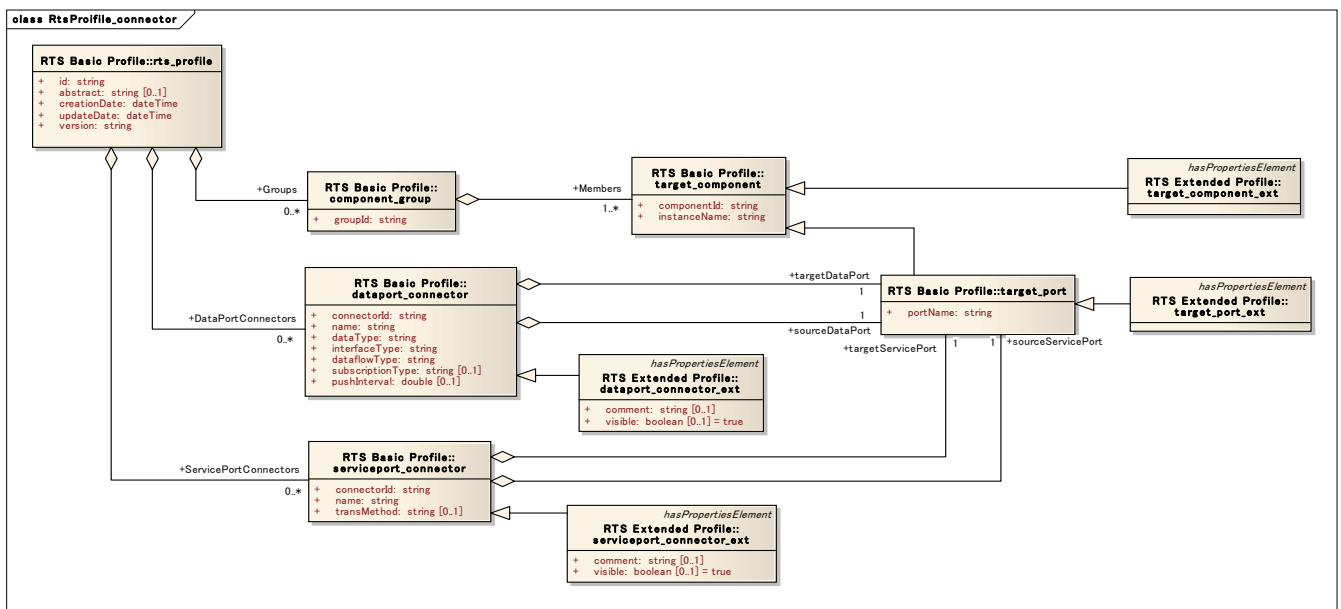


Figure 10 RTS プロファイル(接続情報部分)

属性

dataport_connector_ext		
attributes		
comment	0..1	string
visible	0..1	boolean
relationships		
Properties	0..*	property

2.4.8.1 comment

概要

dataPortのコネクタに対する各種コメントを記述する。

意味

dataPortのコネクタに対する簡単なコメントを記述する。本属性は各種ツールで表示することを意図している(本属性の表示/非表示は各種ツール毎の設定に依存する)。省略可能属性。

2.4.8.2 visible

概要

dataPortのコネクタの表示/非表示を設定する。

意味

当該dataPortコネクタをツール上で表示するかしないかを設定する。本属性が省略されている場合にはTrue(表示)が設定されているとみなす。省略可能属性。

2.4.9 serviceport_connector_ext

serviceport_connector_extはservicePort間の接続に関する付加情報のメタ情報を記述するための親要素である。

属性

serviceport_connector_ext		
attributes		
comment	0..1	string
visible	0..1	boolean
relationships		
Properties	0..*	property

2.4.9.1 comment

概要

servicePortのコネクタに対する各種コメントを記述する。

意味

servicePortのコネクタに対する簡単なコメントを記述する。本属性は各種ツールで表示することを意図している(本属性の表示/非表示は各種ツール毎の設定に依存する)。省略可能属性。

2.4.9.2 visible

概要

servicePortのコネクタの表示/非表示を設定する。

意味

当該servicePortコネクタのツール上での表示/非表示を設定する。省略されている場合にはTrue(表示)が設定されているとみなす。省略可能属性。

属性

<i>condition_ext</i>		
no attributes		
relationships		
Properties	0..*	property

2.4.13 *property*

*property*はユーザが各要素に追加できる付加情報に関するメタ情報を記述する。

属性

<i>property</i>		
attributes		
name	1	string
value	0..1	string
no relationships		

2.4.13.1 *name*

概要

ユーザが定義するプロパティのキー情報を指定する。

意味

各種要素にユーザが追加定義するプロパティ情報のキーを指定する。必須属性。

2.4.13.2 *value*

概要

ユーザが定義するプロパティ情報を指定する。

意味

各種要素にユーザが追加定義するプロパティ情報の内容を指定する。省略可能属性。

3 プラットフォーム依存モデル

相互運用性を保証するために本仕様書では2種類のプラットフォーム依存モデル (PSM: Platform Specific Model) を定義する。これらのPSMはsection 1.4 の準拠性に従って定義される。

Contents

Section Title	Page
UML-XML変換	43
UML-XML変換	46

3.1 UML-XML 変換

本節ではUMLからXMLへの変換ルールのサブセットを示す。PIMで定義されたデータモデルを、各種ツールなどで利用可能なXML形式で表現するために必要なXMLスキーマを定義する。

3.1.1 基本型

PIMでを使用した基本型は、以下のようにXMLスキーマの「基本データ型」にマッピングする。

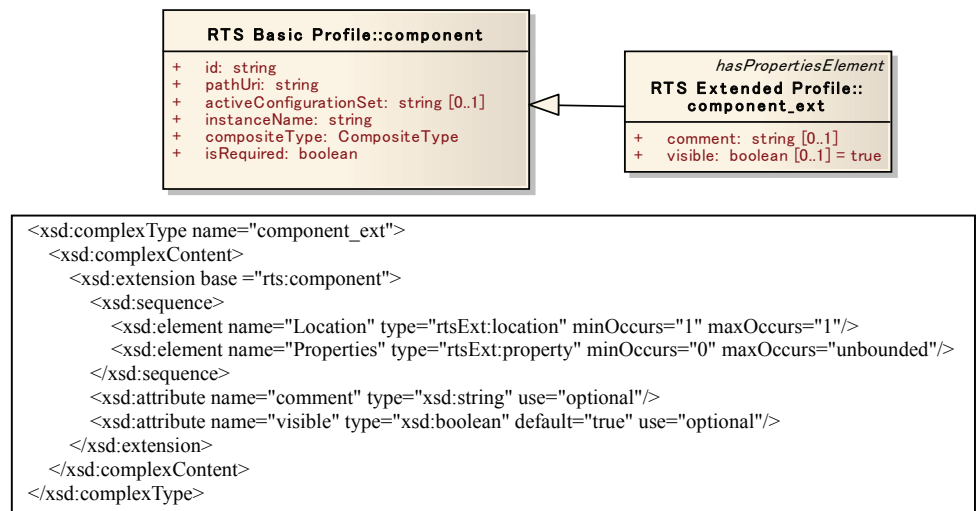
- string → xsd:string
- double → xsd:double
- integer → xsd:integer
- dateTime → xsd:dateTime
- boolean → xsd:boolean

3.1.2 パッケージと名前空間

PIMにて定義したパッケージは、以下のようにXMLスキーマの名前空間としてマッピングする。

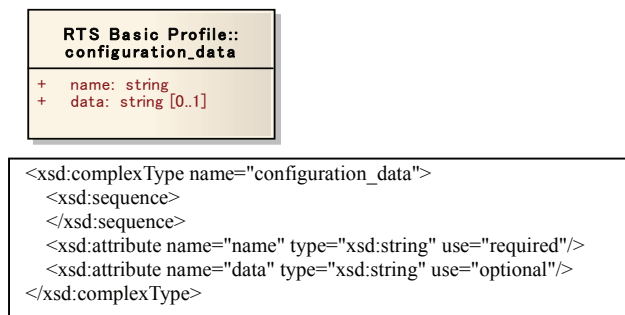
- RTS Basic Profile → rts
- RTS Extended Profile → rtsExt

また、各パッケージ間で汎化関係を用いて定義した要素は、以下のようにXMLスキーマではextensionを用いて定義する。

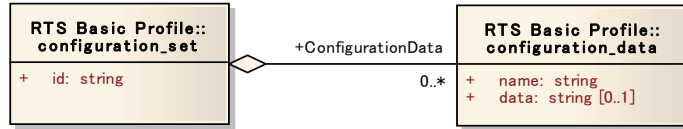


3.1.3 クラス

PIMにてクラスとして定義した要素は、以下のようにXMLスキーマではcomplexTypeの要素にマッピングする。また各クラスの属性として定義した要素は、XMLスキーマにおいても属性にマッピングする。更に属性の多重度は、use属性にマッピングする。

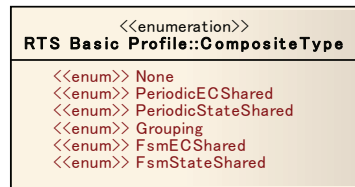


PIMにて集約関係で表現した要素は、XMLスキーマではsequenceを利用した子要素としてマッピングする。そしてロール名は要素名に、多重度はminOccurs属性およびmaxOccurs属性にマッピングする。



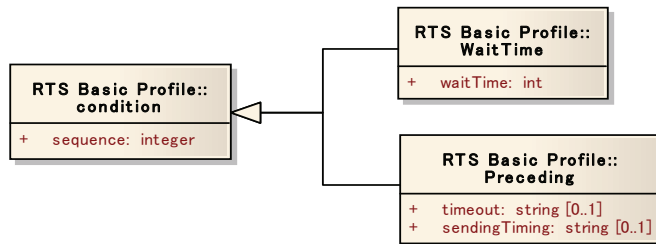
```
<xsd:complexType name="configuration_set">
  <xsd:sequence>
    <xsd:element name="ConfigurationData" type="rts:configuration_data" minOccurs="0" maxOccurs="unbounded"/>
  </xsd:sequence>
  <xsd:attribute name="id" type="xsd:string" use="required"/>
</xsd:complexType>
```

PIMにてenum型として定義した要素は、XMLスキーマではrestrictionおよびenumerationを用いて定義した型にマッピングする。



```
<xsd:attribute name="compositeType">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="None"/>
      <xsd:enumeration value="PeriodicECShared"/>
      <xsd:enumeration value="PeriodicStateShared"/>
      <xsd:enumeration value="Grouping"/>
      <xsd:enumeration value="FsmECShared"/>
      <xsd:enumeration value="FsmStateShared"/>
    </xsd:restriction>
  </xsd:simpleType>
</xsd:attribute>
```

同一パッケージ内で汎化関係を用いて定義した要素は、XMLスキーマではchoiceを利用した子要素としてマッピングする。



```
<xsd:complexType name="condition">
  <xsd:sequence>
    <xsd:choice>
      <xsd:element name="WaitTime" type="rts:waittime"/>
      <xsd:element name="Preceding" type="rts:preceding"/>
    </xsd:choice>
    <xsd:element name="TargetComponent" type="rts:target_executioncontext"/>
  </xsd:sequence>
  <xsd:attribute name="sequence" type="xsd:integer" use="required"/>
</xsd:complexType>
```

3.2 UML-YAML 変換

本節ではUMLからYAMLへの変換ルールのサブセットを示す。PIMで定義されたデータモデルを、各種ツールなどで利用可能なYAML形式で表現するために必要なYAMLスキーマを定義する。

なお、YAMLスキーマ仕様に関しては標準仕様がまだ策定されていない。そのため、今回は以下の仕様をベースにスキーマ作成を行った。

Kwalify (<http://jp.rubyist.net/magazine/?0012-YAML>)

3.2.1 基本型

PIMで使用した基本型は、以下のようにYAMLスキーマの「基本データ型」にマッピングする。

- string → text
- double → number
- integer → int
- dateTime → 各項目毎にint
- boolean → bool

3.2.2 パッケージと名前空間

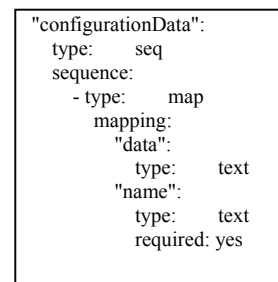
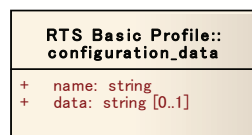
PIMにて定義したパッケージは、以下のようにYAMLスキーマの名前空間としてマッピングする。

- RTS Basic Profile → なし
- RTS Extended Profile → rtsExt

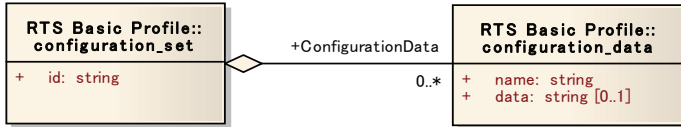
各名前空間に属する要素については、タグ名の前に「::」区切りにて名前空間接頭語を付加する。

3.2.3 クラス

PIMにてクラスとして定義した要素は、YAMLスキーマでは以下のようにマッピングを利用して定義する。また各クラスの属性として定義した要素は、マッピング対象の子要素として定義する。要素および属性の多重度は、required属性で表現する。また、多重度が「*」で設定されている要素に関しては、マッピングを子要素とするシーケンスを用いて定義する。



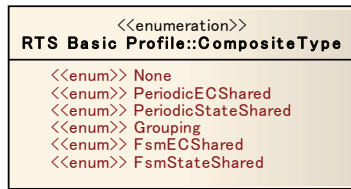
PIMにて集約関係で表現した要素は、YAMLスキーマではsequenceを利用した子要素としてマッピングする。そしてロール名は要素名にマッピングする。



```

"configurationSets":
  type: seq
  sequence:
    - type: map
      mapping:
        "configurationData":
          type: seq
          sequence:
            - type: map
              mapping:
                "data":
                  type: text
                "name":
                  type: text
                  required: yes
            "id":
              type: text
              required: yes
  
```

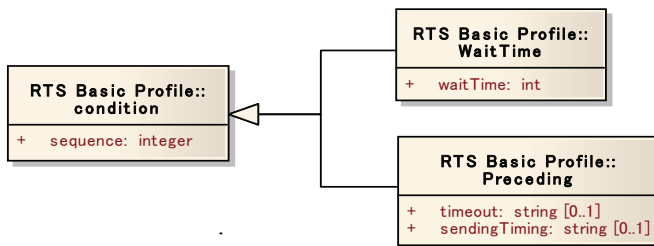
PIMにてenum型として定義した要素は、YAMLスキーマではenumを用いて定義する。



```

"compositeType":
  type: text
  required: yes
  enum:
    - None
    - PeriodicECShared
    - PeriodicStateShared
    - Grouping
    - FsmECShared
    - FsmStateShared
  
```

汎化関係を用いて定義した要素は、YAMLスキーマではマッピングを利用した子要素としてマッピングする。



```

"condition":
  type: map
  mapping:
    "waitTime":
      type: map
      mapping:
        "waitTime":
          type: int
          required: yes
    "preceding":
      type: map
      mapping:
        "sendingTiming":
          type: text
        "timeout":
          type: text
  
```

4 Appendix: 各種定義

4.1 *kind (execution_context)*

PERIODIC
EVENT DRIVEN
OTHER

5 Appendix: XMLスキーマ

5.1 RTS 基本 Profile(RtsProfile.xsd)

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:rts="http://www.openrtp.org/namespaces/rts"
  targetNamespace="http://www.openrtp.org/namespaces/rts"
  elementFormDefault="qualified"
  attributeFormDefault="qualified">

  <xsd:element name="RtsProfile" type="rts:rts_profile"/>
  <xsd:complexType name="rts_profile">
    <xsd:sequence>
      <xsd:element name="Components" type="rts:component" minOccurs="0" maxOccurs="unbounded"/>
      <xsd:element name="Groups" type="rts:componentGroup" minOccurs="0" maxOccurs="unbounded"/>
      <xsd:element name="DataPortConnectors" type="rts:dataport_connector" minOccurs="0" maxOccurs="unbounded"/>
      <xsd:element name="ServicePortConnectors" type="rts:serviceport_connector" minOccurs="0" maxOccurs="unbounded"/>
      <xsd:element name="StartUp" type="rts:startup" minOccurs="0" maxOccurs="1"/>
      <xsd:element name="ShutDown" type="rts:shutdown" minOccurs="0" maxOccurs="1"/>
      <xsd:element name="Activation" type="rts:activation" minOccurs="0" maxOccurs="1"/>
      <xsd:element name="Deactivation" type="rts:deactivation" minOccurs="0" maxOccurs="1"/>
      <xsd:element name="Resetting" type="rts:resetting" minOccurs="0" maxOccurs="1"/>
      <xsd:element name="Initializing" type="rts:initialize" minOccurs="0" maxOccurs="1"/>
      <xsd:element name="Finalizing" type="rts:finalize" minOccurs="0" maxOccurs="1"/>
    </xsd:sequence>
    <xsd:attribute name="id" type="xsd:string" use="required"/>
    <xsd:attribute name="version" type="xsd:string" use="required"/>
    <xsd:attribute name="abstract" type="xsd:string" use="optional"/>
    <xsd:attribute name="creationDate" type="xsd:dateTime" use="required"/>
    <xsd:attribute name="updateDate" type="xsd:dateTime" use="required"/>
  </xsd:complexType>

  <xsd:complexType name="component">
    <xsd:sequence>
      <xsd:element name="DataPorts" type="rts:dataport" minOccurs="0" maxOccurs="unbounded"/>
      <xsd:element name="ServicePorts" type="rts:serviceport" minOccurs="0" maxOccurs="unbounded"/>
      <xsd:element name="ConfigurationSets" type="rts:configuration_set" minOccurs="0" maxOccurs="unbounded"/>
      <xsd:element name="ExecutionContexts" type="rts:execution_context" minOccurs="0" maxOccurs="unbounded"/>
      <xsd:element name="Participants" type="rts:participants" minOccurs="0" maxOccurs="unbounded"/>
    </xsd:sequence>
    <xsd:attribute name="id" type="xsd:string" use="required"/>
    <xsd:attribute name="pathUri" type="xsd:string" use="required"/>
    <xsd:attribute name="instanceName" type="xsd:string" use="required"/>
    <xsd:attribute name="activeConfigurationSet" type="xsd:string" use="optional"/>
  </xsd:complexType>
</xsd:schema>
```

```

<xsd:attribute name="compositeType">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="None"/>
      <xsd:enumeration value="AllShared"/>
      <xsd:enumeration value="ECShared"/>
      <xsd:enumeration value="NonShared"/>
      <xsd:enumeration value="NonShared"/>
      <xsd:enumeration value="NonShared"/>
    </xsd:restriction>
  </xsd:simpleType>
</xsd:attribute>
<xsd:attribute name="isRequired" type="xsd:boolean" use="required"/>
</xsd:complexType>

<xsd:complexType name="dataport">
  <xsd:attribute name="name" type="xsd:string" use="required"/>
</xsd:complexType>

<xsd:complexType name="serviceport">
  <xsd:attribute name="name" type="xsd:string" use="required"/>
</xsd:complexType>

<xsd:complexType name="configuration_set">
  <xsd:sequence>
    <xsd:element name="ConfigurationData" type="rts:configuration_data" minOccurs="0" maxOccurs="unbounded"/>
  </xsd:sequence>
  <xsd:attribute name="id" type="xsd:string" use="required"/>
</xsd:complexType>

<xsd:complexType name="configuration_data">
  <xsd:sequence>
    <xsd:sequence>
      <xsd:attribute name="name" type="xsd:string" use="required"/>
      <xsd:attribute name="data" type="xsd:string" use="optional"/>
    </xsd:sequence>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="execution_context">
  <xsd:sequence>
    <xsd:element name="Participants" type="rts:target_component" minOccurs="0" maxOccurs="unbounded"/>
  </xsd:sequence>
  <xsd:attribute name="id" type="xsd:string" use="required"/>
  <xsd:attribute name="kind" type="xsd:string" use="required"/>
  <xsd:attribute name="rate" type="xsd:double" use="optional"/>
</xsd:complexType>

<xsd:complexType name="participants">
  <xsd:sequence>
    <xsd:element name="Participant" type="rts:target_component" minOccurs="1" maxOccurs="1"/>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="componentGroup">
  <xsd:sequence>
    <xsd:element name="Members" type="rts:target_component" minOccurs="1" maxOccurs="unbounded"/>
  </xsd:sequence>
  <xsd:attribute name="groupId" type="xsd:string" use="required"/>
</xsd:complexType>

<xsd:complexType name="dataport_connector">
  <xsd:sequence>
    <xsd:element name="sourceDataPort" type="rts:target_port" minOccurs="1" maxOccurs="1"/>
    <xsd:element name="targetDataPort" type="rts:target_port" minOccurs="1" maxOccurs="1"/>
  </xsd:sequence>
  <xsd:attribute name="connectorId" type="xsd:string" use="required"/>
  <xsd:attribute name="name" type="xsd:string" use="required"/>
  <xsd:attribute name="dataType" type="xsd:string" use="required"/>
  <xsd:attribute name="interfaceType" type="xsd:string" use="required"/>
  <xsd:attribute name="dataflowType" type="xsd:string" use="required"/>

```

```

    <xsd:attribute name="subscriptionType" type="xsd:string" use="optional"/>
    <xsd:attribute name="pushInterval" type="xsd:double" use="optional"/>
  </xsd:complexType>

  <xsd:complexType name="serviceport_connector">
    <xsd:sequence>
      <xsd:element name="sourceServicePort" type="rts:target_port" minOccurs="1" maxOccurs="1"/>
      <xsd:element name="targetServicePort" type="rts:target_port" minOccurs="1" maxOccurs="1"/>
    </xsd:sequence>
    <xsd:attribute name="connectorId" type="xsd:string" use="required"/>
    <xsd:attribute name="name" type="xsd:string" use="required"/>
    <xsd:attribute name="transMethod" type="xsd:string" use="optional"/>
  </xsd:complexType>

  <xsd:complexType name="target_component">
    <xsd:sequence>
    </xsd:sequence>
    <xsd:attribute name="componentId" type="xsd:string" use="required"/>
    <xsd:attribute name="instanceName" type="xsd:string" use="required"/>
  </xsd:complexType>

  <xsd:complexType name="target_port">
    <xsd:complexContent>
      <xsd:extension base="rts:target_component">
        <xsd:attribute name="portName" type="xsd:string" use="required"/>
      </xsd:extension>
    </xsd:complexContent>
  </xsd:complexType>

  <xsd:complexType name="target_executioncontext">
    <xsd:complexContent>
      <xsd:extension base="rts:target_component">
        <xsd:attribute name="id" type="xsd:string" use="required"/>
      </xsd:extension>
    </xsd:complexContent>
  </xsd:complexType>

  <xsd:complexType name="message_sending">
    <xsd:sequence>
      <xsd:element name="targets" type="rts:condition" minOccurs="0" maxOccurs="unbounded"/>
    </xsd:sequence>
  </xsd:complexType>

  <xsd:complexType name="startup">
    <xsd:complexContent>
      <xsd:extension base="rts:message_sending">
        <xsd:sequence>
        </xsd:sequence>
      </xsd:extension>
    </xsd:complexContent>
  </xsd:complexType>

  <xsd:complexType name="shutdown">
    <xsd:complexContent>
      <xsd:extension base="rts:message_sending">
        <xsd:sequence>
        </xsd:sequence>
      </xsd:extension>
    </xsd:complexContent>
  </xsd:complexType>

  <xsd:complexType name="activation">
    <xsd:complexContent>
      <xsd:extension base="rts:message_sending">
        <xsd:sequence>
        </xsd:sequence>
      </xsd:extension>
    </xsd:complexContent>
  </xsd:complexType>

```

```

<xsd:complexType name="deactivation">
  <xsd:complexContent>
    <xsd:extension base="rts:message_sending">
      <xsd:sequence>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="resetting">
  <xsd:complexContent>
    <xsd:extension base="rts:message_sending">
      <xsd:sequence>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="initialize">
  <xsd:complexContent>
    <xsd:extension base="rts:message_sending">
      <xsd:sequence>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="finalize">
  <xsd:complexContent>
    <xsd:extension base="rts:message_sending">
      <xsd:sequence>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="condition">
  <xsd:sequence>
    <xsd:choice>
      <xsd:element name="WaitTime" type="rts:waittime"/>
      <xsd:element name="Preceding" type="rts:preceding"/>
    </xsd:choice>
    <xsd:element name="TargetComponent" type="rts:target_executioncontext"/>
  </xsd:sequence>
  <xsd:attribute name="sequence" type="xsd:integer" use="required"/>
</xsd:complexType>

<xsd:complexType name="preceding">
  <xsd:sequence>
    <xsd:element name="PrecedingComponents" type="rts:target_executioncontext" minOccurs="0" maxOccurs="unbounded"/>
  </xsd:sequence>
  <xsd:attribute name="timeout" type="xsd:string" use="optional"/>
  <xsd:attribute name="sendingTiming" type="xsd:string" use="optional"/>
</xsd:complexType>

<xsd:complexType name="waittime">
  <xsd:sequence>
  </xsd:sequence>
  <xsd:attribute name="waitTime" type="xsd:integer" use="optional"/>
</xsd:complexType>

</xsd:schema>

```

5.2 RTS 拡張 Profile(RtsProfile_ext.xsd)

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:rtsExt="http://www.openrtsp.org/namespaces/rts_ext"
  xmlns:rts="http://www.openrtsp.org/namespaces/rts"
  targetNamespace="http://www.openrtsp.org/namespaces/rts_ext"
  elementFormDefault="qualified"
  attributeFormDefault="qualified">

  <xsd:import namespace="http://www.openrtsp.org/namespaces/rts" schemaLocation="RtsProfile.xsd"/>

  <xsd:complexType name="rts_profile_ext">
    <xsd:complexContent>
      <xsd:extension base="rts:rts_profile">
        <xsd:sequence>
          <xsd:element name="VersionUpLogs" type="xsd:string" minOccurs="0" maxOccurs="unbounded"/>
          <xsd:element name="Properties" type="rtsExt:property" minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:attribute name="comment" type="xsd:string" use="optional"/>
      </xsd:extension>
    </xsd:complexContent>
  </xsd:complexType>

  <xsd:complexType name="component_ext">
    <xsd:complexContent>
      <xsd:extension base="rts:component">
        <xsd:sequence>
          <xsd:element name="Location" type="rtsExt:location" minOccurs="1" maxOccurs="1"/>
          <xsd:element name="Properties" type="rtsExt:property" minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:attribute name="comment" type="xsd:string" use="optional"/>
        <xsd:attribute name="visible" type="xsd:boolean" default="true" use="optional"/>
      </xsd:extension>
    </xsd:complexContent>
  </xsd:complexType>

  <xsd:complexType name="location">
    <xsd:attribute name="x" type="xsd:integer" use="required"/>
    <xsd:attribute name="y" type="xsd:integer" use="required"/>
    <xsd:attribute name="height" type="xsd:integer" use="required"/>
    <xsd:attribute name="width" type="xsd:integer" use="required"/>
    <xsd:attribute name="direction" use="required">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:enumeration value="DOWN"/>
          <xsd:enumeration value="RIGHT"/>
          <xsd:enumeration value="UP"/>
          <xsd:enumeration value="LEFT"/>
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:attribute>
  </xsd:complexType>

  <xsd:complexType name="dataport_ext">
    <xsd:complexContent>
      <xsd:extension base="rts:dataport">
        <xsd:sequence>
          <xsd:element name="Properties" type="rtsExt:property" minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:attribute name="comment" type="xsd:string" use="optional"/>
        <xsd:attribute name="visible" type="xsd:boolean" default="true" use="optional"/>
      </xsd:extension>
    </xsd:complexContent>
  </xsd:complexType>
</xsd:schema>
```

```

<xsd:complexType name="serviceport_ext">
  <xsd:complexContent>
    <xsd:extension base="rts:serviceport">
      <xsd:sequence>
        <xsd:element name="Properties" type="rtsExt:property" minOccurs="0" maxOccurs="unbounded"/>
      </xsd:sequence>
      <xsd:attribute name="comment" type="xsd:string" use="optional"/>
      <xsd:attribute name="visible" type="xsd:boolean" default="true" use="optional"/>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="execution_context_ext">
  <xsd:complexContent>
    <xsd:extension base="rts:execution_context">
      <xsd:sequence>
        <xsd:element name="Properties" type="rtsExt:property" minOccurs="0" maxOccurs="unbounded"/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="target_component_ext">
  <xsd:complexContent>
    <xsd:extension base="rts:target_component">
      <xsd:sequence>
        <xsd:element name="Properties" type="rtsExt:property" minOccurs="0" maxOccurs="unbounded"/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="target_port_ext">
  <xsd:complexContent>
    <xsd:extension base="rts:target_port">
      <xsd:sequence>
        <xsd:element name="Properties" type="rtsExt:property" minOccurs="0" maxOccurs="unbounded"/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="dataport_connector_ext">
  <xsd:complexContent>
    <xsd:extension base="rts:dataport_connector">
      <xsd:sequence>
        <xsd:element name="Properties" type="rtsExt:property" minOccurs="0" maxOccurs="unbounded"/>
      </xsd:sequence>
      <xsd:attribute name="comment" type="xsd:string" use="optional"/>
      <xsd:attribute name="visible" type="xsd:boolean" default="true" use="optional"/>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="serviceport_connector_ext">
  <xsd:complexContent>
    <xsd:extension base="rts:serviceport_connector">
      <xsd:sequence>
        <xsd:element name="Properties" type="rtsExt:property" minOccurs="0" maxOccurs="unbounded"/>
      </xsd:sequence>
      <xsd:attribute name="comment" type="xsd:string" use="optional"/>
      <xsd:attribute name="visible" type="xsd:boolean" default="true" use="optional"/>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

```

```
<xsd:complexType name="condition_ext">
  <xsd:complexContent>
    <xsd:extension base="rts:condition">
      <xsd:sequence>
        <xsd:element name="Properties" type="rtsExt:property" minOccurs="0" maxOccurs="unbounded"/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="property">
  <xsd:attribute name="name" type="xsd:string" use="required"/>
  <xsd:attribute name="value" type="xsd:string" use="optional"/>
</xsd:complexType>

</xsd:schema>
```

6 Appendix: YAMLスキーマ

```
type: map
mapping:
  "rtsProfile":
    type: map
    mapping:
      "abstract":
        type: text
      "activation":
        type: map
        mapping:
          "targets": &targets
          type: seq
          sequence:
            - type: map
              mapping:
                "condition":
                  type: map
                  mapping:
                    "rtsExt::properties": &properties
                    type: seq
                    sequence:
                      - type: map
                        mapping:
                          "name":
                            type: text
                            required: yes
                          "value":
                            type: text
                "waitTime":
                  type: map
                  mapping:
                    "waitTime":
                      type: int
                      required: yes
```

```
"preceding":
  type: map
  mapping:
    "precedingComponents":
      type: seq
      sequence:
        - type: map
          mapping:
            "componentId":
              type: text
              required: yes
            "id":
              type: text
            "instanceName":
              type: text
              required: yes
    "sendingTiming":
      type: text
    "timeout":
      type: text
"sequence":
  type: int
  required: yes
"targetComponent":
  type: map
  required: yes
  mapping:
    "componentId":
      type: text
      required: yes
    "id":
      type: text
    "instanceName":
      type: text
      required: yes
```



```

"rtsExt::comment":
  type: text
"components":
  type: seq
  sequence:
    - type: map
      mapping:
        "activeConfigurationSet":
          type: text
        "rtsExt::comment":
          type: text
        "compositeType":
          type: text
          required: yes
          enum:
            - None
            - PeriodicECShared
            - PeriodicStateShared
            - Grouping
            - FsmECShared
            - FsmStateShared
        "configurationSets":
          type: seq
          sequence:
            - type: map
              mapping:
                "configurationData":
                  type: seq
                  sequence:
                    - type: map
                      mapping:
                        "data":
                          type: text
                        "name":
                          type: text
                          required: yes
                "id":
                  type: text
                  required: yes
        "dataPorts":
          type: seq
          sequence:
            - type: map
              mapping:
                "rtsExt::comment":
                  type: text
                "name":
                  type: text
                  required: yes
                "rtsExt::properties": *properties
                "rtsExt::visible":
                  type: bool
        "executionContexts":
          type: seq
          sequence:
            - type: map
              mapping:
                "id":
                  type: text
                  required: yes
                "kind":
                  type: text
                  required: yes

```

```

    "participants":
      type: seq
      sequence:
        - type: map
          mapping:
            "componentId":
              type: text
              required: yes
            "instanceName":
              type: text
              required: yes
            "targetComponent":
              type: map
              mapping:
                "rtsExt::properties": *properties
            "targetExecutioncontExt":
              type: map
              mapping:
                "id":
                  type: text
            "targetPort":
              type: map
              mapping:
                "portName":
                  type: text
                  required: yes
                "rtsExt::properties": *properties
            "rtsExt::properties": *properties
          "rate":
            type: number
  "id":
    type: text
    required: yes
  "instanceName":
    type: text
    required: yes
  "isRequired":
    type: bool
    required: yes
  "rtsExt::location":
    type: map
    required: yes
    mapping:
      "direction":
        type: text
        required: yes
        enum:
          - DOWN
          - RIGHT
          - UP
          - LEFT
      "height":
        type: int
        required: yes
      "width":
        type: int
        required: yes
      "x":
        type: int
        required: yes

```

```

    "y":
      type: int
      required: yes
    "participants":
      type: seq
      sequence:
        - type: map
          mapping:
            "participant":
              type: map
              required: yes
              mapping:
                "componentId":
                  type: text
                  required: yes
                "instanceName":
                  type: text
                  required: yes
                "targetComponent":
                  type: map
                  mapping:
                    "rtsExt::properties": *properties
                "targetExecutioncontExt":
                  type: map
                  mapping:
                    "id":
                      type: text
                "targetPort":
                  type: map
                  mapping:
                    "portName":
                      type: text
                      required: yes
                    "rtsExt::properties": *properties
            "pathUri":
              type: text
              required: yes
            "rtsExt::properties": *properties
            "servicePorts":
              type: seq
              sequence:
                - type: map
                  mapping:
                    "rtsExt::comment":
                      type: text
                    "name":
                      type: text
                      required: yes
                    "rtsExt::properties": *properties
                    "rtsExt::visible":
                      type: bool
            "rtsExt::visible":
              type: bool

```

```

    "creationDate": &dateType
      type: map
      required: yes
      mapping:
        "day":
          type: int
          required: yes
        "hour":
          type: int
          required: yes
        "minute":
          type: int
          required: yes
        "month":
          type: int
          required: yes
        "second":
          type: int
          required: yes
        "year":
          type: int
          required: yes
    "dataPortConnectors":
      type: seq
      sequence:
        - type: map
          mapping:
            "rtsExt::comment":
              type: text
            "connectorId":
              type: text
              required: yes
            "dataflowType":
              type: text
              required: yes
            "dataType":
              type: text
              required: yes
            "interfaceType":
              type: text
              required: yes
            "name":
              type: text
              required: yes
            "rtsExt::properties": *properties
            "pushInterval":
              type: number
            "subscriptionType":
              type: text
            "targetDataPort": &targetPort
              type: map
              required: yes
              mapping:
                "componentId":
                  type: text
                  required: yes
                "instanceName":
                  type: text
                  required: yes
                "portName":
                  type: text
                  required: yes

```

```

    "rtsExt::properties": *properties
    "sourceDataPort": *targetPort
    "rtsExt::visible":
      type: bool
  "deactivation":
    type: map
    mapping:
      "targets": *targets
  "finalizing":
    type: map
    mapping:
      "targets": *targets
  "groups":
    type: seq
    sequence:
      - type: map
        mapping:
          "groupId":
            type: text
            required: yes
          "members":
            type: seq
            required: yes
            sequence:
              - type: map
                mapping:
                  "componentId":
                    type: text
                    required: yes
                  "instanceName":
                    type: text
                    required: yes
                  "targetComponent":
                    type: map
                    mapping:
                      "rtsExt::properties": *properties
                  "targetExecutioncontExt":
                    type: map
                    mapping:
                      "id":
                        type: text
                      "targetPort":
                        type: map
                        mapping:
                          "portName":
                            type: text
                            required: yes
                          "rtsExt::properties": *properties

```

```

  "id":
    type: text
    required: yes
  "initializing":
    type: map
    mapping:
      "targets": *targets
  "rtsExt::properties": *properties
  "resetting":
    type: map
    mapping:
      "targets": *targets
  "servicePortConnectors":
    type: seq
    sequence:
      - type: map
        mapping:
          "rtsExt::comment":
            type: text
          "connectorId":
            type: text
            required: yes
          "name":
            type: text
            required: yes
          "rtsExt::properties": *properties
          "transMethod":
            type: text
          "targetServicePort": *targetPort
          "sourceServicePort": *targetPort
          "rtsExt::visible":
            type: bool
  "shutDown":
    type: map
    mapping:
      "targets": *targets
  "stratUp":
    type: map
    mapping:
      "targets": *targets
  "updateDate": *dateType
  "version":
    type: text
    required: yes
  "rtsExt::versionUpLogs":
    type: seq
    sequence:
      - type: text

```

7 Appendix: サンプル

7.1 RT システム仕様

<i>RtsProfile</i>		
id	RTSystem:jp.go.aist:SampleRTS:1.0.0	
abstract	Sample Abstract	
creationDate	2008/5/6 13:49:53	
updateDate	2008/5/6 13:49:53	
version	0.2	
comment	Sample RTSystem	
<i>Component1</i>		
id	RTC:SampleVendor:SampleCategory:SampleComponent:1.0.0	
instanceName	SampleComponent 1	
pathUri	file://localhost/C:\RTM\XML_Sample\XML\SampleXML.xml	
compositeType	None	
activeConfigurationSet	configSet 1	
isRequired	TRUE	
visible	TRUE	
<i>DataPorts</i>		
name	Comp1_output1	
visible	TRUE	
name	Comp1_output2	
visible	FALSE	
name	Comp1_output3	
visible	TRUE	
<i>ServicePorts</i>		
name	SrvPort2	
<i>ConfigurationSet1</i>		
id	configSet 1	
ConfigurationData1	name	name 1
	data	value
ConfigurationData2	name	name 2
	data	value
id	configSet 2	
ConfigurationData1	name	name 1
	data	value
<i>ExecutionContexts</i>		
id	Comp1Ec1	
kind	PeriodicExecutionContext	
rate	1000	

Location		
direction	right	
x	93	
y	93	
height	-1	
width	-1	
Component2		
id	RTC:SampleVendor:SampleCategory:SampleComponent2:1.0.0	
instanceName	SampleComponent2 1	
pathUri	file://localhost/C:%5BRTM%5DXML_Sample%5DXML%5D_SampleXML2.xml	
compositeType	None	
activeConfigurationSet	configSet 1	
isRequired	TRUE	
visible	TRUE	
DataPorts		
name	Comp2 inort1	
visible	TRUE	
ConfigurationSet1		
id	configSet 1	
ConfigurationData1	name	name 1
	data	value
ExecutionContexts		
id	Comp2Ec1	
kind	PeriodicExecutionContext	
rate	500	
Location		
direction	right	
x	93	
y	276	
height	-1	
width	-1	
Component3		
id	RTC:SampleVendor:SampleCategory:SampleComponent3:0.5.0	
instanceName	SampleComponent3 1	
pathUri	file://localhost/C:%5BRTM%5DXML_Sample%5DXML%5D_SampleXML3.xml	
compositeType	None	
isRequired	TRUE	
visible	TRUE	
DataPorts		
name	Comp3 inort2	
visible	TRUE	
ServicePorts		
name	SrvPort4	
ExecutionContexts		
id	Comp3Ec1	
kind	PeriodicExecutionContext	
rate	300	
Location		
direction	left	
x	93	
y	93	
height	-1	
width	-1	
ServicePortConnectors		
connectorId	9fbb4aa9-cd3f-4723-9375-b8e1201d0b6f	
name	SrvPort2_SrvPort4	
sourceServicePort	componentId	RTC:SampleVendor:SampleCategory:SampleComponent:1.0.0
	instanceName	SampleComponent 1
	portName	SrvPort2
targetServicePort	componentId	RTC:SampleVendor:SampleCategory:SampleComponent3:0.5.0
	instanceName	SampleComponent3 1
	portName	SrvPort4

<i>DataPortConnectors</i>			
connectorId	9c477198-dbf4-4298-9713-c5e1b1b30607		
name	Comp1_outport1_Comp2_inport1		
dataType	RTC::TimedLong		
interfaceType	CORBA Any		
dataflowType	PUSH		
subscriptionType	Flush		
sourceDataPort	componentId	RTC:SampleVendor:SampleCategory:SampleComponent:1.0.0	
	instanceName	SampleComponent 1	
	portName	Comp1_outport1	
targetDataPort	componentId	RTC:SampleVendor:SampleCategory:SampleComponent2:1.0.0	
	instanceName	SampleComponent2 1	
	portName	Comp2_inort1	
connectorId	c5e1b1b306-9713-4298-dbf4-9c47719807		
name	Comp1_outport3_Comp3_inport2		
dataType	RTC::TimedDouble		
interfaceType	CORBA Any		
dataflowType	PUSH		
subscriptionType	Flush		
sourceDataPort	componentId	RTC:SampleVendor:SampleCategory:SampleComponent:1.0.0	
	instanceName	SampleComponent 1	
	portName	Comp1_outport3	
targetDataPort	componentId	RTC:SampleVendor:SampleCategory:SampleComponent3:0.5.0	
	instanceName	SampleComponent3 1	
	portName	Comp3_inort2	
<i>StratUp</i>			
sequence			
1	componentId	RTC:SampleVendor:SampleCategory:SampleComponent:1.0.0	
	instanceName	SampleComponent 1	
	EC id	Comp1Ec1	
	WaitTime	1000	
2	componentId	RTC:SampleVendor:SampleCategory:SampleComponent2:1.0.0	
	instanceName	SampleComponent2 1	
	EC id	Comp2Ec1	
	WaitTime	500	
3	componentId	RTC:SampleVendor:SampleCategory:SampleComponent3:0.5.0	
	instanceName	SampleComponent3 1	
	EC id	Comp3Ec1	
	WaitTime	750	
<i>ShutDown</i>			
sequence			
1	componentId	RTC:SampleVendor:SampleCategory:SampleComponent3:0.5.0	
	instanceName	SampleComponent3 1	
	EC id	Comp3Ec1	
	WaitTime	300	
2	componentId	RTC:SampleVendor:SampleCategory:SampleComponent2:1.0.0	
	instanceName	SampleComponent2 1	
	EC id	Comp2Ec1	
	timeout	100	
	sendingTimin	SYNC	
	Preceding	componentId	RTC:SampleVendor:SampleCategory:SampleComponent3:0.5.0
		instanceName	SampleComponent3 1
		EC id	Comp3Ec1
3	componentId	RTC:SampleVendor:SampleCategory:SampleComponent:1.0.0	
	instanceName	SampleComponent 1	
	EC id	Comp1Ec1	
	timeout	100	
	sendingTimin	SYNC	
	Preceding	componentId	RTC:SampleVendor:SampleCategory:SampleComponent3:0.5.0
		instanceName	SampleComponent3 1
		EC id	Comp3Ec1
	Preceding	componentId	RTC:SampleVendor:SampleCategory:SampleComponent2:1.0.0
		instanceName	SampleComponent2 1
		EC id	Comp2Ec1

7.2 XML サンプル

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<rts:RtsProfile rtsExt:comment="Sample RTSystem" rts:updateDate="2008-05-06T13:49:53"
  rts:creationDate="2008-05-06T13:49:53"
  rts:abstract="Sample Abstract" rts:version="0.2" rts:id="RTSystem.jp.go.aist:SampleRTS:1.0.0"
  xmlns:rtsExt="http://www.openrtp.org/namespaces/rts_ext" xmlns:rts="http://www.openrtp.org/namespaces/rts"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <rts:Components xsi:type="rtsExt:component_ext" rtsExt:visible="true" rts:isRequired="true" rts:compositeType="None"
    rts:activeConfigurationSet="configSet_1" rts:instanceName="SampleComponent_1"
    rts:pathUri="file://localhost/C:\RTM\XML\Sample\XML\SampleXML.xml"
    rts:id="RTC:SampleVendor:SampleCategory:SampleComponent:1.0.0">
    <rts:DataPorts xsi:type="rtsExt:dataport_ext" rtsExt:visible="true" rts:name="Comp1_outport1"/>
    <rts:DataPorts xsi:type="rtsExt:dataport_ext" rtsExt:visible="true" rts:name="Comp1_outport2"/>
    <rts:DataPorts xsi:type="rtsExt:dataport_ext" rtsExt:visible="true" rts:name="Comp1_outport3"/>
    <rts:ServicePorts xsi:type="rtsExt:serviceport_ext" rts:name="SrvPort2"/>
    <rts:ConfigurationSets rts:id="configSet_1">
      <rts:ConfigurationData rts:data="value" rts:name="name_1"/>
      <rts:ConfigurationData rts:data="value" rts:name="name_2"/>
    </rts:ConfigurationSets>
    <rts:ExecutionContexts xsi:type="rtsExt:execution_context_ext" rts:rate="1000.0" rts:kind="PeriodicExecutionContext"
      rts:id="Comp1Ec1"/>
    <rtsExt:Location rtsExt:direction="right" rtsExt:width="-1" rtsExt:height="-1" rtsExt:y="93" rtsExt:x="93"/>
  </rts:Components>
  <rts:Components xsi:type="rtsExt:component_ext" rtsExt:visible="true" rts:isRequired="true" rts:compositeType="None"
    rts:activeConfigurationSet="configSet_1" rts:instanceName="SampleComponent2_1"
    rts:pathUri="file://localhost/C:\RTM\XML\Sample\XML\SampleXML2.xml"
    rts:id="RTC:SampleVendor:SampleCategory:SampleComponent2:1.0.0">
    <rts:DataPorts xsi:type="rtsExt:dataport_ext" rtsExt:visible="true" rts:name="Comp2_inport1"/>
    <rts:ExecutionContexts xsi:type="rtsExt:execution_context_ext" rts:rate="500.0" rts:kind="PeriodicExecutionContext"
      rts:id="Comp2Ec1"/>
    <rtsExt:Location rtsExt:direction="right" rtsExt:width="-1" rtsExt:height="-1" rtsExt:y="276" rtsExt:x="93"/>
  </rts:Components>
  <rts:Components xsi:type="rtsExt:component_ext" rtsExt:visible="true" rts:isRequired="true" rts:compositeType="None"
    rts:instanceName="SampleComponent3_1" rts:pathUri="file://localhost/C:\RTM\XML\Sample\XML\SampleXML3.xml"
    rts:id="RTC:SampleVendor:SampleCategory:SampleComponent3:0.5.0">
    <rts:DataPorts xsi:type="rtsExt:dataport_ext" rtsExt:visible="true" rts:name="Comp3_inport2"/>
    <rts:ServicePorts xsi:type="rtsExt:serviceport_ext" rts:name="SrvPort4"/>
    <rts:ExecutionContexts xsi:type="rtsExt:execution_context_ext" rts:rate="300.0" rts:kind="PeriodicExecutionContext"
      rts:id="Comp3Ec1"/>
    <rtsExt:Location rtsExt:direction="left" rtsExt:width="-1" rtsExt:height="-1" rtsExt:y="93" rtsExt:x="93"/>
  </rts:Components>
  <rts:DataPortConnectors xsi:type="rtsExt:dataport_connector_ext" rts:subscriptionType="Flush" rts:dataflowType="PUSH"
    rts:interfaceType="CORBA_Any" rts:dataType="RTC::TimedLong" rts:name="Comp1_outport1_Comp2_inport1"
    rts:connectorId="9c477198-dbf4-4298-9713-c5e1b1b30607">
    <rts:sourceDataPort xsi:type="rtsExt:target_port_ext" rts:portName="Comp1_outport1" rts:instanceName="SampleComponent_1"
      rts:componentId="RTC:SampleVendor:SampleCategory:SampleComponent:1.0.0"/>
    <rts:targetDataPort xsi:type="rtsExt:target_port_ext" rts:portName="Comp2_inport1" rts:instanceName="SampleComponent2_1"
      rts:componentId="RTC:SampleVendor:SampleCategory:SampleComponent2:1.0.0"/>
  </rts:DataPortConnectors>
  <rts:DataPortConnectors xsi:type="rtsExt:dataport_connector_ext" rts:subscriptionType="Flush" rts:dataflowType="PUSH"
    rts:interfaceType="CORBA_Any" rts:dataType="RTC::TimedDouble" rts:name="Comp1_outport3_Comp3_inport2"
    rts:connectorId="c5e1b1b306-9713-4298-dbf4-9c47719807">
    <rts:sourceDataPort xsi:type="rtsExt:target_port_ext" rts:portName="Comp1_outport3" rts:instanceName="SampleComponent_1"
      rts:componentId="RTC:SampleVendor:SampleCategory:SampleComponent:1.0.0"/>
    <rts:targetDataPort xsi:type="rtsExt:target_port_ext" rts:portName="Comp3_inport2" rts:instanceName="SampleComponent3_1"
      rts:componentId="RTC:SampleVendor:SampleCategory:SampleComponent3:0.5.0"/>
  </rts:DataPortConnectors>
  <rts:ServicePortConnectors xsi:type="rtsExt:serviceport_connector_ext" rts:name="SrvPort2_SrvPort4"
    rts:connectorId="9fbb4aa9-cd3f-4723-9375-b8e1201d0b6f">
    <rts:sourceServicePort xsi:type="rtsExt:target_port_ext" rts:portName="SrvPort4" rts:instanceName="SampleComponent3_1"
      rts:componentId="RTC:SampleVendor:SampleCategory:SampleComponent3:0.5.0"/>
  </rts:ServicePortConnectors>
  <rts:StartUp>
    <rts:targets xsi:type="rtsExt:condition_ext" rts:sequence="1">
      <rts:WaitTime rts:waitTime="1000"/>
      <rts:TargetComponent rts:id="Comp1Ec1" rts:instanceName="SampleComponent_1"
        rts:componentId="RTC:SampleVendor:SampleCategory:SampleComponent:1.0.0"/>
    </rts:targets>
  </rts:StartUp>
</rts:RtsProfile>
```



```

<rts:targets xsi:type="rtsExt:condition_ext" rts:sequence="2">
  <rts:WaitTime rts:waitTime="500"/>
  <rts:TargetComponent rts:id="Comp2Ec1" rts:instanceName="SampleComponent2_1"
    rts:componentId="RTC:SampleVendor:SampleCategory:SampleComponent2:1.0.0"/>
</rts:targets>
<rts:targets xsi:type="rtsExt:condition_ext" rts:sequence="3">
  <rts:WaitTime rts:waitTime="750"/>
  <rts:TargetComponent rts:id="Comp3Ec1" rts:instanceName="SampleComponent3_1"
    rts:componentId="RTC:SampleVendor:SampleCategory:SampleComponent3:0.5.0"/>
</rts:targets>
</rts:StartUp>
<rts:ShutDown>
  <rts:targets xsi:type="rtsExt:condition_ext" rts:sequence="1">
    <rts:WaitTime rts:waitTime="300"/>
    <rts:TargetComponent rts:id="Comp3Ec1" rts:instanceName="SampleComponent3_1"
      rts:componentId="RTC:SampleVendor:SampleCategory:SampleComponent3:0.5.0"/>
  </rts:targets>
  <rts:targets xsi:type="rtsExt:condition_ext" rts:sequence="2">
    <rts:Preceding rts:sendingTiming="SYNC" rts:timeout="100">
      <rts:PrecedingComponents rts:id="Comp3Ec1" rts:instanceName="SampleComponent3_1"
        rts:componentId="RTC:SampleVendor:SampleCategory:SampleComponent3:0.5.0"/>
    </rts:Preceding>
    <rts:TargetComponent rts:id="Comp2Ec1" rts:instanceName="SampleComponent2_1"
      rts:componentId="RTC:SampleVendor:SampleCategory:SampleComponent2:1.0.0"/>
  </rts:targets>
  <rts:targets xsi:type="rtsExt:condition_ext" rts:sequence="3">
    <rts:Preceding rts:sendingTiming="SYNC" rts:timeout="100">
      <rts:PrecedingComponents rts:id="Comp3Ec1" rts:instanceName="SampleComponent3_1"
        rts:componentId="RTC:SampleVendor:SampleCategory:SampleComponent3:0.5.0"/>
      <rts:PrecedingComponents rts:id="Comp2Ec1" rts:instanceName="SampleComponent2_1"
        rts:componentId="RTC:SampleVendor:SampleCategory:SampleComponent2:1.0.0"/>
    </rts:Preceding>
    <rts:TargetComponent rts:id="Comp1Ec1" rts:instanceName="SampleComponent_1"
      rts:componentId="RTC:SampleVendor:SampleCategory:SampleComponent:1.0.0"/>
  </rts:targets>
</rts:ShutDown>
</rts:RtsProfile>

```

7.3 YAML サンプル

```
---
rtsProfile:
  abstract: Sample Abstract
  "rtsExt::comment!": Sample RTSystem
  components:
    -
      activeConfigurationSet: configSet_1
      compositeType: None
      configurationSets:
        -
          id: configSet_1
          configurationData:
            -
              name: name1
              data: value
            -
              name: name2
              data: value
        -
          id: configSet_2
          configurationData:
            -
              name: name1
              data: value
      dataPorts:
        -
          name: Comp1_outport1
          "rtsExt::visible": true
        -
          name: Comp1_outport2
          "rtsExt::visible": false
        -
          name: Comp1_outport3
          "rtsExt::visible": true
      executionContexts:
        -
          id: Comp1Ec1
          kind: PeriodicExecutionContext
          rate: 1000
      id: RTC:SampleVendor:SampleCategory:SampleComponent:1.0.0
      instanceName: SampleComponent_1
      isRequired: true
      "rtsExt::location!":
        direction: RIGHT
        x: 93
        y: 93
        height: -1
        width: -1
      pathUri: file://localhost/C:¥RTM¥XMLSample¥XML¥SampleXML.xml
      servicePorts:
        -
          name: SrvPort2
          "rtsExt::visible": true
    -
      activeConfigurationSet: configSet_1
      compositeType: None
      configurationSets:
        -
          id: configSet_2
          configurationData:
            -
              name: name1
              data: value
```

```

dataPorts:
-
  name: Comp2_inort1
  "rtsExt::visible": true
executionContexts:
-
  id: Comp2Ec1
  kind: PeriodicExecutionContext
  rate: 500
id: RTC:SampleVendor:SampleCategory:SampleComponent2:1.0.0
instanceName: SampleComponent2_1
isRequired: true
"rtsExt::location":
  direction: RIGHT
  x: 93
  y: 276
  height: -1
  width: -1
pathUri: file://localhost/C:\RTM\XMLSample\XML\SampleXML2.xml
"rtsExt::visible": true
-
compositeType: None
dataPorts:
-
  name: Comp3_inort2
  "rtsExt::visible": true
executionContexts:
-
  id: Comp3Ec1
  kind: PeriodicExecutionContext
  rate: 300
id: RTC:SampleVendor:SampleCategory:SampleComponent3:0.5.0
instanceName: SampleComponent3_1
isRequired: true
"rtsExt::location":
  direction: LEFT
  x: 93
  y: 93
  height: -1
  width: -1
pathUri: file://localhost/C:\RTM\XMLSample\XML\SampleXML3.xml
"rtsExt::visible": true
servicePorts:
-
  name: SrvPort4
creationDate:
  day: 6
  hour: 13
  minute: 49
  month: 5
  second: 53
  year: 2008
dataPortConnectors:
-
  connectorId: 9c477198-dbf4-4298-9713-c5e1b1b30607
  dataflowType: PUSH
  dataType: RTC::TimedLong
  name: Comp1_outport1_Comp2_inport1
  interfaceType: CORBA_Any
  sourceDataPort:
    componentId: RTC:SampleVendor:SampleCategory:SampleComponent:1.0.0
    instanceName: SampleComponent_1
    portName: Comp1_outport1
  subscriptionType: Flush
  targetDataPort:
    componentId: RTC:SampleVendor:SampleCategory:SampleComponent2:1.0.0
    instanceName: SampleComponent2_1
    portName: Comp2_inort1

```

```

-
connectorId: c5e1b1b306-9713-4298-dbf4-9c47719807
dataflowType: PUSH
dataType: RTC::TimedDouble
name: Comp1_outport3_Comp3_inport2
interfaceType: CORBA_Any
subscriptionType: Flush
sourceDataPort:
  componentId: RTC:SampleVendor:SampleCategory:SampleComponent:1.0.0
  instanceName: SampleComponent_1
  portName: Comp1_outport3
targetDataPort:
  componentId: RTC:SampleVendor:SampleCategory:SampleComponent3:0.5.0
  instanceName: SampleComponent3_1
  portName: Comp3_inport2
id: RTSystem:jp.go.aist.SampleRTS:1.0.0
servicePortConnectors:
-
connectorId: 9fbb4aa9-cd3f-4723-9375-b8e1201d0b6f
name: SrvPort2_SrvPort4
sourceServicePort:
  componentId: RTC:SampleVendor:SampleCategory:SampleComponent:1.0.0
  instanceName: SampleComponent_1
  portName: SrvPort2
targetServicePort:
  componentId: RTC:SampleVendor:SampleCategory:SampleComponent3:0.5.0
  instanceName: SampleComponent3_1
  portName: SrvPort4
shutDown:
targets:
-
sequence: 1
targetComponent:
  componentId: RTC:SampleVendor:SampleCategory:SampleComponent3:0.5.0
  id: Comp3Ec1
  instanceName: SampleComponent3_1
condition:
  waitTime:
    waitTime: 300
-
sequence: 2
targetComponent:
  componentId: RTC:SampleVendor:SampleCategory:SampleComponent2:1.0.0
  id: Comp2Ec1
  instanceName: SampleComponent2_1
condition:
  preceding:
    timeout: 100
    sendingTiming: SYNC
    precedingComponents:
      -
        componentId: RTC:SampleVendor:SampleCategory:SampleComponent3:0.5.0
        id: Comp3Ec1
        instanceName: SampleComponent3_1
-
sequence: 3
targetComponent:
  componentId: RTC:SampleVendor:SampleCategory:SampleComponent:1.0.0
  id: Comp1Ec1
  instanceName: SampleComponent_1
condition:
  preceding:
    timeout: 100
    sendingTiming: SYNC
    precedingComponents:
      -
        componentId: RTC:SampleVendor:SampleCategory:SampleComponent3:0.5.0
        id: Comp3Ec1
        instanceName: SampleComponent3_1

```

```

-
  componentId: RTC:SampleVendor:SampleCategory:SampleComponent2:1.0.0
  id: Comp2Ec1
  instanceName: SampleComponent2_1
stratUp:
  targets:
  -
    sequence: 1
    targetComponent:
      componentId: RTC:SampleVendor:SampleCategory:SampleComponent:1.0.0
      id: Comp1Ec1
      instanceName: SampleComponent_1
    condition:
      waitTime:
        waitTime: 1000
  -
    sequence: 2
    targetComponent:
      componentId: RTC:SampleVendor:SampleCategory:SampleComponent2:1.0.0
      id: Comp2Ec1
      instanceName: SampleComponent2_1
    condition:
      waitTime:
        waitTime: 500
  -
    sequence: 3
    targetComponent:
      componentId: RTC:SampleVendor:SampleCategory:SampleComponent3:0.5.0
      id: Comp3Ec1
      instanceName: SampleComponent3_1
    condition:
      waitTime:
        waitTime: 750
  updateDate:
    day: 6
    hour: 13
    minute: 49
    month: 5
    second: 53
    year: 2008
  version: 0.2

```

8 Appendix: 仕様記述利用例

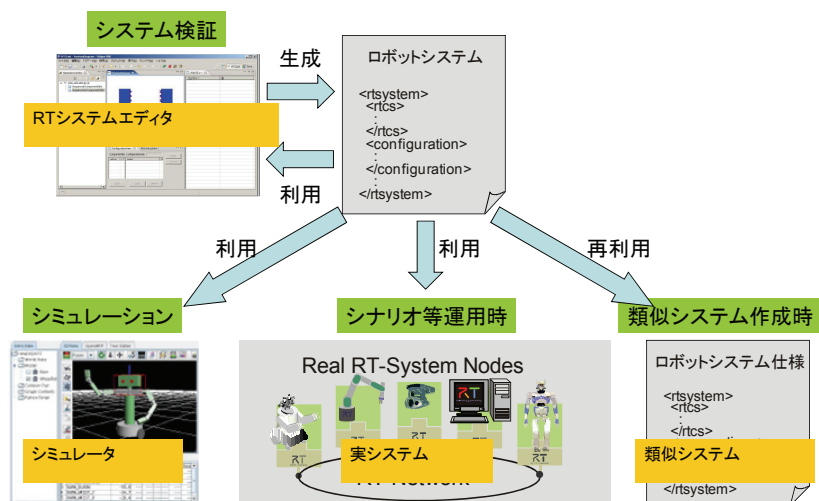
ここでは、RTS仕様記述方式で作成された仕様の実際の使用例を示す。

8.1 RTS 仕様記述の利用例

RTS仕様記述方式で記述されたRTS仕様は、RTシステム開発プラットフォームOpenRTPを構成するツールチェーンにおいて、ツール間のデータ交換のための標準フォーマットとして利用することを想定している。

OpenRTPでは仕様に基づくRTシステム開発環境を構築することを目的としており、RTS仕様記述はRTCで構成されるシステムを記述するためのデータフォーマットである。

図 1にRTS仕様ファイルを利用するツール群の相互関係を示す。



22

図 1 OpenRTP ツールチェーンにおける RTS 仕様の利用

8.2 システム構築・検証での利用

RTCで構成されるシステムにおいては、システム構成情報は、コンポーネントの配置情報・接続情報・コンフィギュレーション情報等から構成される。このシステム構成情報を記述するフォーマットとしてRTS Profileを利用する。

産総研が提供するRTS設計ツール「RTシステムエディタ」は、コンポーネントのプロファイル情報を読み込み、Eclipse上でシステムの構成を編集し、XML形式のRTS仕様ファイルを生成することができる（オフラインエディタ機能）。

また、生成されたRTS XMLファイルを読み込み、動作中のRTCの接続を再構成したり、コンフィギュレーション情報を設定、さらに動作を検証しつつ接続やコンフィギュレーション情報を再構築することができる（オンラインエディタ機能）。

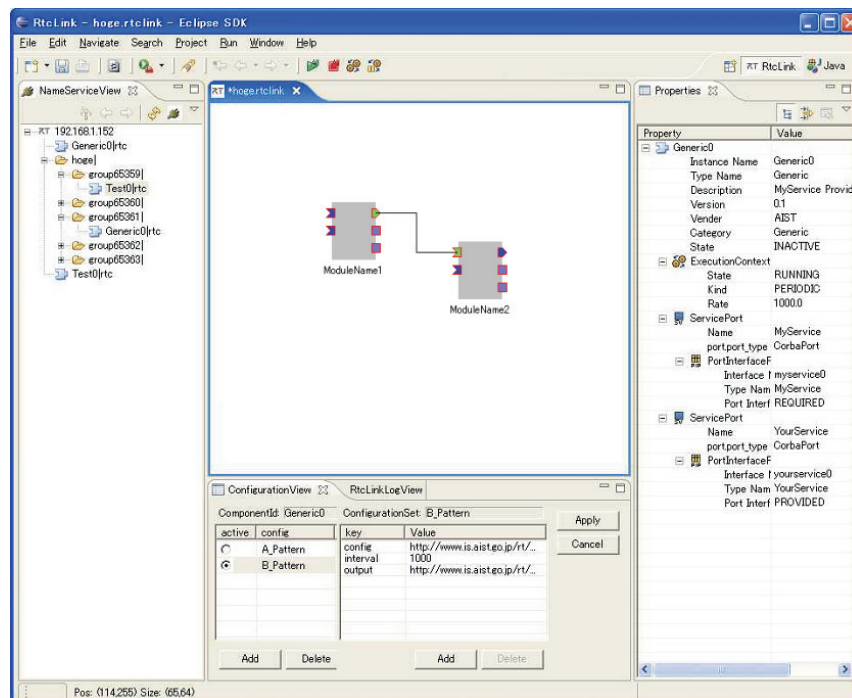


図 2 RT システムエディタ（オフライン編集画面）

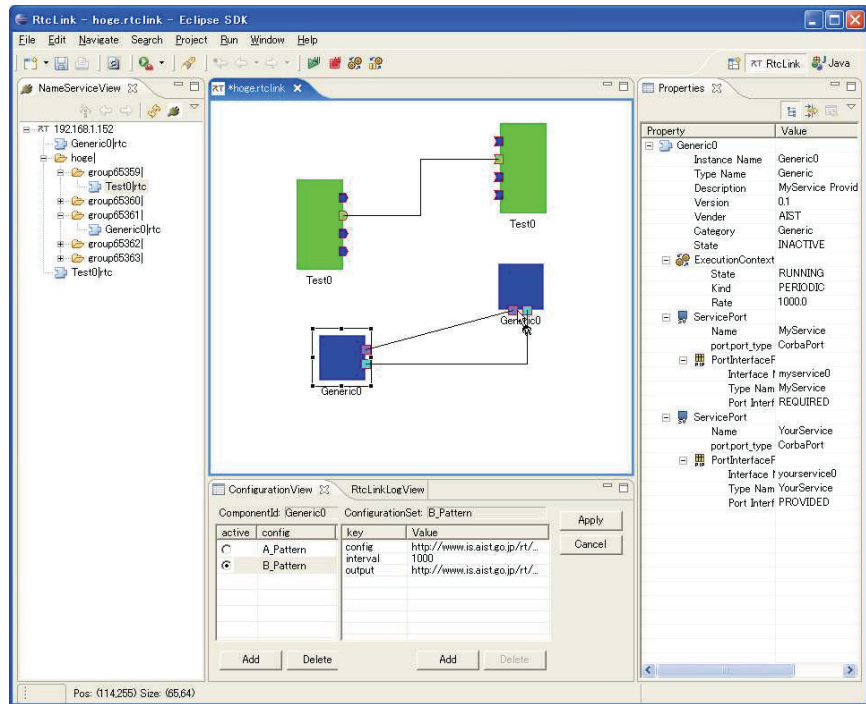


図 3 RT システムエディタ (オンライン編集画面)

8.3 システム運用時に利用

RTS仕様記述ファイルには、RTCから構成されるRTシステムの接続情報、コンフィギュレーション情報などが含まれている。したがって、システム運用時にはこのファイルを元にRTCの起動・接続・設定 (Deployment and Configuration: D&C) を行うために利用することができる。

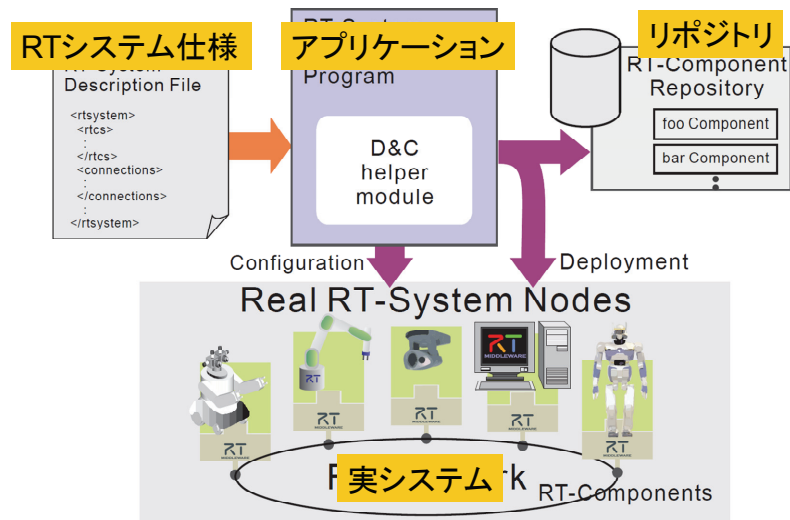


図 4 システム運用イメージ

図 4にシステム運用時のイメージを示す。RTS仕様ファイルを実アプリケーションが読み込み、仕様記述を元にアプリケーションに必要なRTCをリポ

ジトリからノードに配置するとともに、RTCの接続・設定を行い、システム全体を起動する。

システム構成を動的に変更する場合には、個別のRTCを操作して構成を変更することも可能であるが、予め用意された複数のRTS Profile XMLを切り替えることにより、迅速かつ容易にシステム構成を変更することができる。

また、シミュレータを利用する場合でも、実機の部分がシミュレータのコンポーネントに置き換わるだけで、基本的には同様の流れでシステムのD&Cが行える。

8.4 既存仕様の再利用

すでに存在するシステムと類似のシステムを作成する際には、既存のRTS仕様を参照することで、より再利用性の高いシステムを作成することができる。

9 参考文献

9.1 標準仕様

[CORBA] Common Object Request Broker Architecture (CORBA) Core Specification, version 3.0.3 Specification: formal/2004-03-12

http://www.omg.org/technology/documents/formal/corba_iiop.htm

[SDO] Platform Independent Model (PIM) & Platform Specific Model (PSM) for Super Distributed Objects(SDO),version 1.1

<http://www.omg.org/SDO/1.1/>

[UML] Unified Modeling Language, Superstructure Specification version 2.1.2

<http://www.omg.org/technology/documents/formal/uml.htm>

[RTC] Robotic Technology Component Specification Version 1.0

<http://www.omg.org/docs/formal/08-04-04.pdf>

[XML Schema] XML Schema W3C Architecture Domain,

<http://www.w3.org/XML/Schema>