

---

# RT コンポーネント仕様記述方式

---

2008.11.XX  
Version0.2



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

---

<b>1</b>	<b>概要</b> .....	<b>7</b>
1.1	目的.....	7
1.2	仕様概要.....	8
1.2.1	要求仕様.....	8
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.....	9
1.3	提案に関する連絡先.....	9
1.4	準拠性.....	10
1.5	既存の標準仕様との関連.....	10
1.5.1	Platform-Independent Model.....	10
1.5.2	Platform-Specific Models.....	10
<b>2</b>	<b>Platform-Independent Model</b> .....	<b>11</b>
2.1	書式および凡例.....	11
2.2	RTC プロファイル.....	13
2.3	RTC 基本プロファイル.....	16
2.3.1	rtc_profile.....	16
2.3.2	basic_info.....	17
2.3.3	component_type.....	21
2.3.4	activity_type.....	21
2.3.5	actions.....	22
2.3.6	action_status.....	23
2.3.7	configuration_set.....	24
2.3.8	configuration.....	24
2.3.9	dataPort.....	26
2.3.10	port_type.....	28
2.3.11	serviceport.....	29
2.3.12	transmission_method.....	30
2.3.13	serviceinterface.....	30
2.3.14	interface_direction.....	32
2.3.15	parameter.....	32
2.3.16	language.....	33
2.3.17	constraint_type.....	34

2.3.18	constraint_unit_type	35
2.3.19	constraint_list_type	35
2.3.20	constraint_hash_type	35
2.3.21	logic_ops_type	36
2.3.22	unary_logic_op_type	36
2.3.23	not	36
2.3.24	binary_logic_op_type	36
2.3.25	and	37
2.3.26	or	37
2.3.27	comparison_ops_type	37
2.3.28	property_is_null_type	37
2.3.29	property_is_null	38
2.3.30	binary_comparison_op_type	38
2.3.31	property_is_equal_to	38
2.3.32	property_is_not_equal_to	39
2.3.33	property_is_less_than	39
2.3.34	property_is_greater_than	39
2.3.35	property_is_less_than_or_equal_to	39
2.3.36	property_is_greater_than_or_equal_to	40
2.3.37	property_is_like_type	40
2.3.38	property_is_like	41
2.3.39	property_is_between_type	41
2.3.40	property_is_between	41
2.3.41	literal	41
2.4	RTC ドキュメントプロファイル	42
2.4.1	doc_basic	42
2.4.2	doc_action	44
2.4.3	doc_configuration	45
2.4.4	doc_dataport	47
2.4.5	doc_serviceport	49
2.4.6	doc_serviceinterface	50
2.5	RTC 拡張プロファイル	52
2.5.1	basic_info_ext	52
2.5.2	configuration_ext	53
2.5.3	dataport_ext	54
2.5.4	serviceport_ext	56
2.5.5	serviceinterface_ext	57
2.5.6	Position	57
2.5.7	language_ext	58
2.5.8	target_environment	58
2.5.9	library	59
2.5.10	property	59
3	プラットフォーム依存モデル	61
3.1	UML-XML 変換	61
3.1.1	基本型	61
3.1.2	パッケージと名前空間	62
3.1.3	クラス	62
3.2	UML-YAML 変換	65
3.2.1	基本型	65
3.2.2	パッケージと名前空間	65
3.2.3	クラス	65
4	RTC 表記法	67
4.1	RTC 構成図記述方式	67
4.2	RTC 構成図例	69

<b>5</b>	<b>Appendix: 各種定義</b> .....	<b>70</b>
5.1	ComponentKind.....	70
5.2	executionType.....	70
5.3	データ型.....	70
5.4	システムコンフィギュレーション.....	71
<b>6</b>	<b>Appendix: XML スキーマ</b> .....	<b>72</b>
6.1	RTC 基本 Profile(RtcProfile_basic.xsd).....	72
6.2	RTC ドキュメント Profile(RtcProfile_doc.xsd).....	77
6.3	RTC 拡張 Profile(RtcProfile_ext.xsd).....	79
<b>7</b>	<b>Appendix: YAML スキーマ</b> .....	<b>81</b>
<b>8</b>	<b>Appendix: 制約記述サンプル</b> .....	<b>86</b>
8.1	XML 86	
8.2	YAML 88	
<b>9</b>	<b>Appendix: サンプル</b> .....	<b>90</b>
9.1	コンポーネント仕様.....	90
9.2	RTC.xml.....	94
9.3	YAML 形式.....	97
<b>10</b>	<b>Appendix: 仕様記述利用例</b> .....	<b>100</b>
10.1	RTC 仕様記述の利用例.....	100
10.2	RTC 仕様記述フォーマットとしての利用.....	101
10.3	コード生成のための情報として利用.....	101
10.4	システム構成時に利用.....	102
10.5	RTC リポジトリにおける検索情報として利用.....	103
10.6	既存仕様の再利用.....	103
10.7	ドキュメントとして利用.....	104
<b>11</b>	<b>参考文献</b> .....	<b>105</b>
11.1	標準仕様.....	105

## 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、知能モジュール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サイトにて公開する。

<http://www.is.aist.go.jp/rt/OpenRTP/html>

### 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

This chapter contains the following sections.

Section Title	Page
目的	7
仕様概要	8
提案に関する連絡先	9
既存の標準仕様との関連	10

### 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、コンポーネントモデル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コンポーネント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 にお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コンポーネント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3つのパッケージから構成される：

1. **RTC Basic Profile.** RTCの基本的なコンポーネントモデルを記述する際に必要な情報を定義する。RTCの基本プロファイルおよびポートやコンフィギュレーション情報がこれに含まれる。
2. **RTC Extended Profile.** RTC 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。
3. **RTC Documentation Profile.** 上記プロファイル情報以外に、詳細な仕様を文書として記述するためのデータ構造を定義する。



以下に、上記3つのパッケージの関係を示す。

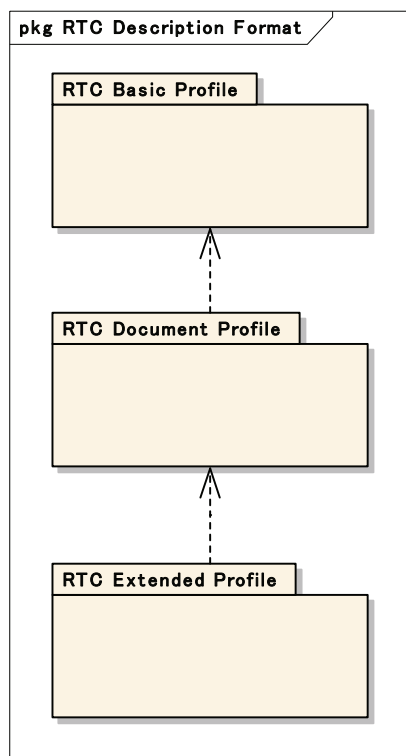


Figure 1 RTC Description Format

### 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.** RTC仕様記述方式をXMLで記述するためのXMLスキーマ。

**YAML.** RTC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](mailto: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

This chapter contains the following sections.

Section Title	Page
書式および凡例	11
RTCプロファイル	13
RTC基本プロファイル	16
RTCドキュメントプロファイル	42
RTC拡張プロファイル	52

### 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。

<b><i>MyClass</i></b>		
attributes		
my_attribute	0..1	long
relationships		
my_relational_attribute1	0..*	SomeClass
my_relational_attribute2	1	string

## 2.2 RTC プロファイル

RTCプロファイルは、ロボット用コンポーネントの各種属性を定義するデータモデルである。RTCプロファイルのPIMの全体図を図2に示す。また、プロパティ情報に関するクラスの部分を図3に、各種制約情報を表現するための要素に関するクラス図を図4にそれぞれ示す。

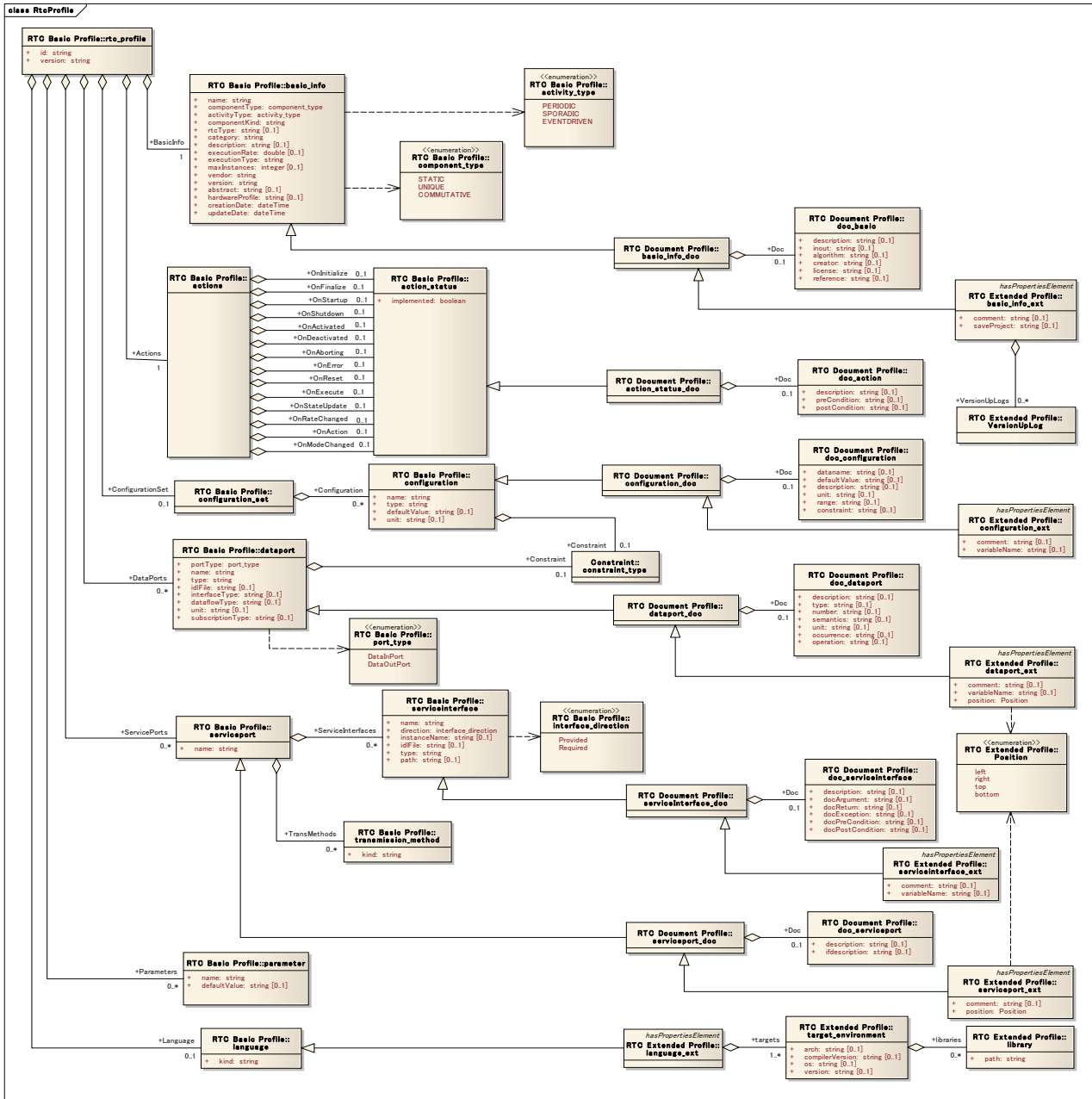


Figure 2 RTC プロファイル(全体図)

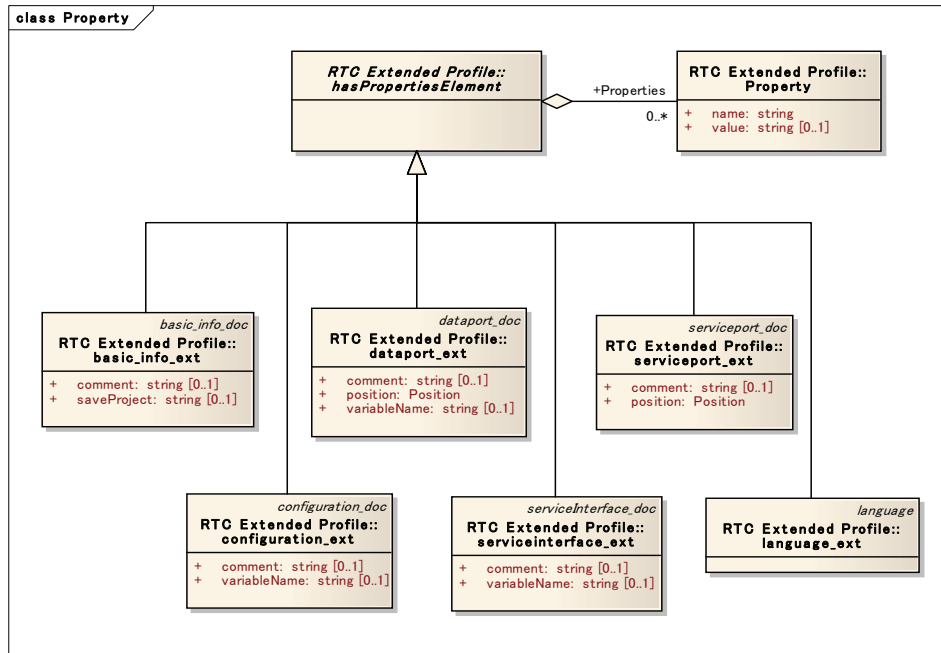


Figure 3 RTC プロファイル(プロパティ部分)

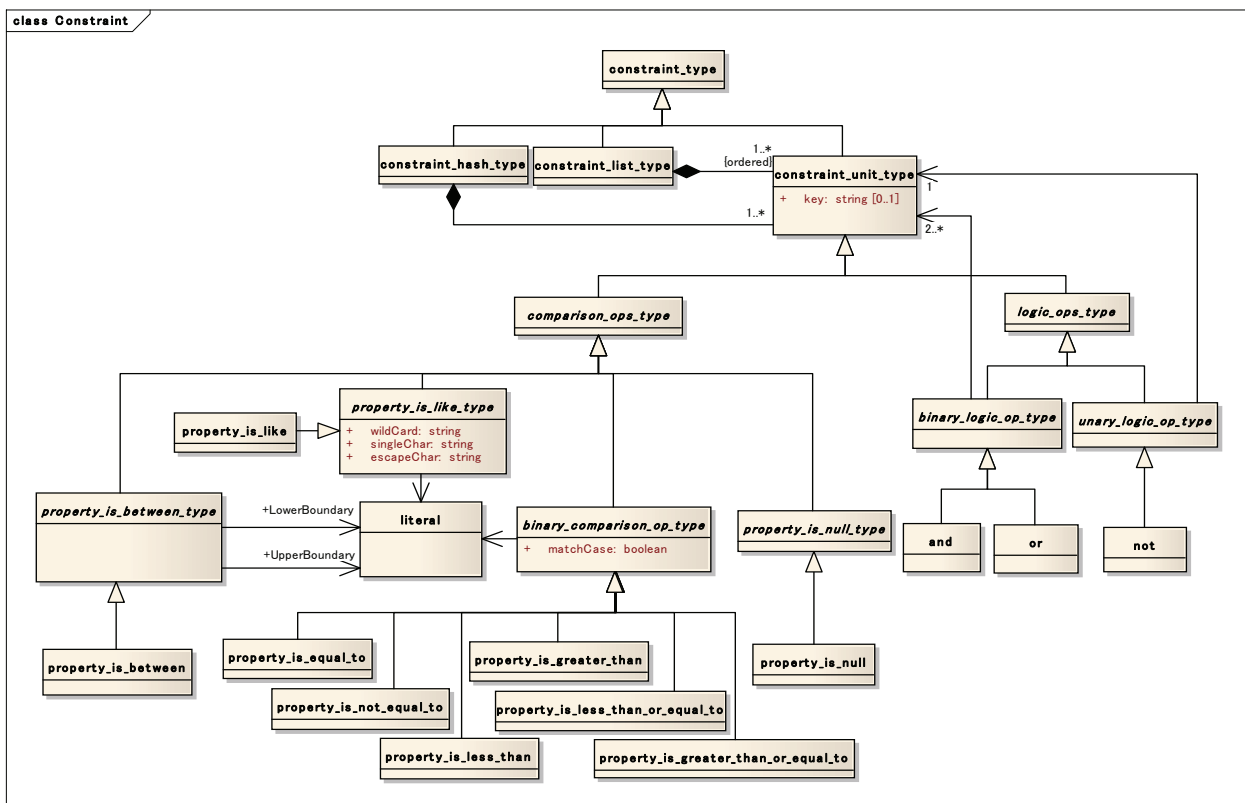


Figure 4 RTC プロファイル(制約部分)

**RTC基本プロフィール (RTC Basic Profile)**

RTC基本プロフィールは、RTCの基本メタ情報を含む。

**RTC拡張プロフィール (RTC Extended Profile)**

RTC拡張プロフィールは、RTC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ィールである。

**RTCドキュメントプロフィール (RTC Document Profile)**

RTCドキュメントプロフィールは、RTCの詳細な仕様を文書として記述するために用意されているプロフィールである。

## 2.3 RTC 基本プロフィール

RTCプロフィール(PIM)のコンポーネント部分の拡大図を図5に示す。

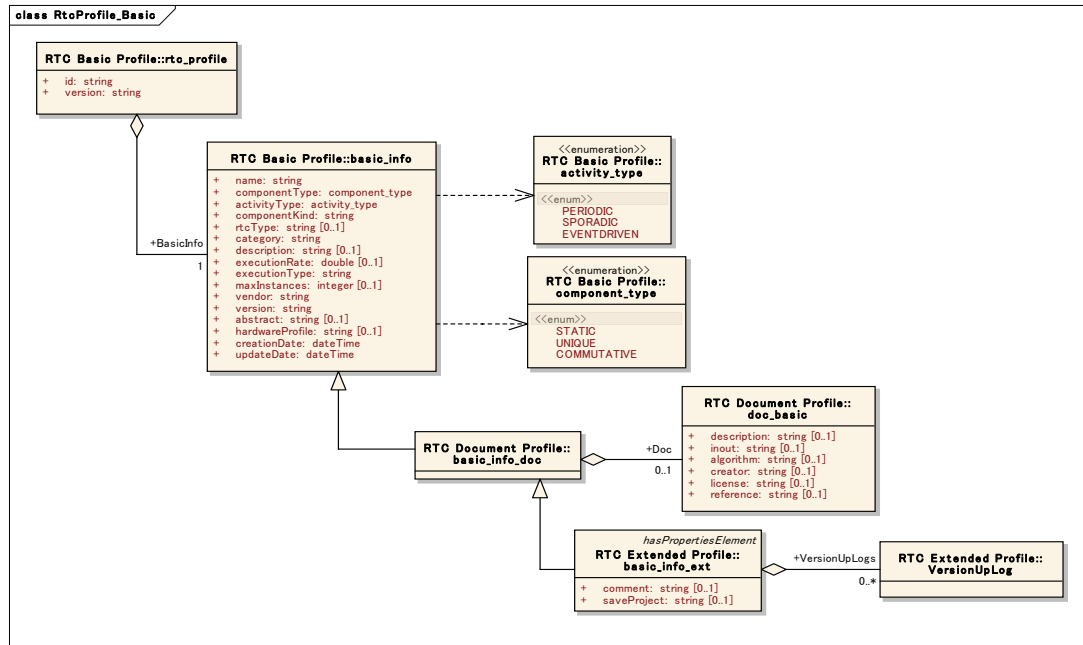


Figure 5 RTC プロフィール(コンポーネント部分)

### 2.3.1 rtc\_profile

#### 概要

rtc\_profileは、RTC仕様記述のルート要素である。また、仕様記述対象のRTC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。

#### 意味

RTC仕様記述の各種詳細情報は、rtc\_profile要素以下の各要素に記述される。

#### 属性

<i>rtc_profile</i>		
attributes		
id	1	string
version	1	string
relationships		
BasicInfo	1	basic_info
Actions	1	actions
ConfigurationSet	0..1	configuration_set
DataPorts	0..*	dataport
ServicePorts	0..*	serviceport
Parameters	0..*	parameter
Language	0..1	language



### 2.3.1.1 id

#### 概要

仕様記述対象のRTCを一意に識別するための識別子を指定する。

#### 意味

RTCの識別子を指定する。識別子の書式は以下の構成とする。

RTC : [ベンダ名]. [カテゴリ名]. [コンポーネント名] : [バージョン番号]

### 2.3.1.2 version

#### 概要

RTC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RTC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 basic\_info

#### 概要

basic\_infoはRTCの基本情報を記述する要素である。

#### 意味

basic\_infoはRTCが共通に保持する基本情報を記述する。

#### 属性

<b>basic_info</b>		
attributes		
name	1	string
componentType	1	component_type
activityType	1	activity_type
componentKind	1	string
rtcType	0..1	string
category	1	string
description	0..1	string
executionRate	0..1	double
executionType	1	string
maxInstances	0..1	integer
vendor	1	string
version	1	string
abstract	0..1	string
hardwareProfile	0..1	string
creationDate	1	dateTime
updateDate	1	dateTime
no relationships		

### 2.3.2.1 *name*

#### 概要

この属性はRTCの名称を指定する。

#### 意味

RTCの機能を端的に表す名称を指定する。この属性はRTCのIDの一部としても使用される。必須属性。

### 2.3.2.2 *componentType*

#### 概要

RTCの型を指定する。

#### 意味

コンポーネント自身の型を指定する。コンポーネントの型は、`component_type`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。

### 2.3.2.3 *activityType*

#### 概要

RTCのアクティビティのタイプを指定する。

#### 意味

RTCのアクティビティのタイプを指定する。アクティビティタイプは、`activity_type`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。

### 2.3.2.4 *componentKind*

#### 概要

RTCの実行形態の種類を指定する。

#### 意味

RTCの実行形態の種類を指定する。RTCの種類はAppendixに指定されている種類から選択可能である。必須属性。

### 2.3.2.5 *rtcType*

#### 概要

RTCの種類を指定する。

#### 意味

RTCの種類を指定する。ある特定の機能を実現するRTCなどを作成し、RTCの種類を区別する必要がある場合に利用する。値が省略された場合には通常のRTCとして解釈される。省略可能属性。

### 2.3.2.6 *category*

#### 概要

RTCのカテゴリを指定する。

#### 意味

RTCが属するカテゴリを指定する。RTCの設計者が自由に選択し記述することができる。この属性はRTCのIDの一部としても使用される。必須属性。

### 2.3.2.7 *description*

#### 概要

RTCの概要情報を記述する。

#### 意味

当該RTCの概要について説明を記述する。この属性値はコンポーネントの使用者の利便性のためだけに設けられ、ソフトウェア的に処理されることは意図しない。省略可能属性。

### 2.3.2.8 *executionRate*

#### 概要

RTCのアクションの実行周期を指定する。

#### 意味

コンポーネントアクションの実行周期を[Hz]で記述する。この属性値はデータフロー型のコンポーネントのみにおいて有効であり、それ以外の型(イベントドリブン型など)のコンポーネントにおいては無視される。省略可能属性。

### 2.3.2.9 *executionType*

#### 概要

RTCのアクションの実行タイプを指定する。

#### 意味

コンポーネント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。

### 2.3.2.10 *maxInstance*

#### 概要

RTCの最大のインスタンス数を指定する。

#### 意味

存在可能なRTCのインスタンスの最大数を指定する。負数もしくは空文字が指定されている場合は、「制限なし」と解釈する。省略可能属性。

### 2.3.2.11 *vendor*

#### 概要

RTCの作成ベンダ名を指定する。

#### 意味

ベンダ名はドメイン名をトップレベルドメイン側から並べ替えた名称を記述する。ドメイン名がない場合は可能な限り衝突しない独自名称を記述することもできる。ただし、ベンダ名には空白文字、「.」を含んではならない。この属性はRTCのIDの一部としても使用される。必須属性。

### 2.3.2.12 *version*

#### 概要

RTCのバージョン番号を指定する。

#### 意味

バージョン番号は、原則「x. y. z」の書式にて記述する。この属性はRTCのIDの一部としても使用される。必須属性。

### 2.3.2.13 *abstract*

#### 概要

RTCに関する説明を指定する。

#### 意味

この属性値はコンポーネント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。

### 2.3.2.14 *hardwareProfile*

#### 概要

RTCが利用するハードウェアのプロファイル情報を指定する。

#### 意味

この属性値は、各種センサやアクチュエータなど、RTCがハードウェアを利用している場合に、対象ハードウェアプロファイルのプロファイルを指定するために設けられている。省略可能属性。

### 2.3.2.15 *creationDate*

#### 概要

RTC仕様の作成日時を指定する。

#### 意味

この属性値は当該RTC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。

### 2.3.2.16 *updateDate*

#### 概要

RTC仕様の最終更新日時を指定する。

#### 意味

この属性値は当該RTC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。

### 2.3.3 *component\_type*

*component\_type*は、RTCの型を指定するための列挙型である。

属性

<i>component_type</i>	
attributes	
STATIC	<i>component_type</i>
UNIQUE	<i>component_type</i>
COMMUTATIVE	<i>component_type</i>

#### 2.3.3.1 *STATIC*

概要

静的に存在するコンポーネントを表す型。

意味

コンポーネントはモジュールの初期化と同時に生成され、以後動的に生成されたり削除されたりはしない。ハードウェアと深く結びつき、コンポーネントの数が物理的なデバイスの数に制限される場合など、動的なコンポーネントの生成・削除が意味を成さない場合に指定する。

#### 2.3.3.2 *UNIQUE*

概要

動的に生成・削除が可能であるが、各コンポーネントが内部に固有の状態を持っており、必ずしも交換可能ではないコンポーネントを表す型。

#### 2.3.3.3 *COMMUTATIVE*

概要

動的に生成・削除が可能であるとともに、内部に固有の状態を持たないため、生成されたコンポーネントが交換可能であるコンポーネントを表す型。

### 2.3.4 *activity\_type*

*activity\_type*は、RTCのアクティビティ型を指定するための列挙型である。

属性

<i>activity_type</i>	
attributes	
PERIODIC	<i>activity_type</i>
SPORADIC	<i>activity_type</i>
EVENTDRIVEN	<i>activity_type</i>

#### 2.3.4.1 *PERIODIC*

概要

一定周期でRTCのアクションを実行するアクティビティ型。

### 2.3.4.2 SPORADIC

#### 概要

RTCのアクションを不定期に実行するアクティビティ型。

### 2.3.4.3 EVENTDRIVEN

#### 概要

RTCのアクションがイベントドリブンであるアクティビティ型。

### 2.3.5 actions

#### 概要

RTCに定義された各種アクション(on\_initialize, on\_finalize, on\_startup, on\_shutdown, on\_activated, on\_deactivated, on\_aborting, on\_error, on\_reset, on\_execute, on\_state\_update, on\_rate\_change, on\_action, on\_mode\_changed)に関する情報を保持する要素である。

RTCプロファイル(PIM)のアクション部分の拡大図を図6に示す。

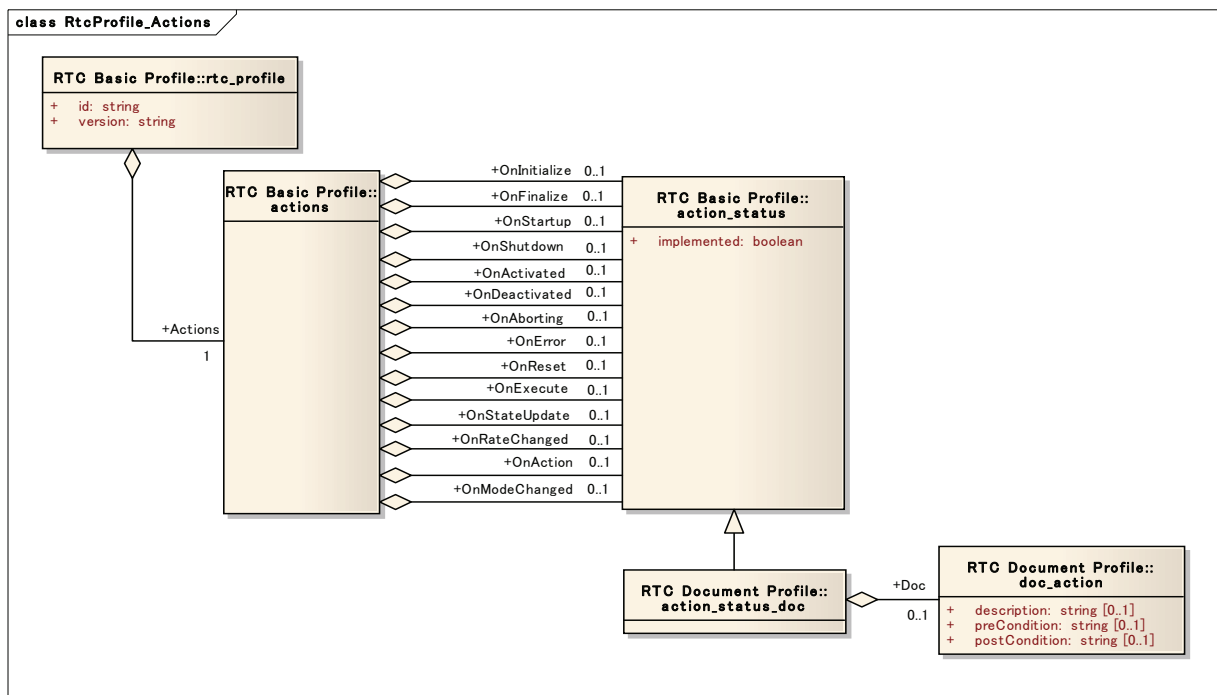


Figure 6 RTC プロファイル(アクション部分)

### 属性

<b>actions</b>		
no attributes		
relationships		
OnInitialize	0..1	action_status
OnFinalize	0..1	action_status
OnStartup	0..1	action_status
OnShutdown	0..1	action_status
OnActivated	0..1	action_status
OnDeactivated	0..1	action_status
OnAborting	0..1	action_status
OnError	0..1	action_status
OnReset	0..1	action_status
OnExecute	0..1	action_status
OnStateUpdate	0..1	action_status
OnRateChanged	0..1	action_status
OnAction	0..1	action_status
OnModeChanged	0..1	action_status

## 2.3.6 *action\_status*

### 概要

個々のアクションに関する情報を記述する。

### 属性

<b>action_status</b>		
attributes		
implemented	1	boolean
no relationships		

### 2.3.6.1 *implemented*

### 概要

該当するアクションが実装されているかを指定する。必須属性。

### 2.3.7 configuration\_set

#### 概要

RTC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の集合である。

RTCプロファイル(PIM)のコンフィギュレーション部分の拡大図を図7に示す。

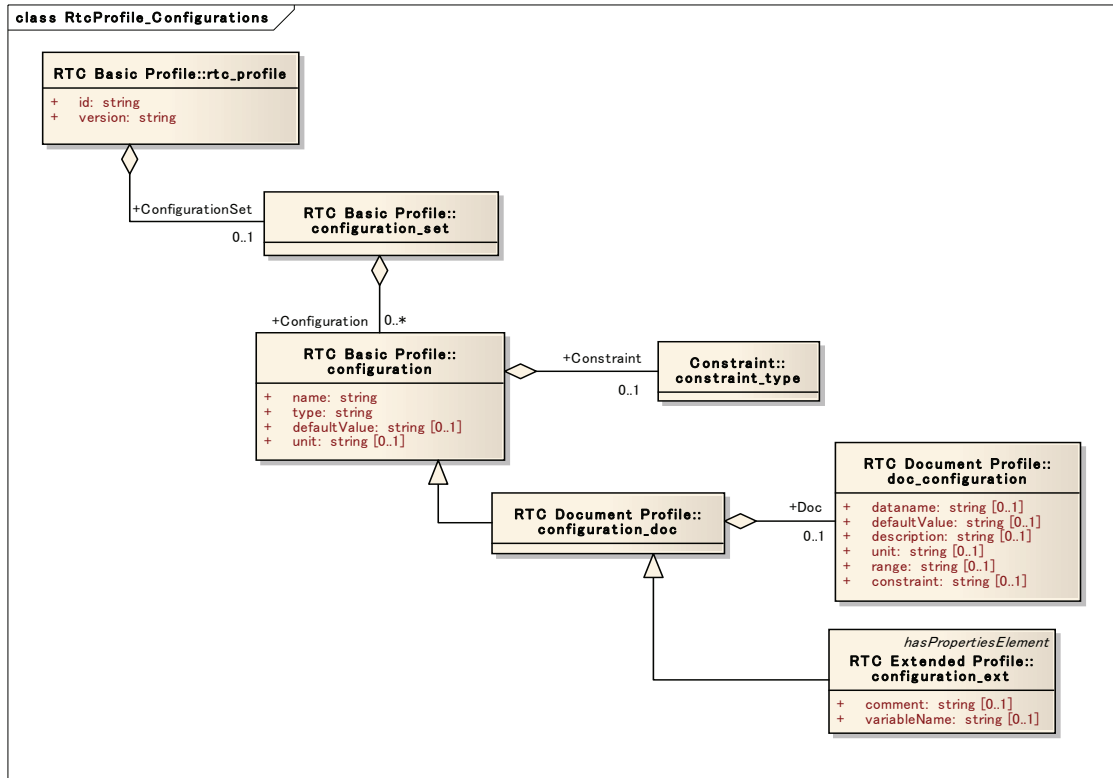


Figure 7 RTC プロファイル(コンフィギュレーション部分)

#### 属性

<i>configuration_set</i>		
no attributes		
relationships		
Configuration	0..*	configuration

### 2.3.8 configuration

configurationはRTC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。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</i>		
attributes		
name	1	string
type	1	string
defaultValue	0..1	string
unit	0..1	string
relationships		
constraint	0..1	constraint_type

#### 2.3.8.1 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

RTCのコンフィギュレーションパラメータ名を記述する。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。パラメータの内容を端的に表す名称とし、空白文字を含んではならない。必須属性。

#### 2.3.8.2 type

##### 概要

当該パラメータの型名を指定する。

##### 意味

当該パラメータの型名を指定する。型名にはAppendixで列挙される型などが使用可能である。型名と実装時の型とのマッピングはPSMにおいて指定される。必須属性。

#### 2.3.8.3 defaultValue

##### 概要

当該パラメータのデフォルト値を記述する。

##### 意味

当該パラメータにデフォルト値が指定されている場合に記述する。当該パラメータに値が設定されていない場合、または不当な値が設定される場合はこのデフォルト値を使用することができる。省略可能属性。

#### 2.3.8.4 unit

##### 概要

当該パラメータの単位を記述する。

##### 意味

この属性値は、当該パラメータの単位を指定する。省略可能属性。

### 2.3.9 dataPort

RTCはコンポーネント間でデータを交換するためのポート：データポートを持つ。データポートにはデータを出力するDataOutPortおよびデータを入力するDataInPortがある。DataPortはRTCのデータポートに関するメタ情報を定義する。

RTCプロファイル(PIM)のデータポート部分の拡大図を図8に示す。

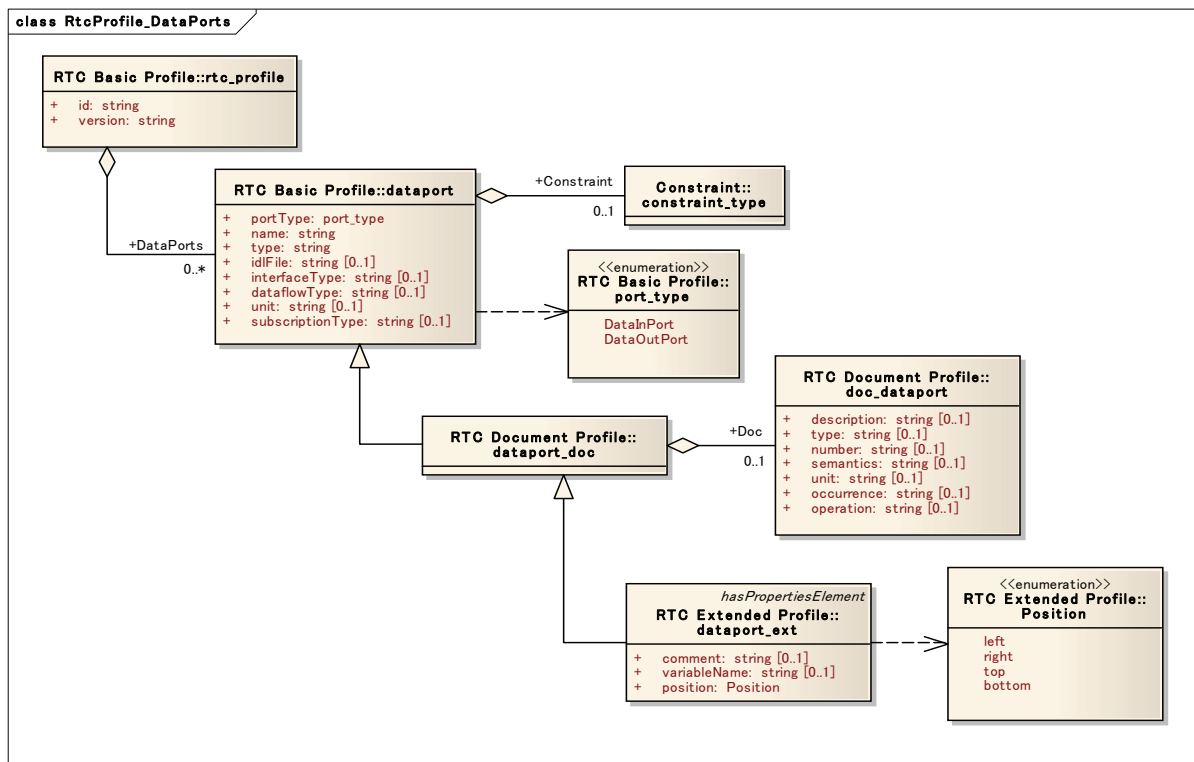


Figure 8 RTC プロファイル(データポート部分)

#### 属性

<i>dataPort</i>		
attributes		
portType	1	port_type
name	1	string
type	1	string
idlFile	0..1	string
interfaceType	0..1	string
dataflowType	0..1	string
subscriptionType	0..1	string
unit	0..1	string
relationships		
constraint	0..1	constraint_type

### 2.3.9.1 *portType*

#### 概要

DataPortの型を指定する。

#### 意味

DataPortの型としては、`port_type`列挙型で定義されている中から選択可能。必須属性。

### 2.3.9.2 *name*

#### 概要

DataPortの名称を指定する。

#### 意味

当該DataPortの名称を指定する。名称は同一RTC内でユニークでなくてはならない。この名称は外部からデータポートを参照する際のキーとして使用される。パラメータの内容を端的に表す名称とし、空白文字を含んではならない。必須属性。

### 2.3.9.3 *type*

#### 概要

DataPortの型名を指定する。

#### 意味

当該DataPortの型を指定する。型名にはAppendixで列挙される基本型を使用することが推奨されるが、RTCの設計者が独自の型を新たに定義し、使用することも可能である。必須属性。

### 2.3.9.4 *idlFile*

#### 概要

DataPortで使用するidlFileを指定する。

#### 意味

当該DataPortで使用する型がユーザ定義型の場合に、その内容を定義しているIDLファイルを指定する。省略可能属性。

### 2.3.9.5 *interfaceType*

#### 概要

DataPortがサポートするinterfaceTypeを指定する。

#### 意味

開発者が設計時に想定している通信用interfaceTypeを指定する。実際の動作時には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。

### 2.3.9.6 *dataflowType*

#### 概要

DataPortがサポートするdataflowTypeを指定する。

#### 意味

開発者が設計時に想定しているdataflowTypeを指定する。データ送受信のタイミングをInPort側で制御するか、OutPort側で制御するかを選択する。実際の動作時には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。

### 2.3.9.7 subscriptionType

#### 概要

DataPortがサポートするsubscriptionTypeを指定する。

#### 意味

開発者が設計時に想定しているsubscriptionTypeを指定する。OutPortからInPortにデータを送信するタイミングを制御するパラメータである。

本属性はdataflowTypeでPUSH型通信を選択したときのみ有効となる。

実際の動作時には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。

### 2.3.9.8 unit

#### 概要

DataPortが扱うデータの単位を記述する。

#### 意味

当該DataPortにてやり取りを行うデータの単位を指定する。省略可能属性。

### 2.3.10 port\_type

port\_typeは、DataPortの型を指定するための列挙型である。

#### 属性

<i>port_type</i>	
attributes	
DataInPort	port_type
DataOutPort	port_type

#### 2.3.10.1 DataInPort

#### 概要

DataInPortを表す型。

#### 意味

他のRTコンポーネントからの出力を受け取り、ハンドリングするポート。

#### 2.3.10.2 DataOutPort

#### 概要

DataOutPortを表す型。

意味

他のRTコンポーネントへ処理結果のデータストリームを渡すポート。

### 2.3.11 serviceport

RTCはコンポーネント間で相互作用をおこなうためのポート：サービスポートを持つ。serviceportではサービスポートに関するメタ情報を定義する。

RTCプロファイル(PIM)のサービスポート部分の拡大図を図9に示す。

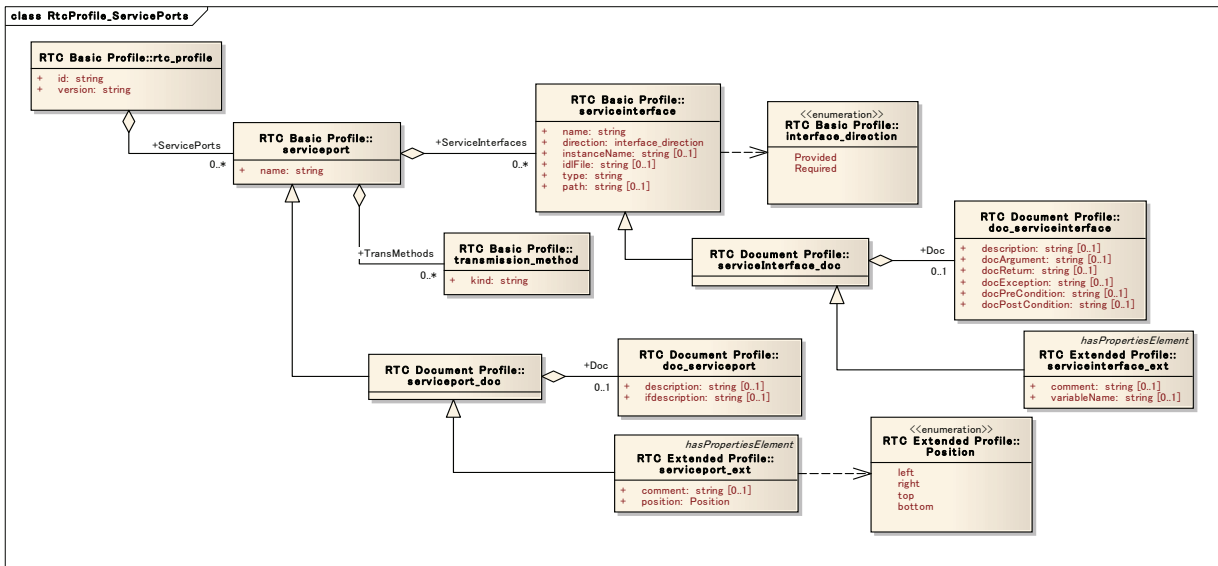


Figure9 RTC プロファイル(サービスポート部分)

属性

<i>serviceport</i>		
attributes		
name	1	string
relationships		
TransMethod	0..*	transmission_method
ServiceInterface	0..*	serviceinterface

#### 2.3.11.1 name

概要

ServicePortの名称を指定する。

意味

当該ServicePortの名称を指定する。名称は同一RTC内でユニークでなくてはならない。この名称は外部からサービスポートを参照する際のキーとして使用される。パラメータの内容を端的に表す名称とし、空白文字を含んではならない。必須属性。

### 2.3.12 *transmission\_method*

ServicePortが利用する通信方式。RTCでは多様な通信用ミドルウェアへ対応するために、通信部分の抽象化層を導入している。このため、通常のオブジェクトと通信ミドルウェアへのインターフェースマッピングを行うとともに、対象通信ミドルウェア用のアダプタオブジェクトを実装するだけで様々な通信方式への対応が可能となる。transmission\_methodでは通信方式に関するメタ情報を定義する。

#### 属性

<i>transmission_method</i>		
attributes		
kind	1	string
no relationships		

#### 2.3.12.1 *kind*

##### 概要

選択可能な通信方式の名称を指定する。

##### 意味

利用可能な通信方式を指定する。サービスポートで利用可能な通信方式の種類(CORBA、ICE、SOAPなど)を設定する。省略可能属性。

### 2.3.13 *serviceinterface*

サービスポートは複数のインターフェースを関連付けることができる。serviceinterfaceではサービスインターフェースに関するメタ情報を定義する。

#### 属性

<i>serviceinterface</i>		
attributes		
name	1	string
type	1	string
direction	1	interface_direction
instanceName	0..1	string
idlFile	0..1	string
path	0..1	string
no relationships		

#### 2.3.13.1 *name*

##### 概要

ServiceInterfaceの名称を指定する。

##### 意味

ServiceInterfaceの名称を使用する。名称は同一ポート内でユニークでなく

てはならない。この名称は外部からサービスインターフェースを参照する際のキーとして使用される。インターフェースの内容を端的に表す名称とし、空白文字を含んではならない。必須属性。

### 2.3.13.2 *type*

#### 概要

ServiceInterfaceの型名を指定する。

#### 意味

当該ServiceInterfaceが提供しているサービスの型名を指定する。必須属性。

### 2.3.13.3 *instanceName*

#### 概要

ServiceInterfaceのインスタンス名を指定する。

#### 意味

ServiceInterfaceのインスタンス名を指定する。省略可能属性。

### 2.3.13.4 *direction*

#### 概要

ServiceInterfaceの方向を指定する。

#### 意味

ServiceInterfaceの方向は、interface\_direction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。

### 2.3.13.5 *idlFile*

#### 概要

ServiceInterfaceの定義ファイルを指定する。

#### 意味

当該ServiceInterfaceが提供しているサービス内容を定義したファイルを指定する。省略可能属性。

### 2.3.13.6 *path*

#### 概要

ServiceInterfaceの定義ファイルへのパスを指定する。

#### 意味

当該ServiceInterfaceが提供しているサービス内容を定義したファイルへのパスを指定する。省略可能属性。

### 2.3.14 interface\_direction

interface\_directionは、ServiceInterfaceの型を指定するための列挙型である。

属性

<i>interface_direction</i>	
attributes	
Provided	interface_direction
Required	interface_direction

#### 2.3.14.1 Provided

概要

Provided(提供)インターフェースを表す型。

意味

Provided(提供)インターフェースは、他のRTコンポーネントへサービスを提供するためのインターフェースである。

#### 2.3.14.2 Required

概要

Required(要求)インターフェースを表す型。

意味

Required(要求)インターフェースは、他のRTコンポーネントが提供するサービスを利用するためのインターフェースである。

### 2.3.15 parameter

概要

parameterはRTCを実行する際に必要なシステムコンフィギュレーションに関するメタ情報を記述する。

RTCプロファイル(PIM)のパラメータおよび言語部分の拡大図を図10に示す。

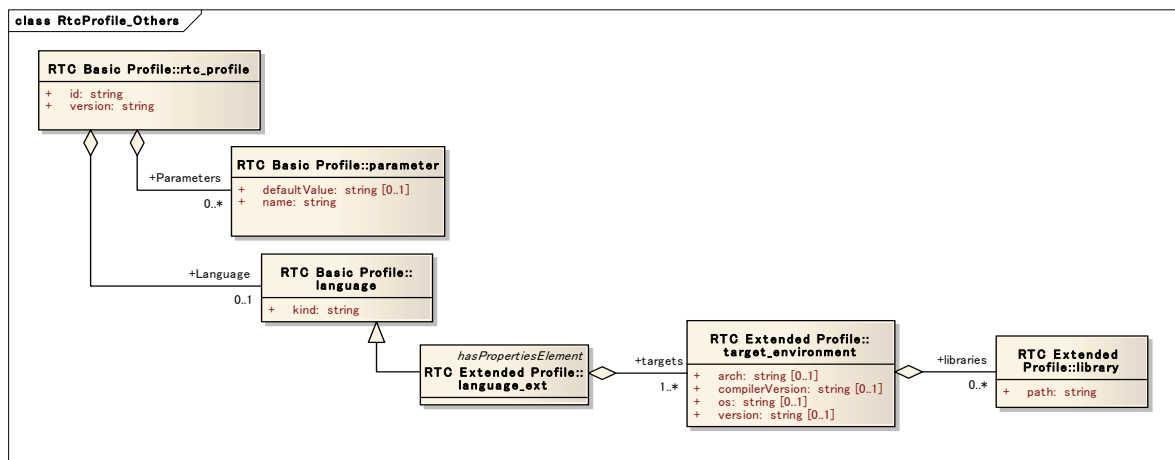


Figure10 RTC プロファイル(パラメータ, 言語部分)



#### 属性

<i>parameter</i>		
attributes		
name	1	string
defaultValue	0..1	string
no relationships		

#### 2.3.15.1 name

##### 概要

システムコンフィギュレーションの名称を指定する。

##### 意味

設定するシステムコンフィギュレーションの名称を指定する。Appendixで列挙されたシステムコンフィギュレーションが設定可能である。必須属性。

#### 2.3.15.2 defaultValue

##### 概要

システムコンフィギュレーションの設定値を指定する。

##### 意味

指定したシステムコンフィギュレーションの設定値を指定する。指定した値はあくまでもデフォルト値であり、RTコンポーネント実行時に変更される可能性もある。省略可能属性。

### 2.3.16 language

languageは対象RTCの実装言語に関するメタ情報を記述する。

#### 属性

<i>language</i>		
attributes		
kind	1	string
no relationships		

#### 2.3.16.1 kind

##### 概要

RTCの実装言語を指定する。

##### 意味

本属性は、リポジトリ内での検索キー、Manager内でのロード可能判断などに利用する。本属性で指定する実装言語名は以下のルールに従う形で記述する。

- ・アルファベット表記、ローマ字表記を用いて記述する。
- ・基本的には全て小文字で記述する。ただし、最初の1文字のみについて

は、大文字で記述してもよい。

- ・アルファベット以外の記号(+、#など)を用いても良い。

→「+」記号については、アルファベットの「X」にて代用してもよい。

以下に幾つかの実装言語の記述例を示す。

言語	記述例
C++	C++, c++, Cxx, cxx
C#	C#, csharp, Csharp
C	C, c
Python	Python, python
Ruby	Ruby, ruby
Perl	Perl, perl
Java	Java, java

必須属性。

### 2.3.17 *constraint\_type*

#### 概要

*constraint\_type*は各種パラメータの制約情報を記述するためのメタ情報のルート要素である。

RTCプロファイル(PIM)の制約部分の拡大図を図 1 1 に示す。

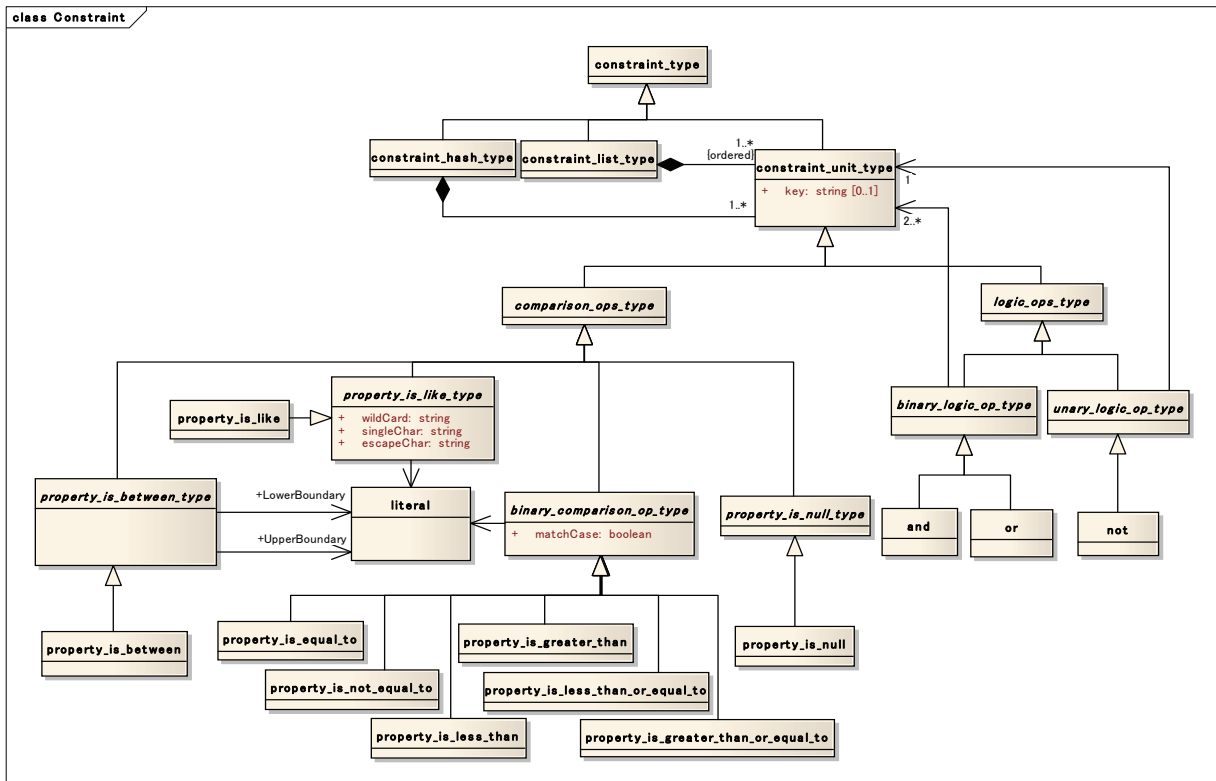


Figure11 RTC プロファイル(制約部分)

属性など

<i>constraint_type</i>	
no attributes	
no relationships	

### 2.3.18 *constraint\_unit\_type*

*constraint\_unit\_type*は、単一要素に対する制約を記述するための要素である。配列に対する制約(*constraint\_list\_type*)、ハッシュに対する制約(*constraint\_hash\_type*)の構成要素となる。

属性など

<i>logic_ops_type</i>		
Derived from	<i>constraint_type</i>	
attributes		
key	0..1	string
no relationships		

#### 2.3.18.1 *key*

概要

制約に対するキー情報を指定する。

意味

制約の記述対象がハッシュの場合に、ハッシュのキーを指定する。記述対象がハッシュ以外の場合には使用しない。省略可能属性。

### 2.3.19 *constraint\_list\_type*

*constraint\_list\_type*は、配列に対して制約を記述するための要素である。子要素として、各値に対する制約情報を持つ。

属性など

<i>constraint_list_type</i>		
Derived from	<i>constraint_type</i>	
no attributes		
relationships		
child	1..* {ordered}	<i>constraint_unit_type</i>

### 2.3.20 *constraint\_hash\_type*

*constraint\_hash\_type*は、ハッシュに対して制約を記述するための要素である。子要素として、各値に対する制約情報を持つ。

属性など

<b><i>constraint_hash_type</i></b>		
Derived from	constraint_type	
no attributes		
relationships		
child	1..*	constraint_unit_type

### 2.3.21 *logic\_ops\_type*

*logic\_ops\_type*は、論理演算子を記述するための抽象要素である。

属性など

<b><i>logic_ops_type</i></b>	
Derived from	constraint_type
no attributes	
no relationships	

### 2.3.22 *unary\_logic\_op\_type*

*unary\_logic\_op\_type*は、単項論理演算子を記述するための抽象要素である。子要素として制約情報を記述するためのルート要素である*constraint\_type*を持つことが可能であり、複合演算にも使用可能である。

属性など

<b><i>unary_logic_op_type</i></b>		
Derived from	logic_ops_type	
no attributes		
relationships		
child	1	Constraint_type

### 2.3.23 *not*

*not*は論理値を反転させる要素である。下位要素として設定された内容の論理値を反転させた結果を制約として設定する場合に使用する。

属性など

<b><i>not</i></b>	
Derived from	unary_logic_op_type
no attributes	
no relationships	

### 2.3.24 *binary\_logic\_op\_type*

*binary\_logic\_op\_type*は、二項論理演算子を記述するための抽象要素である。子要素として制約情報を記述するためのルート要素である*constraint\_type*を持つことが可能であり、複合演算にも使用可能である。

属性など

<b><i>binary_logic_op_type</i></b>		
Derived from	logic_ops_type	
no attributes		
relationships		
child	2..*	constraint_type

### 2.3.25 *and*

*and*は論理積の演算子である。下位要素に設定された内容の論理積の演算結果を制約として設定する場合に使用する。

属性など

<b><i>and</i></b>	
Derived from	binary_logic_op_type
no attributes	
no relationships	

### 2.3.26 *or*

*or*は論理和の演算子である。下位要素に設定された内容の論理和の演算結果を制約として設定する場合に使用する。

属性など

<b><i>or</i></b>	
Derived from	binary_logic_op_type
no attributes	
no relationships	

### 2.3.27 *comparison\_ops\_type*

*comparison\_ops\_type*は、比較演算子を記述するための抽象要素である。

属性など

<b><i>comparison_ops_type</i></b>	
Derived from	constraint_type
no attributes	
no relationships	

### 2.3.28 *property\_is\_null\_type*

*property\_is\_null\_type*は、ヌル演算子を記述するための抽象要素である。

属性など

<b><i>property_is_null_type</i></b>	
Derived from	comparison_ops_type
no attributes	

no relationships
------------------

### 2.3.29 *property\_is\_null*

*property\_is\_null*はヌル制約である。設定された要素はヌルでなければならないことを指定する場合に使用する。

属性など

<b><i>property_is_null</i></b>	
Derived from	<i>property_is_null_type</i>
no attributes	
no relationships	

### 2.3.30 *binary\_comparison\_op\_type*

*binary\_comparison\_op\_type*は、値制約、範囲制約を記述するための抽象要素である。

属性など

<b><i>binary_comparison_op_type</i></b>		
Derived from	<i>comparison_ops_type</i>	
attributes		
matchCase	1	boolean
relationships		
literal	1	literal

#### 2.3.30.1 *matchCase*

概要

大文字小文字の区別を行うかを指定する。

意味

比較対象が文字列であった場合に、大文字小文字の区別を行うかを指定する。本属性にTrueが設定されている場合には、大文字小文字を区別して比較を行う。逆にFalseが設定されている場合には、大文字小文字は区別せずに比較を行う。必須属性。

### 2.3.31 *property\_is\_equal\_to*

*property\_is\_equal\_to*は等価制約である。設定された要素が指定された値でなければならないことを指定する場合に使用する。

属性など

<b><i>property_is_equal_to</i></b>	
Derived from	<i>binary_comparison_op_type</i>
no attributes	
no relationships	

### 2.3.32 *property\_is\_not\_equal\_to*

*property\_is\_not\_equal\_to*は非等価制約である。設定された要素が指定された値ではないことを指定する場合に使用する。

属性など

<b><i>property_is_not_equal_to</i></b>	
Derived from	binary_comparison_op_type
no attributes	
no relationships	

### 2.3.33 *property\_is\_less\_than*

*property\_is\_less\_than*は範囲制約である。設定された要素が、指定された値未満（指定された要素は含まれない）であることを指定する場合に使用する。

属性など

<b><i>property_is_less_than</i></b>	
Derived from	binary_comparison_op_type
no attributes	
no relationships	

### 2.3.34 *property\_is\_greater\_than*

*property\_is\_greater\_than*は範囲制約である。設定された要素が、指定された値を超える（指定された要素は含まれない）ことを指定する場合に使用する。

属性など

<b><i>property_is_greater_than</i></b>	
Derived from	binary_comparison_op_type
no attributes	
no relationships	

### 2.3.35 *property\_is\_less\_than\_or\_equal\_to*

*property\_is\_less\_than\_or\_equa\_to*は範囲制約である。設定された要素が、指定された値以下（指定された要素は含まれる）であることを指定する場合に使用する。

属性など

<b><i>property_is_less_than_or_equal_to</i></b>	
Derived from	binary_comparison_op_type
no attributes	
no relationships	

### 2.3.36 *property\_is\_greater\_than\_or\_equal\_to*

*property\_is\_greater\_than\_or\_eqla\_to*は範囲制約である。設定された要素が、指定された値以上（指定された要素は含まれる）であることを指定する場合に使用する。

属性など

<b><i>property_is_greater_than_or_equal_to</i></b>	
Derived from	binary_comparison_op_type
no attributes	
no relationships	

### 2.3.37 *property\_is\_like\_type*

*property\_is\_lile\_type*は、正規表現を用いたパターンマッチングによる文字列約を記述するための抽象要素である。

属性など

<b><i>property_is_like_type</i></b>		
Derived from	comparison_ops_type	
attributes		
wildChar	1	string
singleChar	1	string
escapeChar	1	string
relationships		
literal	1	literal

#### 2.3.37.1 *wildChar*

概要

複数文字にマッチする文字を指定する。

意味

正規表現を用いたパターンマッチングにおいて、0以上複数個の文字と一致することを表現する記号を設定する。必須属性。

#### 2.3.37.2 *singleChar*

概要

単一文字にマッチする文字を指定する。

意味

正規表現を用いたパターンマッチングにおいて、単一文字と一致することを表現する記号を設定する。必須属性。

#### 2.3.37.3 *escapeChar*

概要

マッチング対象から除外することを指示する文字を指定する。



### 意味

正規表現を用いたパターンマッチングにおいて、wildChar、singleChar、escapeCharをマッチング処理から除外するための記号を設定する。必須属性。

## 2.3.38 *property\_is\_like*

property\_is\_likeは、文字列に対するパターン制約である。正規表現を用いて設定可能な文字列のパターンを指定する場合に使用する。

### 属性など

<b><i>property_is_like</i></b>	
Derived from	property_is_like_type
no attributes	
no relationships	

## 2.3.39 *property\_is\_between\_type*

property\_is\_between\_typeは、範囲制約を記述するための抽象要素である。

### 属性など

<b><i>property_is_between_type</i></b>		
Derived from	comparison_ops_type	
no attributes		
relationships		
LowerBoundary	1	literal
UpperBoundary	1	literal

## 2.3.40 *property\_is\_between*

property\_is\_betweenは、範囲制約を指定する場合に使用する。上限値(UpperBoundary)および下限値(LowerBoundary)として指定された値も含まれる。

### 属性など

<b><i>property_is_between</i></b>	
Derived from	property_is_between_type
no attributes	
no relationships	

## 2.3.41 *literal*

literalは、各種制約を指定するための値を表す。

### 属性など

<b><i>literal</i></b>	
no attributes	
no relationships	

## 2.4 RTC ドキュメントプロファイル

### 2.4.1 doc\_basic

doc\_basicはRTCに関する仕様を文書化するためのメタ情報を記述する。RTCプロファイル(PIM)のコンポーネント部分の拡大図を図12に再掲する。

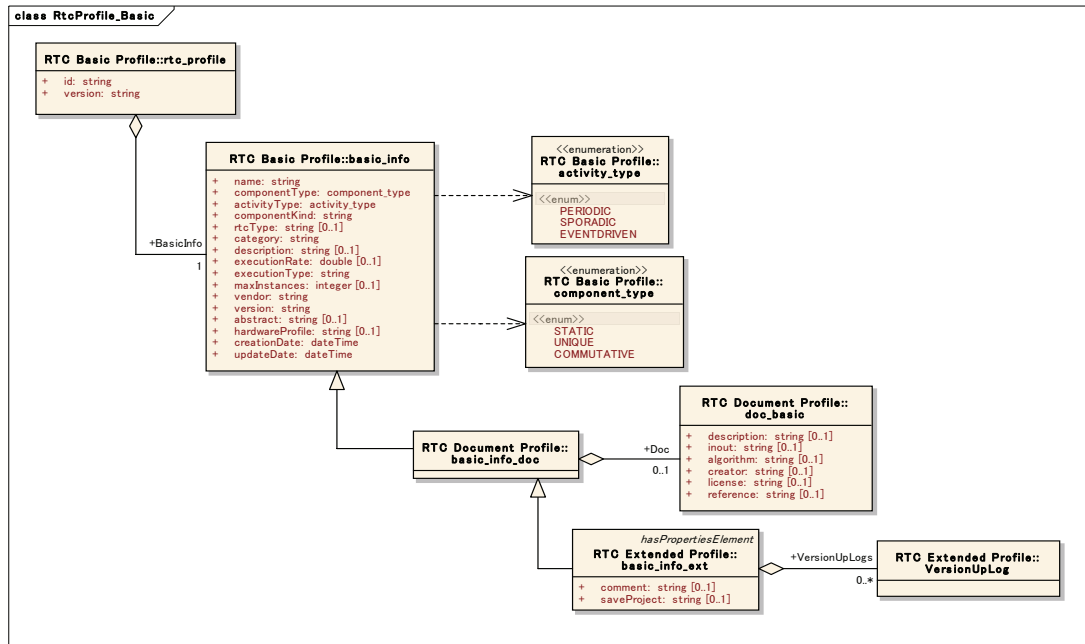


Figure 12 RTC プロファイル(コンポーネント部分)

#### 属性

<b>doc_basic</b>		
attributes		
algorithm	0..1	string
description	0..1	string
inout	0..1	string
creator	0..1	string
license	0..1	string
reference	0..1	string
no relationships		

#### 2.4.1.1 algorithm

##### 概要

RTCで使用しているアルゴリズムを記述する。

##### 意味

当該RTCにて利用しているアルゴリズムに関する説明を記述する。省略可能属性。

### 2.4.1.2 *description*

**概要**

RTCの概要説明を記述する。

**意味**

当該RTCの概要に関する説明を記述する。省略可能属性。

### 2.4.1.3 *inout*

**概要**

RTCの入出力の説明を記述する。

**意味**

当該RTCの入出力に関する説明を記述する。省略可能属性。

### 2.4.1.4 *creator*

**概要**

RTCの作成者情報を指定する。

**意味**

当該RTCの作成者、連絡先に関する説明を記述する。省略可能属性。

### 2.4.1.5 *license*

**概要**

RTCのライセンスの説明を記述する。

**意味**

当該RTCのライセンス、使用条件に関する説明を記述する。省略可能属性。

### 2.4.1.6 *reference*

**概要**

RTCの参考情報を記述する。

**意味**

当該RTCの参考情報、参考文献に関する説明を記述する。省略可能属性。

## 2.4.2 doc\_action

doc\_actionはRTCの各アクションに関する仕様を文書化するためのメタ情報を記述する。

RTCプロファイル(PIM)のアクション部分の拡大図を図13に再掲する。

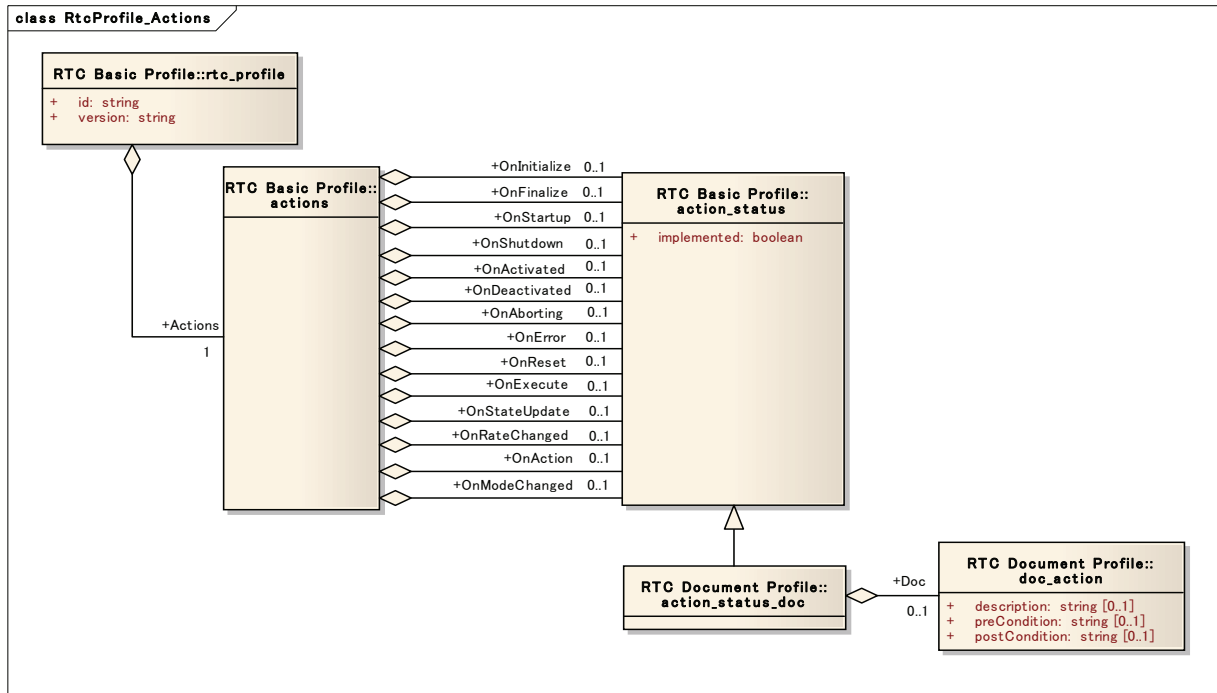


Figure13 RTC プロファイル(アクション部分)

### 属性

<i>doc_action</i>			
attributes			
description	0..1	String	
preCondition	0..1	String	
postCondition	0..1	string	
no relationships			

### 2.4.2.1 description

#### 概要

アクションの概要説明を記述する。

#### 意味

当該アクションの概要に関する説明を記述する。省略可能属性。

### 2.4.2.2 preCondition

#### 概要

アクションの事前条件を記述する。

### 意味

当該アクションを実行するための事前条件に関する説明を記述する。省略可能属性。

### 2.4.2.3 *postCondition*

#### 概要

アクションの事後条件を記述する。

#### 意味

当該アクションを実行した後成立する事後条件に関する説明を記述する。ただし、事前条件が満たされていない状態でアクションを実行した場合には事後条件の成立は保証されない。省略可能属性。

### 2.4.3 *doc\_configuration*

*doc\_configuration*はRTCのコンフィギュレーションに関する仕様を文書化するためのメタ情報を記述する。

RTCプロファイル(PIM)のコンフィギュレーション部分の拡大図を図14に再掲する。

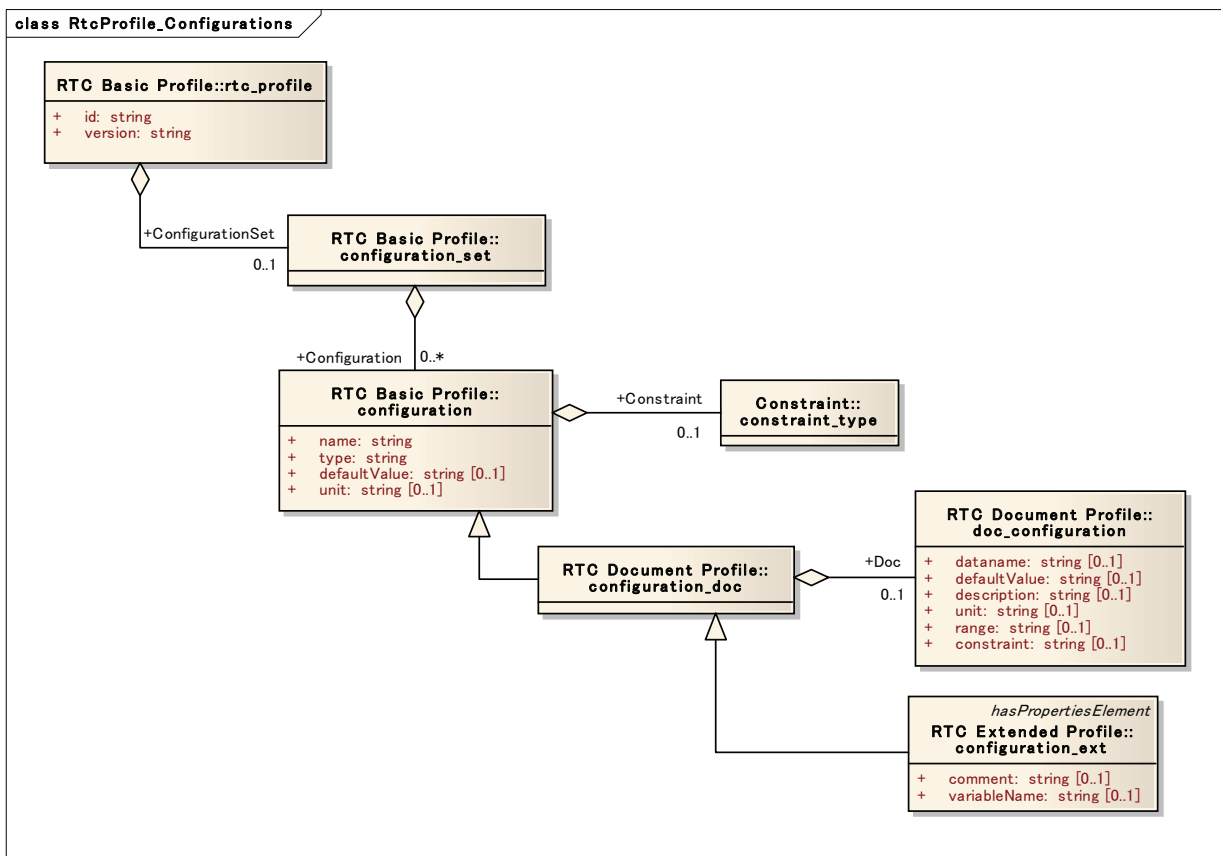


Figure 14 RTC プロファイル(コンフィギュレーション部分)

#### 属性

<b><i>doc_configuration</i></b>		
attributes		
description	0..1	string
dataname	0..1	string
defaultValue	0..1	string
unit	0..1	string
range	0..1	string
constraint	0..1	string
no relationships		

#### 2.4.3.1 *description*

##### 概要

コンフィギュレーションの概要説明を記述する。

##### 意味

当該コンフィギュレーションの概要に関する説明を記述する。省略可能属性。

#### 2.4.3.2 *dataname*

##### 概要

コンフィギュレーションのデータ名の説明を記述する。

##### 意味

当該コンフィギュレーションのデータ名に関する説明を記述する。省略可能属性。

#### 2.4.3.3 *defaultValue*

##### 概要

コンフィギュレーションのデフォルト値の説明を記述する。

##### 意味

当該コンフィギュレーションのデフォルト値に関する説明を記述する。省略可能属性。

#### 2.4.3.4 *unit*

##### 概要

コンフィギュレーションの単位に関する説明を記述する。

##### 意味

当該コンフィギュレーションの単位に関する説明を記述する。省略可能属性。

#### 2.4.3.5 *range*

##### 概要

コンフィギュレーションのデータ範囲の説明を記述する。

### 意味

当該コンフィギュレーションのデータ範囲に関する説明を記述する。省略可能属性。

## 2.4.3.6 constraint

### 概要

コンフィギュレーションの制約条件を記述する。

### 意味

当該コンフィギュレーションの制約条件に関する説明を記述する。省略可能属性。

## 2.4.4 doc\_dataport

doc\_dataportはRTCのDataPortに関する仕様を文書化するためのメタ情報を記述する。

RTCプロファイル(PIM)のデータポート部分の拡大図を図15に再掲する。

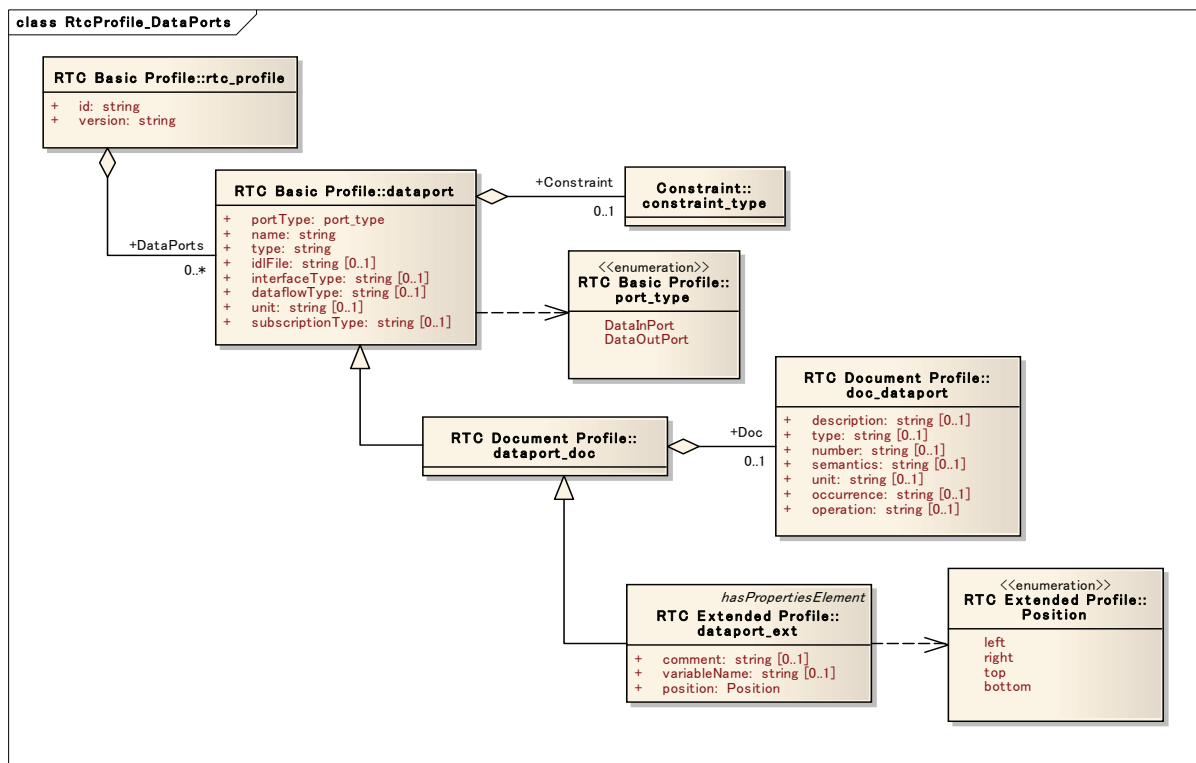


Figure15 RTC プロファイル(データポート部分)

#### 属性

<i>doc_dataport</i>		
attributes		
description	0..1	string
type	0..1	string
number	0..1	string
semantics	0..1	string
unit	0..1	string
occurrence	0..1	string
operation	0..1	string
no relationships		

#### 2.4.4.1 *description*

##### 概要

DataPortの概要説明を記述する。

##### 意味

当該DataPortの概要に関する説明を記述する。省略可能属性。

#### 2.4.4.2 *type*

##### 概要

DataPortのデータ型の説明を記述する。

##### 意味

当該DataPortにて送受信するデータの型に関する説明を記述する。省略可能属性。

#### 2.4.4.3 *number*

##### 概要

DataPortにて送受信するデータ数の説明を記述する。

##### 意味

当該DataPortにて送受信するデータ数に関する説明を記述する。送受信するデータが配列形式の時に使用。省略可能属性。

#### 2.4.4.4 *semantics*

##### 概要

DataPortで送受信するデータの意味を記述する。

##### 意味

当該DataPortにて送受信するデータの意味に関する説明を記述する。省略可能属性。

#### 2.4.4.5 *unit*

##### 概要

DataPortで送受信するデータの単位の説明を指定する。



### 意味

当該DataPortにて送受信するデータの単位に関する説明を記述する。省略可能属性。

#### 2.4.4.6 occurence

### 概要

DataPortの送受信の発生頻度、周期の説明を記述する。

### 意味

当該DataPortにて発生するデータ送受信の発生頻度、周期に関する説明を記述する。省略可能属性。

#### 2.4.4.7 operation

### 概要

DataPortの処理頻度、周期の説明を記述する。

### 意味

当該DataPortにて送受信したデータの処理速度、処理周期に関する説明を記述する。省略可能属性。

#### 2.4.5 doc\_serviceport

doc\_serviceportはRTCのServicePortに関する仕様を文書化するためのメタ情報を記述する。

RTCプロファイル(PIM)のサービスポート部分の拡大図を図16に再掲する。

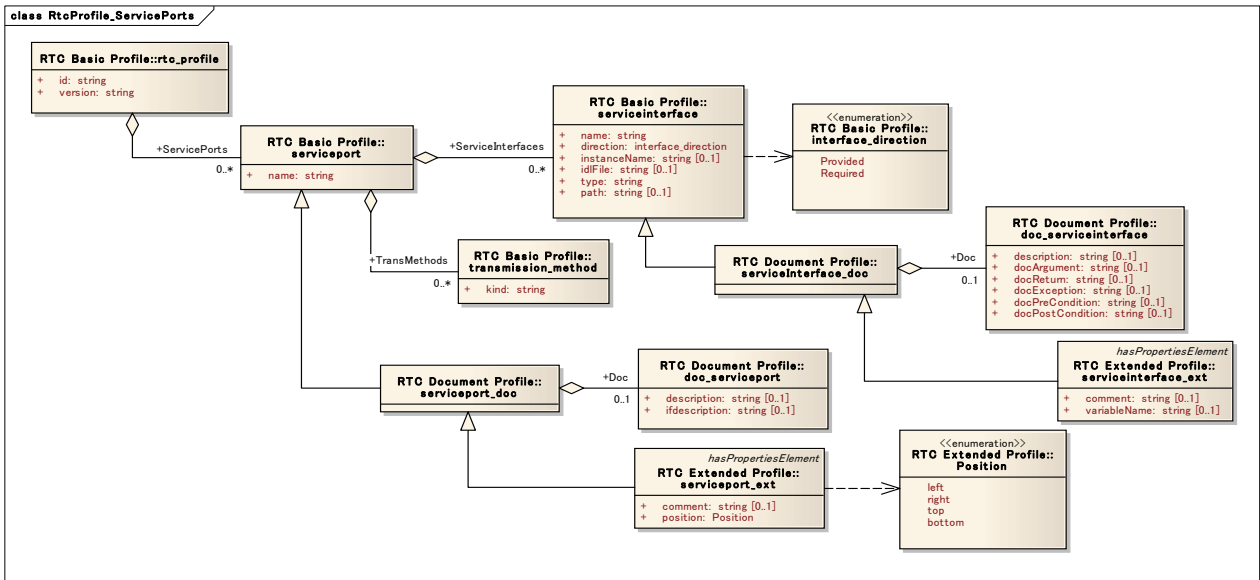


Figure16 RTC プロファイル(サービスポート部分)

属性

<b><i>doc_serviceport</i></b>		
attributes		
description	0..1	string
ifdescription	0..1	string
no relationships		

2.4.5.1 *description*

概要

ServicePortの概要説明を記述する。

意味

当該ServicePortの概要に関する説明を記述する。省略可能属性。

2.4.5.2 *ifdescription*

概要

ServicePortに指定されたインターフェースの概要説明を記述する。

意味

当該ServicePortで提供、利用しているインターフェースの概要に関する説明を記述する。省略可能属性。

2.4.6 *doc\_serviceinterface*

*doc\_serviceinterface*は、ServiceInterfaceに関する仕様を文書化するためのメタ情報を記述する。

属性

<b><i>doc_serviceinterface</i></b>		
attributes		
description	0..1	string
docArgument	0..1	string
docReturn	0..1	string
docException	0..1	string
docPreCondition	0..1	string
docPostCondition	0..1	string
no relationships		

2.4.6.1 *description*

概要

ServiceInterfaceの概要説明を記述する。

意味

当該ServiceInterfaceの概要に関する説明を記述する。省略可能属性。

#### 2.4.6.2 *docArgument*

##### 概要

ServiceInterfaceの引数の説明を記述する。

##### 意味

当該ServiceInterfaceで定義されている引数に関する説明を記述する。省略可能属性。

#### 2.4.6.3 *docReturn*

##### 概要

ServiceInterfaceの戻り値の説明を記述する。

##### 意味

当該ServiceInterfaceで定義されている戻り値に関する説明を記述する。省略可能属性。

#### 2.4.6.4 *docException*

##### 概要

ServiceInterfaceの例外の説明を記述する。

##### 意味

当該ServiceInterfaceを呼び出した際に発生する可能性のある例外に関する説明を記述する。省略可能属性。

#### 2.4.6.5 *docPreCondition*

##### 概要

ServiceInterfaceの事前条件の説明を記述する。

##### 意味

当該ServiceInterfaceを呼び出す前に成立すべき事前条件に関する説明を記述する。省略可能属性。

#### 2.4.6.6 *docPostCondition*

##### 概要

当該ServiceInterfaceの事後条件の説明を記述する。

##### 意味

当該ServiceInterfaceを呼びだし後に成立している事後条件に関する説明を記述する。ただし、事前条件が満たされていない状態でサービスが呼び出された場合には事後条件の成立は保証されない。省略可能属性。

## 2.5 RTC 拡張プロファイル

### 2.5.1 basic\_info\_ext

basic\_info\_extはRTCの付加情報に関するメタ情報を記述する。

RTCプロファイル(PIM)のコンポーネント部分の拡大図を図17に再掲する。

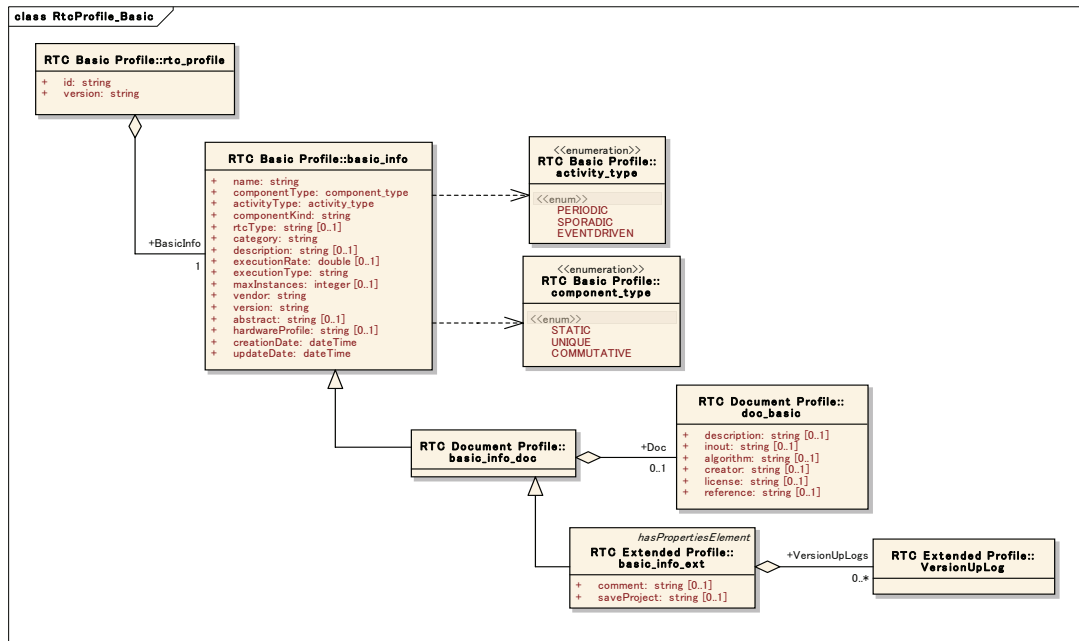


Figure17 RTC プロファイル(基本部分)

#### 属性

<b>basic_info_ext</b>			
Attributes			
comment	0..1		string
saveProject	0..1		string
VersionUpLog	0..*		string
relationships			
Properties	0..*		Property

#### 2.5.1.1 comment

##### 概要

RTCに対する各種コメントを記述する。

##### 意味

RTC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。

### 2.5.1.2 saveProject

#### 概要

RTC設計情報の保存先プロジェクト名を設定する。

#### 意味

RtcBuilderを用いて当該RTCの設計を行った際に、設計内容を保存したプロジェクト名を設定する。省略可能属性。

### 2.5.1.3 VersionUpLog

#### 概要

RTCの各バージョンに関する補足説明を記述する。

#### 意味

RTC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。

### 2.5.2 configuration\_ext

configuration\_extはConfigurationの付加情報に関するメタ情報を記述する。

RTCプロファイル(PIM)のコンフィギュレーション部分の拡大図を図 1 8 に示す。

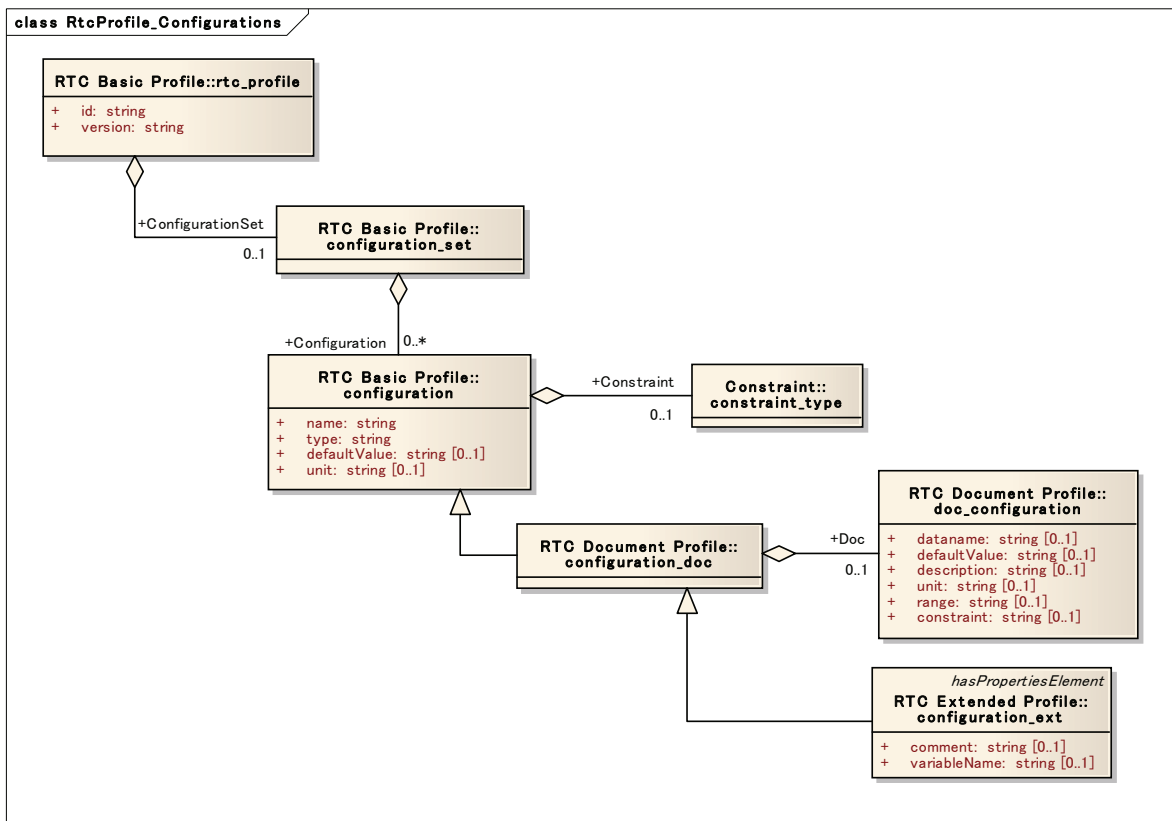


Figure18 RTCプロファイル(コンフィギュレーション部分)

## 属性

<i>configuration_ext</i>		
No attributes		
comment	0..1	string
variableName	0..1	string
relationships		
Properties	0..*	Property

### 2.5.2.1 comment

#### 概要

コンフィギュレーションに対する各種コメントを記述する。

#### 意味

コンフィギュレーション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。

### 2.5.2.2 variableName

#### 概要

パラメータの変数名を指定する。

#### 意味

実装において使用されるパラメータの変数名を指定する。この属性は省略可能である。省略された場合は、Basic Profile内のnameが代わりに使用される。実際の変数名へのマッピングはPSMにおいて定義される。省略可能属性。

### 2.5.3 dataport\_ext

dataport\_extはDataPortの付加情報に関するメタ情報を記述する。

RTCプロフィール(PIM)のデータポート部分の拡大図を図19に再掲する。

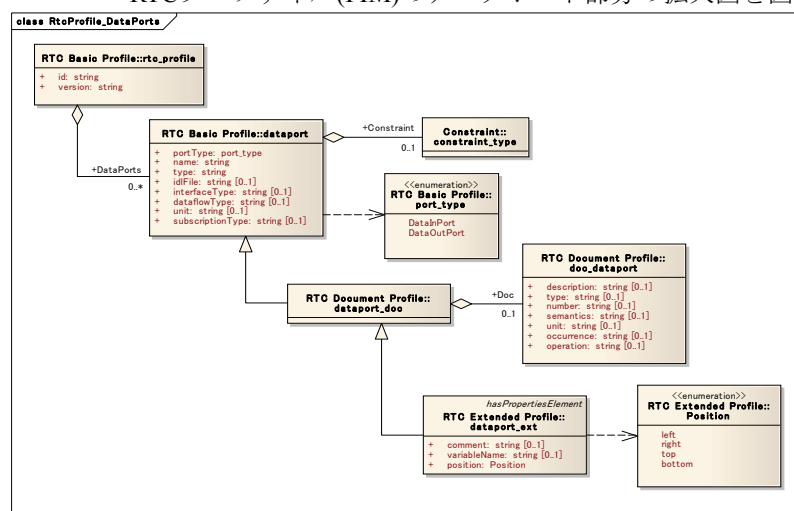


Figure 19 RTC プロファイル(データポート部分)

## 属性

<b><i>dataport_ext</i></b>		
attributes		
comment	0..1	string
variableName	0..1	string
position	1	Position
relationships		
Properties	0..*	Property

### 2.5.3.1 *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。

### 2.5.3.2 *variableName*

#### 概要

DataPortの変数名を指定する。

#### 意味

実装において使用されるDataPortの変数名を指定する。この属性は省略可能であり、存在しない場合はnameが代わりに使用される。実際の変数名へのマッピングはPSMにおいて定義される。省略可能属性。

### 2.5.3.3 *position*

#### 概要

DataPortの表示位置を指定する。

#### 意味

各種ツールにおいて当該DataPortを表示する際の表示位置を指定する。必須属性。

## 2.5.4 serviceport\_ext

serviceport\_extはServicePortの付加情報に関するメタ情報を記述する。

RTCプロファイル(PIM)のサービスポート部分の拡大図を図20に再掲する。

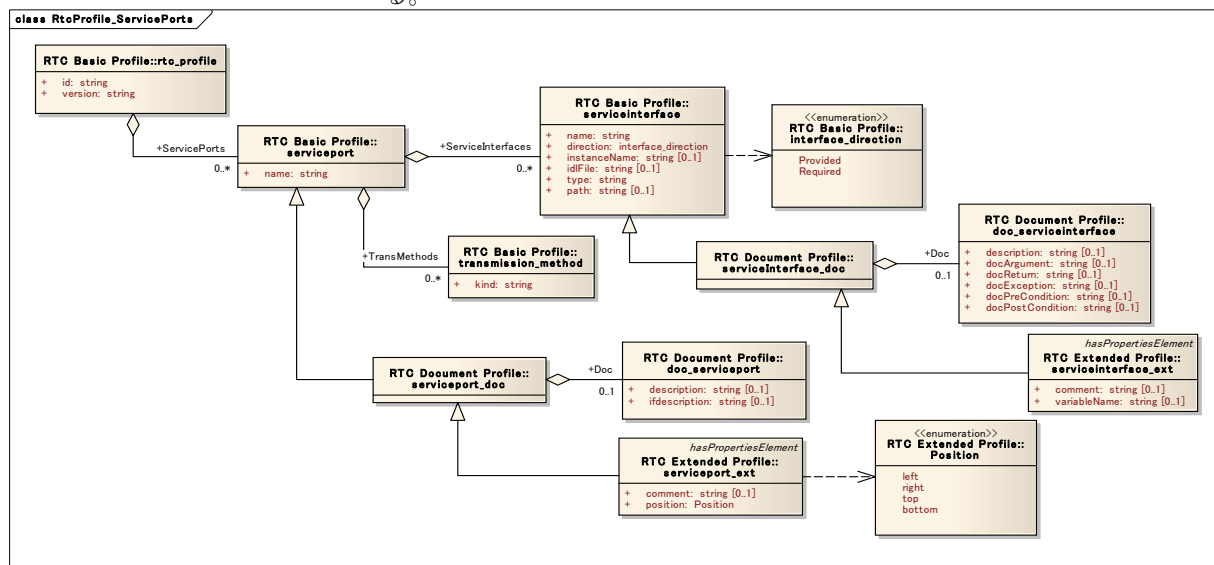


Figure 20 RTC プロファイル(サービスポート部分)

### 属性

<i>serviceport_ext</i>		
attributes		
comment	0..1	string
position	1	Position
relationships		
Properties	0..*	Property

### 2.5.4.1 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。

### 2.5.4.2 position

#### 概要

ServicePortの表示位置を指定する。

#### 意味

各種ツールにおいて当該ServicePortを表示する際の表示位置を指定する。必須属性。



## 2.5.5 *serviceinterface\_ext*

*serviceinterface\_ext*はServiceInterfaceの付加情報に関するメタ情報を記述する。

属性

<b><i>serviceinterface_ext</i></b>		
attributes		
comment	0..1	string
variableName	0..1	string
relationships		
Properties	0..*	Property

### 2.5.5.1 *comment*

概要

ServiceInterfaceに対する各種コメントを記述する。

意味

ServiceInterface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。

### 2.5.5.2 *variableName*

概要

ServiceInterfaceの変数名を指定する。

意味

実装において使用されるServiceInterfaceの変数名を指定する。省略可能属性。

## 2.5.6 *Position*

*Position*は、各種Portの表示位置を指定するための列挙型である。

属性

<b><i>Position</i></b>	
attributes	
left	Position
right	Position
top	Position
bottom	Position

## 2.5.7 language\_ext

language\_extは当該RTC生成言語の付加情報に関するメタ情報を記述する。  
RTCプロファイル(PIM)の言語部分の拡大図を図21に再掲する。

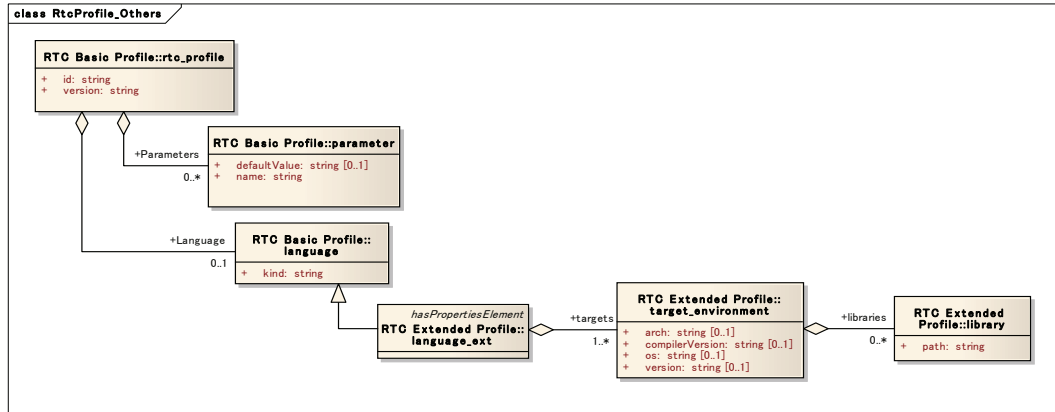


Figure 21 RTC プロファイル(言語部分)

属性

<i>language_ext</i>		
no attributes		
relationships		
targets	1..*	target_environment
Properties	0..*	Property

## 2.5.8 target\_environment

target\_environmentは、当該RTCの実行環境に関するメタ情報を記述する。

属性

<i>target_environment</i>			
attributes			
arch	0..1	string	
compilerVersion	0..1	string	
os	0..1	string	
version	0..1	string	
relationships			
libraries	0..*	library	

### 2.5.8.1 arch

概要

RTCの実行環境であるアーキテクチャを指定する。

意味

当該RTCが動作するアーキテクチャを設定する。本属性は当該RTCの設計者が設計時に想定したアーキテクチャを設定する。省略可能属性。

### 2.5.8.2 *compilerVersion*

#### 概要

RTCのコンパイラバージョンを指定する。

#### 意味

当該RTCがコンパイル言語で実装されている場合、コンパイラのバージョンを設定する。本属性は当該RTCの設計者が設計時に想定したコンパイラバージョンを設定する。省略可能属性。

### 2.5.8.3 *os*

#### 概要

RTCの実行環境であるOSを指定する。

#### 意味

当該RTCが動作するOSを設定する。本属性は当該RTCの設計者が設計時に想定したOSを設定する。省略可能属性。

### 2.5.8.4 *version*

#### 概要

RTCの実行環境であるOSのバージョンを指定する。

#### 意味

当該RTCが動作するOSのバージョンを設定する。本属性は当該RTCの設計者が設計時に想定したOSのバージョンを設定する。省略可能属性。

## 2.5.9 *library*

*library*は対象RTCが必要とする外部ライブラリに関するメタ情報を記述する。

#### 属性

<i>library</i>		
attributes		
path	1	string
no relationships		

### 2.5.9.1 *path*

#### 概要

RTCが利用する外部ライブラリの場所を指定する。

#### 意味

対象RTCが必要とする外部ライブラリの場所を指定する。

## 2.5.10 *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		

#### 2.5.10.1 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。

#### 2.5.10.2 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 エラー! 参照元が見つかりません。 の準拠性に従って定義される。

### Contents

This chapter contains the following sections.

Section Title	Page
UML-XML変換	61
UML-YAML変換	65

### 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。

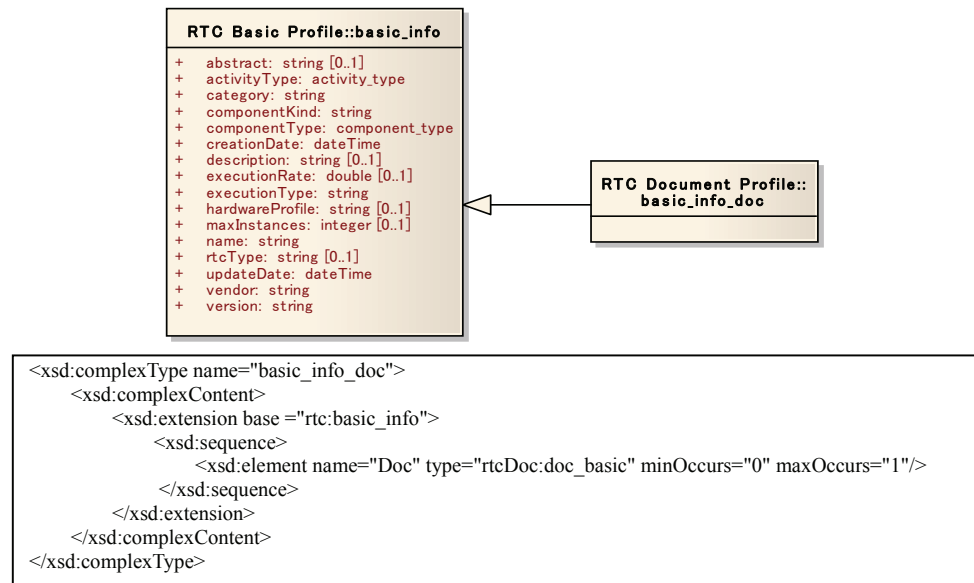
- string → xsd:string
- double → xsd:double
- integer → xsd:integer
- dateTime → xsd:dateTime
- boolean → xsd:boolean

### 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。

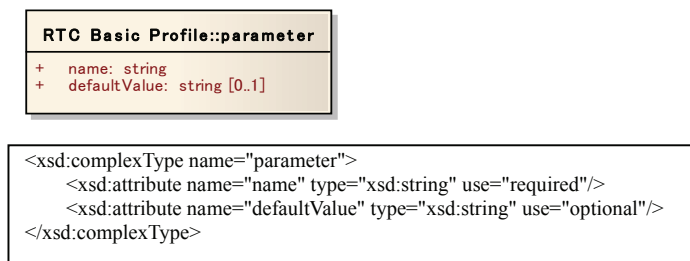
- RTC Basic Profile → rtc
- RTC Document Profile → rtcDoc
- RTC Extended Profile → rtcExt

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。

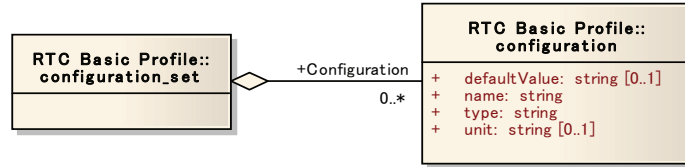


### 3.1.3 クラス

PIMにてクラスとして定義した要素は、XMLスキーマでは以下のように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。

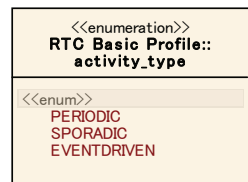


```

<xsd:complexType name="configuration_set">
  <xsd:sequence>
    <xsd:element name="Configuration" type="rtc:configuration" minOccurs="0" maxOccurs="unbounded"/>
  </xsd:sequence>
</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。

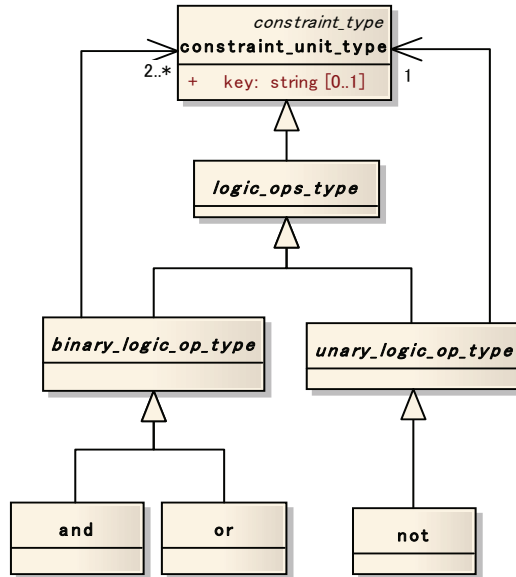


```

<xsd:attribute name="activityType" use="required">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="PERIODIC"/>
      <xsd:enumeration value="SPORADIC"/>
      <xsd:enumeration value="EVENTDRIVEN"/>
    </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。ただし、抽象クラスについてはマッピングを行わず、下位の具象クラスのレベルでのマッピングを行う。



```

<xsd:complexType name="constraint_unit_type">
  <xsd:choice>
    <xsd:element name="Not" type="rtc:not"/>
    <xsd:element name="And" type="rtc:and"/>
    <xsd:element name="Or" type="rtc:or"/>
  </xsd:choice>
  <xsd:attribute name="key" type="xsd:string" use="optional"/>
</xsd:complexType>

<xsd:complexType name="not">
  <xsd:sequence>
    <xsd:element name="ConstraintType" type="rtc:constraint_type" minOccurs="1" maxOccurs="1"/>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="and">
  <xsd:sequence>
    <xsd:element name="ConstraintType" type="rtc:constraint_type" minOccurs="2" maxOccurs="2"/>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="or">
  <xsd:sequence>
    <xsd:element name="ConstraintType" type="rtc:constraint_type" minOccurs="2" maxOccurs="2"/>
  </xsd:sequence>
</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

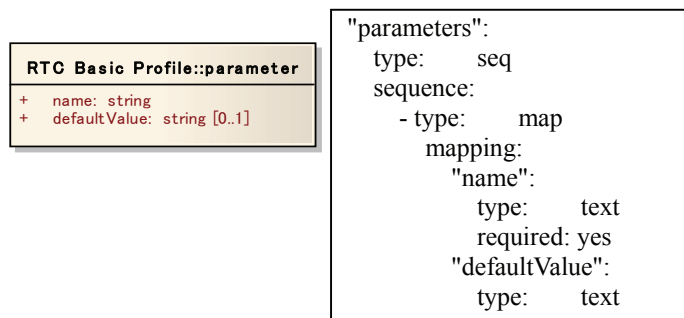
PIMにて定義したパッケージは、以下のようにYAMLスキーマの名前空間としてマッピングする。

- RTC Basic Profile → なし
- RTC Document Profile → rtcDoc
- RTC Extended Profile → rtcExt

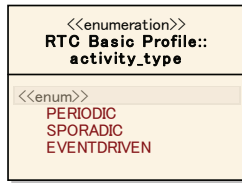
各名前空間に属する要素については、タグ名の前に「::」区切りにて名前空間接頭語を付加する。

### 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。各クラスの属性として定義した要素は、マッピング対象の各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。



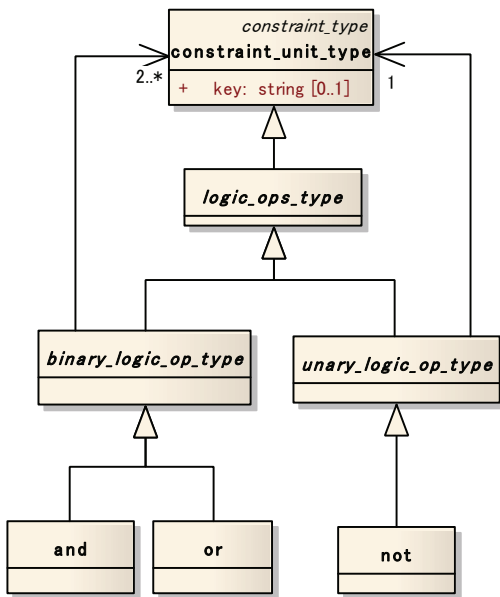
PIMにてenum型として定義した要素は、YAMLスキーマではenumを用いて定義する。



```

    "activityType":
      type: text
      required: yes
      enum:
        - PERIODIC
        - SPORADIC
        - EVENTDRIVEN
  
```

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。ただし、抽象クラスについてはマッピングを行わず、下位の具象クラスのレベルでのマッピングを行う。



```

    "constraintUnit": &constraintUnit
      type: map
      mapping:
        "key":
          type: text
        "not":
          type: map
          mapping:
            "constraint": * constraintUnit
        "and":
          type: seq
          sequence:
            - *constraintUnit
        "or":
          type: seq
          sequence:
            - *constraintUnit
  
```

## 4 RTC表記法

---

ここでは、UML表記法に従ったRTCの構成図の表記法を示す。

### 4.1 RTC 構成図記述方式

RTコンポーネント構成図は、複数のRTコンポーネントの接続関係、データの流れを可視化するために使用する。

RTコンポーネント構成図の記法は、以下のとおり。

- UML2のコンポーネント図にて記載する。
- RTコンポーネントには、ステレオタイプ<< RTC >>を付与する。
- データ入力ポートは、ステレオタイプ<< DataInPort >>を付与し、データポート名（データ名+Port、省略化）を記述する。
- データ出力ポートは、ステレオタイプ<< DataOutPort >>を付与し、データポート名（データ名+Port、省略化）を記述する。
- サービスポートは、インタフェース（プロバイダ、コンシューマ）として表現し、サービスポート名（省略化）を記述する。
- 必要に応じて、ノート（メモ）を記載する。
- 外部要素を記述する場合にはアクタを用いて表現する。
- コンポジット構造型のRTCを表現する場合には、RTCを入れ子で記述する。

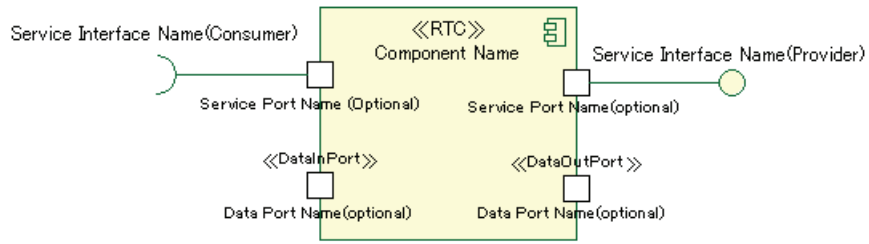


図 1 RT コンポーネント構成図凡例 (RT コンポーネント単体)

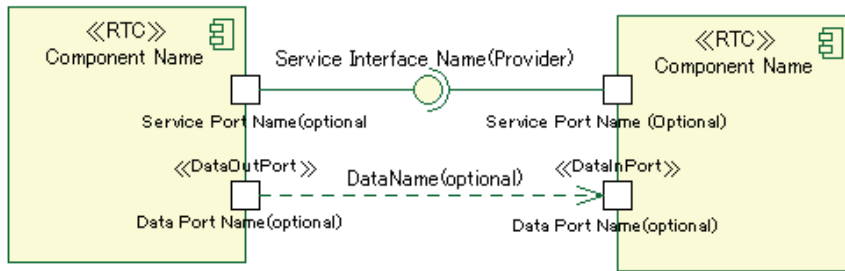


図 1 RT コンポーネント構成図凡例 (RT コンポーネント複数)

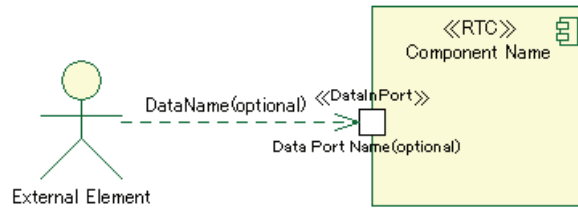


図 3 RT コンポーネント構成図凡例 (外部要素あり)

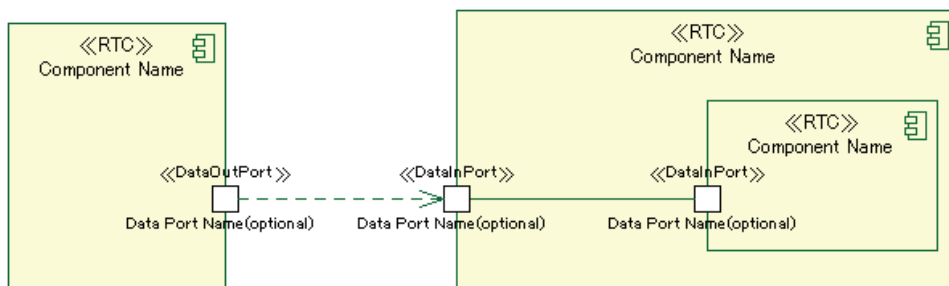


図 4 RT コンポーネント構成図凡例 (コンポジットコンポーネント)

## 4.2 RTC 構成図例

RT コンポーネント構成図のサンプルを以下に示す。

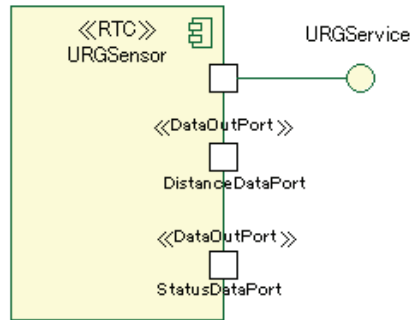


図 2 RT コンポーネント構成図サンプル (RT コンポーネント単体)

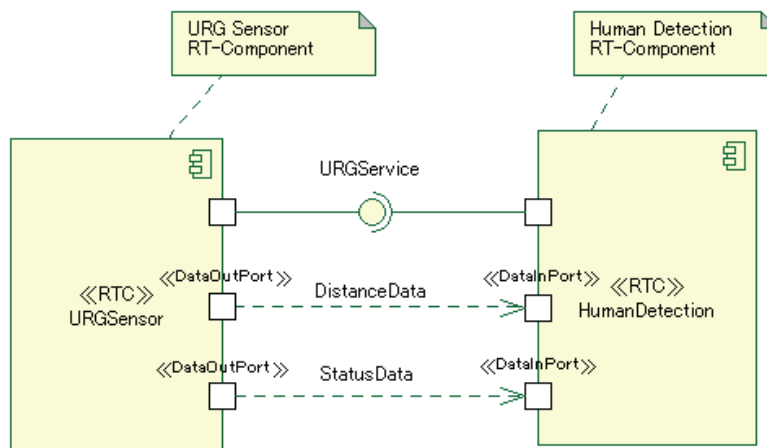


図3 RTコンポーネント構成図サンプル( RTコンポーネント複数)

## 5 Appendix: 各種定義

---

### 5.1 ComponentKind

DataFlowComponent
FiniteStateMachineComponent
DataFlowFiniteStateMachineComponent
FiniteStateMachineMultiModeComponent
DataFlowMultiModeComponent
DataFlowFiniteStateMachineMultiMode Component

### 5.2 executionType

PERIODIC
EVENTDRIVEN
OTHER

### 5.3 データ型

TimedShort	TimedShortSeq
TimedLong	TimedLongSeq
TimedUShort	TimedUShortSeq
TimedULong	TimedULongSeq
TimedFloat	TimedFloatSeq
TimedDouble	TimedDoubleSeq
TimedChar	TimedCharSeq
TimedBoolean	TimedBooleanSeq
TimedOctet	TimedOctetSeq
TimedString	TimedStringSeq

## 5.4 システムコンフィギュレーション

設定値名称	意味	デフォルト値
corba.args	ORB 利用時に引き渡す引数	(なし)
corba.endpoint	CORBA 利用時のインターフェース設定	(なし)
corba.nameservers	CORBA 利用時のネームサーバ名称	(なし)
exec_cxt.evdriven.type	イベントドリブン型 ExecutionContext のクラス名	EventDrivenExecutionContext
exec_cxt.periodic.rate	周期実行型 ExecutionContext の実行周期 [Hz]	1000
exec_cxt.periodic.type	周期実行型 ExecutionContext のクラス名	PeriodicExecutionContext
logger.data_format	ログファイル内に出力する日時書式	yyyy/MM/dd HH:mm:ss
logger.enable	ロギング機能 有効/無効 設定	NO (無効)
logger.filename	ログ出力ファイル名称	./rtc%p.log
logger.log_level	ロギングレベル	NORMAL
logger.master_logger		(なし)
logger.stream_lock	ログファイルのロック機能 有効/無効 設定	NO (無効)
manager.modules.load_path	マネージャモジュールのロードパス	(なし)
manager.name	RTC 用マネージャの名称	manager
manager.pid	RTC 用マネージャのプロセス ID	(なし)
module.conf_path	モジュールのコンフィギュレーション情報を配置している場所へのパス	(なし)
naming.enable	ネームサービス機能 有効/無効 設定	YES (有効)
naming.formats	ネームサービス上での書式設定	%h.host/%n.rtc
naming.type	ネームサービスの型	corba
naming.update.enable	ネームサービスの自動更新 有効/無効 設定	YES (有効)
naming.update.interval	ネームサービス自動更新時の更新周期 (s)	10.0
os.arch	利用している OS のアーキテクチャ	(なし)
os.hostname	利用している OS 上でのホスト名	(なし)
os.name	利用している OS の名称	(なし)
os.release	利用している OS のリリース	(なし)
os.version	利用している OS のバージョン	(なし)
timer.enable	タイマ機能 有効/無効 設定	YES (有効)
timer.tick	タイマの動作周期 (s)	0.1

## 6 Appendix: XMLスキーマ

### 6.1 RTC 基本Profile(RtcProfile\_basic.xsd)

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

  <xsd:element name="RtcProfile" type="rtc:rtc_profile"/>
  <xsd:complexType name="rtc_profile">
    <xsd:sequence>
      <xsd:element name="BasicInfo" type="rtc:basic_info" minOccurs="1" maxOccurs="1"/>
      <xsd:element name="Actions" type="rtc:actions" minOccurs="1" maxOccurs="1"/>
      <xsd:element name="ConfigurationSet" type="rtc:configuration_set" minOccurs="0" maxOccurs="1"/>
      <xsd:element name="DataPorts" type="rtc:dataport" minOccurs="0" maxOccurs="unbounded"/>
      <xsd:element name="ServicePorts" type="rtc:serviceport" minOccurs="0" maxOccurs="unbounded"/>
      <xsd:element name="Parameters" type="rtc:parameter" minOccurs="0" maxOccurs="unbounded"/>
      <xsd:element name="Language" type="rtc:language" 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:complexType>

  <xsd:complexType name="basic_info">
    <xsd:attribute name="name" type="xsd:string" use="required"/>
    <xsd:attribute name="componentType" use="required">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:enumeration value="STATIC"/>
          <xsd:enumeration value="UNIQUE"/>
          <xsd:enumeration value="COMMUTATIVE"/>
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:attribute>
    <xsd:attribute name="activityType" use="required">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:enumeration value="PERIODIC"/>
          <xsd:enumeration value="SPORADIC"/>
          <xsd:enumeration value="EVENTDRIVEN"/>
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:attribute>
  </xsd:complexType>

```



```

<xsd:attribute name="componentKind" type="xsd:string" use="required"/>
<xsd:attribute name="rtcType" type="xsd:string" use="optional"/>
<xsd:attribute name="category" type="xsd:string" use="required"/>
<xsd:attribute name="description" type="xsd:string" use="optional"/>
<xsd:attribute name="executionRate" type="xsd:double" use="optional"/>
<xsd:attribute name="executionType" type="xsd:string" use="required"/>
<xsd:attribute name="maxInstances" type="xsd:integer" use="optional"/>
<xsd:attribute name="vendor" 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="hardwareProfile" 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="actions">
  <xsd:sequence>
    <xsd:element name="OnInitialize" type="rtc:action_status" minOccurs="0" maxOccurs="1"/>
    <xsd:element name="OnFinalize" type="rtc:action_status" minOccurs="0" maxOccurs="1"/>
    <xsd:element name="OnStartup" type="rtc:action_status" minOccurs="0" maxOccurs="1"/>
    <xsd:element name="OnShutdown" type="rtc:action_status" minOccurs="0" maxOccurs="1"/>
    <xsd:element name="OnActivated" type="rtc:action_status" minOccurs="0" maxOccurs="1"/>
    <xsd:element name="OnDeactivated" type="rtc:action_status" minOccurs="0" maxOccurs="1"/>
    <xsd:element name="OnAborting" type="rtc:action_status" minOccurs="0" maxOccurs="1"/>
    <xsd:element name="OnError" type="rtc:action_status" minOccurs="0" maxOccurs="1"/>
    <xsd:element name="OnReset" type="rtc:action_status" minOccurs="0" maxOccurs="1"/>
    <xsd:element name="OnExecute" type="rtc:action_status" minOccurs="0" maxOccurs="1"/>
    <xsd:element name="OnStateUpdate" type="rtc:action_status" minOccurs="0" maxOccurs="1"/>
    <xsd:element name="OnRateChanged" type="rtc:action_status" minOccurs="0" maxOccurs="1"/>
    <xsd:element name="OnAction" type="rtc:action_status" minOccurs="0" maxOccurs="1"/>
    <xsd:element name="OnModeChanged" type="rtc:action_status" minOccurs="0" maxOccurs="1"/>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="action_status">
  <xsd:attribute name="implemented" type="xsd:boolean" use="required"/>
</xsd:complexType>

<xsd:complexType name="configuration_set">
  <xsd:sequence>
    <xsd:element name="Configuration" type="rtc:configuration" minOccurs="0" maxOccurs="unbounded"/>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="configuration">
  <xsd:sequence>
    <xsd:element name="Constraint" type="rtc:constraint_type" minOccurs="0" maxOccurs="1"/>
  </xsd:sequence>
  <xsd:attribute name="name" type="xsd:string" use="required"/>
  <xsd:attribute name="type" type="xsd:string" use="required"/>
  <xsd:attribute name="defaultValue" type="xsd:string" use="optional"/>
  <xsd:attribute name="unit" type="xsd:string" use="optional"/>
</xsd:complexType>

<xsd:complexType name="dataport">
  <xsd:sequence>
    <xsd:element name="Constraint" type="rtc:constraint_type" minOccurs="0" maxOccurs="1"/>
  </xsd:sequence>
  <xsd:attribute name="portType" use="required">
    <xsd:simpleType>
      <xsd:restriction base="xsd:string">
        <xsd:enumeration value="DataInPort"/>
        <xsd:enumeration value="DataOutPort"/>
      </xsd:restriction>
    </xsd:simpleType>
  </xsd:attribute>
  <xsd:attribute name="name" type="xsd:string" use="required"/>
  <xsd:attribute name="type" type="xsd:string" use="required"/>
  <xsd:attribute name="idlFile" type="xsd:string" use="optional"/>

```

```

<xsd:attribute name="interfaceType" type="xsd:string" use="optional"/>
<xsd:attribute name="dataflowType" type="xsd:string" use="optional"/>
<xsd:attribute name="subscripionType" type="xsd:string" use="optional"/>
<xsd:attribute name="unit" type="xsd:string" use="optional"/>
</xsd:complexType>

<xsd:complexType name="serviceport">
  <xsd:sequence>
    <xsd:element name="ServiceInterface" type="rtc:serviceinterface" minOccurs="0" maxOccurs="unbounded"/>
    <xsd:element name="TransMethods" type="rtc:transmission_method" minOccurs="0" maxOccurs="unbounded"/>
  </xsd:sequence>
  <xsd:attribute name="name" type="xsd:string" use="required"/>
</xsd:complexType>

<xsd:complexType name="serviceinterface">
  <xsd:attribute name="name" type="xsd:string" use="required"/>
  <xsd:attribute name="direction" use="required">
    <xsd:simpleType>
      <xsd:restriction base="xsd:string">
        <xsd:enumeration value="Provided"/>
        <xsd:enumeration value="Required"/>
      </xsd:restriction>
    </xsd:simpleType>
  </xsd:attribute>
  <xsd:attribute name="instanceName" type="xsd:string" use="optional"/>
  <xsd:attribute name="idlFile" type="xsd:string" use="optional"/>
  <xsd:attribute name="type" type="xsd:string" use="required"/>
  <xsd:attribute name="path" type="xsd:string" use="optional"/>
</xsd:complexType>

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

<xsd:complexType name="parameter">
  <xsd:attribute name="name" type="xsd:string" use="required"/>
  <xsd:attribute name="defaultValue" type="xsd:string" use="optional"/>
</xsd:complexType>

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

<xsd:complexType name="constraint_type">
  <xsd:choice>
    <xsd:element name="ConstraintHashType" type="rtc:constraint_hash_type"/>
    <xsd:element name="ConstraintListType" type="rtc:constraint_list_type"/>
    <xsd:element name="ConstraintUnitType" type="rtc:constraint_unit_type"/>
  </xsd:choice>
</xsd:complexType>

<xsd:complexType name="constraint_unit_type">
  <xsd:choice>
    <xsd:element name="Not" type="rtc:not"/>
    <xsd:element name="And" type="rtc:and"/>
    <xsd:element name="Or" type="rtc:or"/>
    <xsd:element name="propertyIsNullType" type="rtc:property_is_null_type"/>
    <xsd:element name="propertyIsEqualTo" type="rtc:property_is_equal_to"/>
    <xsd:element name="propertyIsNotEqualTo" type="rtc:property_is_not_equal_to"/>
    <xsd:element name="propertyIsLessThan" type="rtc:property_is_less_than"/>
    <xsd:element name="propertyIsGreaterThan" type="rtc:property_is_greater_than"/>
    <xsd:element name="propertyIsLessThanOrEqualTo" type="rtc:property_is_less_than_or_equal_to"/>
    <xsd:element name="propertyIsGreaterThanOrEqualTo" type="rtc:property_is_greater_than_or_equal_to"/>
    <xsd:element name="propertyIsLike" type="rtc:property_is_like"/>
    <xsd:element name="propertyIsBetween" type="rtc:property_is_between"/>
  </xsd:choice>
  <xsd:attribute name="key" type="xsd:string" use="optional"/>
</xsd:complexType>

```

```

<xsd:complexType name="constraint_hash_type">
  <xsd:sequence>
    <xsd:element name="ConstraintUnit" type="rtc:constraint_unit_type" minOccurs="0" maxOccurs="unbounded"/>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="constraint_list_type">
  <xsd:sequence>
    <xsd:element name="ConstraintUnit" type="rtc:constraint_unit_type" minOccurs="0" maxOccurs="unbounded"/>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="not">
  <xsd:sequence>
    <xsd:element name="Constraint" type="rtc:constraint_type" minOccurs="1" maxOccurs="1"/>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="and">
  <xsd:sequence>
    <xsd:element name="Constraint" type="rtc:constraint_type" minOccurs="2" maxOccurs="2"/>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="or">
  <xsd:sequence>
    <xsd:element name="Constraint" type="rtc:constraint_type" minOccurs="2" maxOccurs="2"/>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="property_is_null_type">
  <xsd:choice>
    <xsd:element name="propertyIsNull" type="xsd:string"/>
  </xsd:choice>
</xsd:complexType>

<xsd:complexType name="property_is_equal_to">
  <xsd:sequence>
    <xsd:element name="Literal" type="xsd:string" minOccurs="1" maxOccurs="1"/>
  </xsd:sequence>
  <xsd:attribute name="matchCase" type="xsd:boolean" use="required"/>
</xsd:complexType>

<xsd:complexType name="property_is_not_equal_to">
  <xsd:sequence>
    <xsd:element name="Literal" type="xsd:string" minOccurs="1" maxOccurs="1"/>
  </xsd:sequence>
  <xsd:attribute name="matchCase" type="xsd:boolean" use="required"/>
</xsd:complexType>

<xsd:complexType name="property_is_less_than">
  <xsd:sequence>
    <xsd:element name="Literal" type="xsd:string" minOccurs="1" maxOccurs="1"/>
  </xsd:sequence>
  <xsd:attribute name="matchCase" type="xsd:boolean" use="required"/>
</xsd:complexType>

<xsd:complexType name="property_is_greater_than">
  <xsd:sequence>
    <xsd:element name="Literal" type="xsd:string" minOccurs="1" maxOccurs="1"/>
  </xsd:sequence>
  <xsd:attribute name="matchCase" type="xsd:boolean" use="required"/>
</xsd:complexType>

<xsd:complexType name="property_is_less_than_or_equal_to">
  <xsd:sequence>
    <xsd:element name="Literal" type="xsd:string" minOccurs="1" maxOccurs="1"/>
  </xsd:sequence>
  <xsd:attribute name="matchCase" type="xsd:boolean" use="required"/>
</xsd:complexType>

```

```
<xsd:complexType name="property_is_greater_than_or_equal_to">
  <xsd:sequence>
    <xsd:element name="Literal" type="xsd:string" minOccurs="1" maxOccurs="1"/>
  </xsd:sequence>
  <xsd:attribute name="matchCase" type="xsd:boolean" use="required"/>
</xsd:complexType>

<xsd:complexType name="property_is_like">
  <xsd:sequence>
    <xsd:element name="Literal" type="xsd:string" minOccurs="1" maxOccurs="1"/>
  </xsd:sequence>
  <xsd:attribute name="wildCard" type="xsd:string" use="required"/>
  <xsd:attribute name="singleChar" type="xsd:string" use="required"/>
  <xsd:attribute name="escapeChar" type="xsd:string" use="required"/>
</xsd:complexType>

<xsd:complexType name="property_is_between">
  <xsd:sequence>
    <xsd:element name="LowerBoundary" type="xsd:string" minOccurs="1" maxOccurs="1"/>
    <xsd:element name="UpperBoundary" type="xsd:string" minOccurs="1" maxOccurs="1"/>
  </xsd:sequence>
</xsd:complexType>

</xsd:schema>
```

## 6.2 RTC ドキュメント Profile(RtcProfile\_doc.xsd)

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

  <xsd:import namespace="http://www.openrtp.org/namespaces/rtc" schemaLocation="RtcProfile_basic.xsd"/>

  <xsd:complexType name="basic_info_doc">
    <xsd:complexContent>
      <xsd:extension base="rtc:basic_info">
        <xsd:sequence>
          <xsd:element name="Doc" type="rtcDoc:doc_basic" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
      </xsd:extension>
    </xsd:complexContent>
  </xsd:complexType>

  <xsd:complexType name="doc_basic">
    <xsd:attribute name="description" type="xsd:string" use="optional"/>
    <xsd:attribute name="inout" type="xsd:string" use="optional"/>
    <xsd:attribute name="algorithm" type="xsd:string" use="optional"/>
    <xsd:attribute name="creator" type="xsd:string" use="optional"/>
    <xsd:attribute name="license" type="xsd:string" use="optional"/>
    <xsd:attribute name="reference" type="xsd:string" use="optional"/>
  </xsd:complexType>

  <xsd:complexType name="action_status_doc">
    <xsd:complexContent>
      <xsd:extension base="rtc:action_status">
        <xsd:sequence>
          <xsd:element name="Doc" type="rtcDoc:doc_action" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
      </xsd:extension>
    </xsd:complexContent>
  </xsd:complexType>

  <xsd:complexType name="doc_action">
    <xsd:attribute name="description" type="xsd:string" use="optional"/>
    <xsd:attribute name="preCondition" type="xsd:string" use="optional"/>
    <xsd:attribute name="postCondition" type="xsd:string" use="optional"/>
  </xsd:complexType>

  <xsd:complexType name="configuration_doc">
    <xsd:complexContent>
      <xsd:extension base="rtc:configuration">
        <xsd:sequence>
          <xsd:element name="Doc" type="rtcDoc:doc_configuration" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
      </xsd:extension>
    </xsd:complexContent>
  </xsd:complexType>

  <xsd:complexType name="doc_configuration">
    <xsd:attribute name="dataname" type="xsd:string" use="optional"/>
    <xsd:attribute name="defaultValue" type="xsd:string" use="optional"/>
    <xsd:attribute name="description" type="xsd:string" use="optional"/>
    <xsd:attribute name="unit" type="xsd:string" use="optional"/>
    <xsd:attribute name="range" type="xsd:string" use="optional"/>
    <xsd:attribute name="constraint" type="xsd:string" use="optional"/>
  </xsd:complexType>
```

```

<xsd:complexType name="dataport_doc">
  <xsd:complexContent>
    <xsd:extension base="rtc:dataport">
      <xsd:sequence>
        <xsd:element name="Doc" type="rtcDoc:doc_dataport" minOccurs="0" maxOccurs="1"/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="doc_dataport">
  <xsd:attribute name="description" type="xsd:string" use="optional"/>
  <xsd:attribute name="type" type="xsd:string" use="optional"/>
  <xsd:attribute name="number" type="xsd:string" use="optional"/>
  <xsd:attribute name="semantics" type="xsd:string" use="optional"/>
  <xsd:attribute name="unit" type="xsd:string" use="optional"/>
  <xsd:attribute name="occurrence" type="xsd:string" use="optional"/>
  <xsd:attribute name="operation" type="xsd:string" use="optional"/>
</xsd:complexType>

<xsd:complexType name="serviceport_doc">
  <xsd:complexContent>
    <xsd:extension base="rtc:serviceport">
      <xsd:sequence>
        <xsd:element name="Doc" type="rtcDoc:doc_serviceport" minOccurs="0" maxOccurs="1"/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="doc_serviceport">
  <xsd:attribute name="description" type="xsd:string" use="optional"/>
  <xsd:attribute name="ifdescription" type="xsd:string" use="optional"/>
</xsd:complexType>

<xsd:complexType name="serviceinterface_doc">
  <xsd:complexContent>
    <xsd:extension base="rtc:serviceinterface">
      <xsd:sequence>
        <xsd:element name="Doc" type="rtcDoc:doc_serviceinterface" minOccurs="0" maxOccurs="1"/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="doc_serviceinterface">
  <xsd:attribute name="description" type="xsd:string" use="optional"/>
  <xsd:attribute name="docArgument" type="xsd:string" use="optional"/>
  <xsd:attribute name="docReturn" type="xsd:string" use="optional"/>
  <xsd:attribute name="docException" type="xsd:string" use="optional"/>
  <xsd:attribute name="docPreCondition" type="xsd:string" use="optional"/>
  <xsd:attribute name="docPostCondition" type="xsd:string" use="optional"/>
</xsd:complexType>

</xsd:schema>

```

### 6.3 RTC 拡張 Profile(RtcProfile\_ext.xsd)

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

  <xsd:import namespace="http://www.openrtp.org/namespaces/rtc" schemaLocation="RtcProfile_basic.xsd"/>
  <xsd:import namespace="http://www.openrtp.org/namespaces/rtc_doc" schemaLocation="RtcProfile_doc.xsd"/>

  <xsd:complexType name="basic_info_ext">
    <xsd:complexContent>
      <xsd:extension base="rtcDoc:basic_info_doc">
        <xsd:sequence>
          <xsd:element name="VersionUpLogs" type="xsd:string" minOccurs="0" maxOccurs="unbounded"/>
          <xsd:element name="Properties" type="rtcExt:property" minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:attribute name="comment" type="xsd:string" use="optional"/>
        <xsd:attribute name="saveProject" type="xsd:string" use="optional"/>
      </xsd:extension>
    </xsd:complexContent>
  </xsd:complexType>

  <xsd:complexType name="configuration_ext">
    <xsd:complexContent>
      <xsd:extension base="rtcDoc:configuration_doc">
        <xsd:sequence>
          <xsd:element name="Properties" type="rtcExt:property" minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:attribute name="comment" type="xsd:string" use="optional"/>
        <xsd:attribute name="variableName" type="xsd:string" use="optional"/>
      </xsd:extension>
    </xsd:complexContent>
  </xsd:complexType>

  <xsd:complexType name="dataport_ext">
    <xsd:complexContent>
      <xsd:extension base="rtcDoc:dataport_doc">
        <xsd:sequence>
          <xsd:element name="Properties" type="rtcExt:property" minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:attribute name="comment" type="xsd:string" use="optional"/>
        <xsd:attribute name="variableName" type="xsd:string" use="optional"/>
        <xsd:attribute name="position" type="rtcExt:position" use="required"/>
      </xsd:extension>
    </xsd:complexContent>
  </xsd:complexType>

  <xsd:complexType name="serviceport_ext">
    <xsd:complexContent>
      <xsd:extension base="rtcDoc:serviceport_doc">
        <xsd:sequence>
          <xsd:element name="Properties" type="rtcExt:property" minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:attribute name="comment" type="xsd:string" use="optional"/>
        <xsd:attribute name="position" type="rtcExt:position" use="required"/>
      </xsd:extension>
    </xsd:complexContent>
  </xsd:complexType>
</xsd:schema>
```

```

<xsd:complexType name="serviceinterface_ext">
  <xsd:complexContent>
    <xsd:extension base="rtcDoc:serviceinterface_doc">
      <xsd:sequence>
        <xsd:element name="Properties" type="rtcExt:property" minOccurs="0" maxOccurs="unbounded"/>
      </xsd:sequence>
      <xsd:attribute name="comment" type="xsd:string" use="optional"/>
      <xsd:attribute name="variableName" type="xsd:string" use="optional"/>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

<xsd:simpleType name="position">
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value="left"/>
    <xsd:enumeration value="right"/>
    <xsd:enumeration value="top"/>
    <xsd:enumeration value="bottom"/>
  </xsd:restriction>
</xsd:simpleType>

<xsd:complexType name="language_ext">
  <xsd:complexContent>
    <xsd:extension base="rtc:language">
      <xsd:sequence>
        <xsd:element name="targets" type="rtcExt:target_environment" minOccurs="1" maxOccurs="unbounded"/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="target_environment">
  <xsd:sequence>
    <xsd:element name="libraries" type="rtcExt:library" minOccurs="0" maxOccurs="unbounded"/>
  </xsd:sequence>
  <xsd:attribute name="arch" type="xsd:string" use="optional"/>
  <xsd:attribute name="compilerVersion" type="xsd:string" use="optional"/>
  <xsd:attribute name="os" type="xsd:string" use="optional"/>
  <xsd:attribute name="version" type="xsd:string" use="optional"/>
</xsd:complexType>

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

```



## 7 Appendix: YAMLスキーマ

```
type: map
mapping:
  "rtcProfile":
    required: yes
    type: map
    mapping:
      "actions":
        type: map
        required: yes
        mapping:
          "onAborting": &actions
          type: map
          mapping:
            "implemented":
              type: bool
              required: yes
            "rtcDoc::doc":
              type: map
              mapping:
                "description":
                  type: text
                "postCondition":
                  type: text
                "preCondition":
                  type: text
            "onActivated": *actions
            "onDeactivated": *actions
            "onError": *actions
            "onExecute": *actions
            "onFinalize": *actions
            "onInitialize": *actions
            "onRateChanged": *actions
            "onReset": *actions
            "onShutdown": *actions
            "onStartup": *actions
            "onStateUpdate": *actions
            "onAction": *actions
            "onModeChanged": *actions
```

```
"basicInfo":
  type: map
  required: yes
  mapping:
    "abstract":
      type: text
    "activityType":
      type: text
      required: yes
      enum:
        - PERIODIC
        - SPORADIC
        - EVENTDRIVEN
    "category":
      type: text
      required: yes
    "rtcExt::comment":
      type: text
    "componentKind":
      type: text
      required: yes
    "componentType":
      type: text
      required: yes
      enum:
        - STATIC
        - UNIQUE
        - COMMUTATIVE
```

```

"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
"description":
  type:    text
"rtcDoc::doc":
  type:    map
  mapping:
    "algorithm":
      type:    text
    "creator":
      type:    text
    "description":
      type:    text
    "inout":
      type:    text
    "license":
      type:    text
    "reference":
      type:    text
"executionRate":
  type:    number
"executionType":
  type:    text
  required: yes
"hardwareProfile":
  type:    text
"maxInstances":
  type:    int
"name":
  type:    text
  required: yes
"rtcExt::property":  &property
  type:    seq
  sequence:
    - type:    map
      mapping:
        "name":
          type:    text
          required: yes
        "value":
          type:    text

```

```

"rtcType":
  type:    text
"rtcExt::saveProject":
  type:    text
"updateDate":    *dateType
"vendor":
  type:    text
  required: yes
"version":
  type:    text
  required: yes
"rtcExt::versionUpLog":
  type:    seq
  sequence:
    - type:    text
"configurationSet":
  type:    map
  mapping:
    "configuration":
      type:    seq
      sequence:
        - type:    map
          mapping:
            "rtcExt::comment":
              type:    text
            "constraint":    &constraintRoot
              type:    map
              mapping:
                "constraintUnit":    &constraintUnit
                  type:    map
                  mapping:
                    "key":
                      type:    text
                    "not":
                      type:    map
                      mapping:
                        "constraint":    *constraintUnit
                    "and":
                      type:    seq
                      sequence:
                        - *constraintUnit
                    "or":
                      type:    seq
                      sequence:
                        - *constraintUnit
                "propertyIsBetween":
                  type:    map
                  mapping:
                    "lowerBoundary":
                      type:    text
                      required: yes
                    "upperBoundary":
                      type:    text
                      required: yes
                "propertyIsEqualTo":
                  type:    map
                  mapping:
                    "literal":
                      type:    text
                      required: yes
                    "matchCase":
                      type:    bool
                      required: yes

```

```

"propertyIsNotEqualTo":
  type: map
  mapping:
    "literal":
      type: text
      required: yes
    "matchCase":
      type: bool
      required: yes
"propertyIsLessThan":
  type: map
  mapping:
    "literal":
      type: text
      required: yes
    "matchCase":
      type: bool
      required: yes
"propertyIsGreaterThan":
  type: map
  mapping:
    "literal":
      type: text
      required: yes
    "matchCase":
      type: bool
      required: yes
"propertyIsLessThanOrEqualTo":
  type: map
  mapping:
    "literal":
      type: text
      required: yes
    "matchCase":
      type: bool
      required: yes
"propertyIsGreaterThanOrEqualTo":
  type: map
  mapping:
    "literal":
      type: text
      required: yes
    "matchCase":
      type: bool
      required: yes
"propertyIsLike":
  type: map
  mapping:
    "literal":
      type: text
      required: yes
    "escapeChar":
      type: text
      required: yes
    "singleChar":
      type: text
      required: yes
    "wildCard":
      type: text
      required: yes
"propertyIsNull":
  type: text

```

```

"constraintList":
  type: seq
  sequence:
    - *constraintUnit
"constraintHash":
  type: seq
  sequence:
    - *constraintUnit
"defaultValue":
  type: text
"rtcDoc::doc":
  type: map
  mapping:
    "constraint":
      type: text
    "dataname":
      type: text
    "defaultValue":
      type: text
    "description":
      type: text
    "range":
      type: text
    "unit":
      type: text
"name":
  type: text
  required: yes
"rtcExt::property": *property
"type":
  type: text
  required: yes
"unit":
  type: text
"rtcExt::variableName":
  type: text

"dataPorts":
  type: seq
  sequence:
    - type: map
      mapping:
        "rtcExt::comment":
          type: text
        "constraint": *constraintRoot
        "dataflowType":
          type: text
        "rtcDoc::doc":
          type: map
          mapping:
            "description":
              type: text
            "number":
              type: text
            "occurrence":
              type: text
            "operation":
              type: text
            "semantics":
              type: text
            "type":
              type: text
            "unit":
              type: text

```

```

    "idFile":
      type: text
    "interfaceType":
      type: text
    "name":
      type: text
      required: yes
    "portType":
      type: text
      required: yes
      enum:
        - DataInPort
        - DataOutPort
    "rtcExt::position": &pos
      type: text
      enum:
        - LEFT
        - RIGHT
        - TOP
        - BOTTOM
    "rtcExt::property": *property
    "subscriptionType":
      type: text
    "type":
      type: text
      required: yes
    "unit":
      type: text
    "rtcExt::variableName":
      type: text

  "id":
    type: text
    required: yes
  "language":
    type: map
    mapping:
      "kind":
        type: text
        required: yes
      "targets":
        type: seq
        required: yes
        sequence:
          - type: map
            mapping:
              "arch":
                type: text
              "compilerVersion":
                type: text
              "os":
                type: text
              "libraries":
                type: seq
                sequence:
                  - type: map
                    mapping:
                      "path":
                        type: text
                        required: yes
              "version":
                type: text
      "rtcExt::property": *property

```

```

  "parameters":
    type: seq
    sequence:
      - type: map
        mapping:
          "name":
            type: text
            required: yes
          "defaultValue":
            type: text
  "servicePorts":
    type: seq
    sequence:
      - type: map
        mapping:
          "rtcExt::comment":
            type: text
          "rtcDoc::doc":
            type: map
            mapping:
              "description":
                type: text
              "ifdescription":
                type: text
          "name":
            type: text
            required: yes
          "rtcExt::property": *property
          "rtcExt::position": *pos
          "serviceInterface":
            type: seq
            sequence:
              - type: map
                mapping:
                  "rtcExt::comment":
                    type: text
                  "direction":
                    type: text
                    required: yes
                    enum:
                      - Provided
                      - Required
                  "rtcDoc::doc":
                    type: map
                    mapping:
                      "description":
                        type: text
                      "docArgument":
                        type: text
                      "docException":
                        type: text
                      "docPostCondition":
                        type: text
                      "docPreCondition":
                        type: text
                      "docReturn":
                        type: text
          "idFile":
            type: text
          "instanceName":
            type: text

```

```
    "name":
      type: text
      required: yes
    "path":
      type: text
    "rtcExt::property": *property
    "type":
      type: text
      required: yes
    "rtcExt::variableName":
      type: text
  "transMethod":
    type: seq
    sequence:
      - type: map
        mapping:
          "kind":
            type: text
            required: yes
  "version":
    type: text
    required: yes
```

## 8 Appendix: 制約記述サンプル

---

### 8.1 XML

○定数(100)指定

```
<Constraint>
  <ConstraintUnitType>
    <PropertyIsEqualTo>
      <Literal>100</Literal>
    </PropertyIsEqualTo>
  </ConstraintUnitType>
</Constraint>
```

○定数(100)以外指定

```
<Constraint>
  <ConstraintUnitType>
    <PropertyIsNotEqualTo>
      <Literal>100</Literal>
    </PropertyIsNotEqualTo>
  </ConstraintUnitType>
</Constraint>
```

○定数(100)未満

```
<Constraint>
  <ConstraintUnitType>
    <PropertyIsLessThan>
      <Literal>100</Literal>
    </PropertyIsLessThan>
  </ConstraintUnitType>
</Constraint>
```

○定数(100)超

```
<Constraint>
  <ConstraintUnitType>
    <PropertyIsGreaterThan>
      <Literal>100</Literal>
    </PropertyIsGreaterThan>
  </ConstraintUnitType>
</Constraint>
```

○定数(100)以下

```
<Constraint>
  <ConstraintUnitType>
    <PropertyIsLessThanOrEqualTo>
      <Literal>100</Literal>
    </PropertyIsLessThanOrEqualTo>
  </ConstraintUnitType>
</Constraint>
```

○定数(100)以上

```
<Constraint>
  <ConstraintUnitType>
    <PropertyIsGreaterThanOrEqualTo>
      <Literal>100</Literal>
    </PropertyIsGreaterThanOrEqualTo>
  </ConstraintUnitType>
</Constraint>
```

○定数範囲 (100 以上 200 以下)

```
<Constraint>
  <ConstraintUnitType>
    <PropertyIsBetween>
      <LowerBoundary>100</LowerBoundary>
      <UpperBoundary>200</UpperBoundary>
    </PropertyIsBetween>
  </ConstraintUnitType>
</Constraint>
```

○大文字, 小文字区別あり文字定数 (test)

```
<Constraint>
  <ConstraintUnitType>
    <PropertyIsEqualTo>
      <Literal>test</Literal>
    </PropertyIsEqualTo>
  </ConstraintUnitType>
</Constraint>
```

○大文字, 小文字区別なし文字定数 (test)

```
<Constraint>
  <ConstraintUnitType>
    <PropertyIsEqualTo matchCase="false">
      <Literal>test</Literal>
    </PropertyIsEqualTo>
  </ConstraintUnitType>
</Constraint>
```

○書式指定文字定数 (「4 桁文字」:「3 桁文字」:「任意文字」)

```
<Constraint>
  <ConstraintUnitType>
    <PropertyIsLike wildCard="*" singleChar="." escapeChar="¥">
      <Literal>... : ... : *</Literal>
    </PropertyIsLike>
  </ConstraintUnitType>
</Constraint>
```

○Null 以外

```
<Constraint>
  <ConstraintUnitType>
    <Not>
      <Constraint>
        <ConstraintUnitType>
          <PropertyIsNull />
        </ConstraintUnitType>
      </Constraint>
    </Not>
  </ConstraintUnitType>
</Constraint>
```

○100 以下もしくは 200 以上

```
<Constraint>
  <ConstraintUnitType>
    <Or>
      <Constraint>
        <ConstraintUnitType>
          <PropertyIsLessThanOrEqualTo>
            <Literal>100</Literal>
          </PropertyIsLessThanOrEqualTo>
        </ConstraintUnitType>
      </Constraint>
      <Constraint>
        <ConstraintUnitType>
          <PropertyIsGreaterThanOrEqualTo>
            <Literal>200</Literal>
          </PropertyIsGreaterThanOrEqualTo>
        </ConstraintUnitType>
      </Constraint>
    </Or>
  </ConstraintUnitType>
</Constraint>
```

○文字列列举 ("red", "blue", "white")

```

<Constraint>
  <ConstraintUnitType>
    <Or>
      <Constraint>
        <ConstraintUnitType>
          <PropertyIsEqualTo>
            <Literal>red</Literal>
          </PropertyIsEqualTo>
        </ConstraintUnitType>
      </Constraint>
      <Constraint>
        <ConstraintUnitType>
          <PropertyIsEqualTo>
            <Literal>blue</Literal>
          </PropertyIsEqualTo>
        </ConstraintUnitType>
      </Constraint>
      <Constraint>
        <ConstraintUnitType>
          <PropertyIsEqualTo>
            <Literal>green</Literal>
          </PropertyIsEqualTo>
        </ConstraintUnitType>
      </Constraint>
    </Or>
  </ConstraintUnitType>
</Constraint>

```

○数值列举 (5, 10, 15, 20)

```

<Constraint>
  <ConstraintUnitType>
    <Or>
      <Constraint>
        <ConstraintUnitType>
          <PropertyIsEqualTo>
            <Literal>5</Literal>
          </PropertyIsEqualTo>
        </ConstraintUnitType>
      </Constraint>
      <Constraint>
        <ConstraintUnitType>
          <PropertyIsEqualTo>
            <Literal>10</Literal>
          </PropertyIsEqualTo>
        </ConstraintUnitType>
      </Constraint>
      <Constraint>
        <ConstraintUnitType>
          <PropertyIsEqualTo>
            <Literal>15</Literal>
          </PropertyIsEqualTo>
        </ConstraintUnitType>
      </Constraint>
      <Constraint>
        <ConstraintUnitType>
          <PropertyIsEqualTo>
            <Literal>20</Literal>
          </PropertyIsEqualTo>
        </ConstraintUnitType>
      </Constraint>
    </Or>
  </ConstraintUnitType>
</Constraint>

```

## 8.2 YAML

○定数 (100) 指定

```

constraint:
  constraintUnit:
    propertyIsEqualTo:
      matchCase: false
      literal: 100

```

○定数 (100) 以外指定

```

constraint:
  constraintUnit:
    propertyIsNotEqualTo:
      matchCase: false
      literal: 100

```

○定数(100)未滿

```

constraint:
  constraintUnit:
    propertyIsLessThan:
      matchCase: false
      literal: 100

```

○定数(100)超

```

constraint:
  constraintUnit:
    propertyIsGreaterThan:
      matchCase: false
      literal: 100

```

○定数 (100) 以下

```

constraint:
  constraintUnit:
    propertyIsLessThanOrEqualTo:
      matchCase: false
      literal: 100

```

○定数 (100) 以上

```

constraint:
  constraintUnit:
    propertyIsGreaterThanOrEqualTo:
      matchCase: false
      literal: 100

```



○定数範囲(100以上200以下)

```
constraint:
  constraintUnit:
    propertyIsBetween:
      lowerBoundary: 100
      upperBoundary: 200
```

○大文字, 小文字区別あり文字定数(test)

```
constraint:
  constraintUnit:
    propertyIsEqualTo:
      matchCase: true
      literal: test
```

○大文字, 小文字区別なし文字定数(test)

```
constraint:
  constraintUnit:
    propertyIsEqualTo:
      matchCase: false
      literal: test
```

○書式指定文字定数(「4桁文字」:「3桁文字」:「任意文字」)

```
constraint:
  constraintUnit:
    propertyIsLike:
      wildcard: "*"
      singleChar: "."
      escapeChar: "\\"
      literal: "... :... :*
```

○Null以外

```
constraint:
  constraintUnit:
    not:
      constraint:
        constraintUnitType:
          propertyIsNull:
```

○100以下もしくは200以上

```
constraint:
  constraintUnit:
    or:
      -
        propertyIsLessThanOrEqualTo:
          matchCase: false
          literal: 100
      -
        propertyIsGreaterThanOrEqualTo:
          matchCase: false
          literal: 200
```

○文字列列挙("red","blue","white")

```
constraint:
  constraintUnit:
    or:
      -
        propertyIsEqualTo:
          matchCase: false
          literal: red
      -
        propertyIsEqualTo:
          matchCase: false
          literal: blue
      -
        propertyIsEqualTo:
          matchCase: false
          literal: green
```

○数値列挙(5,10,15,20)

```
constraint:
  constraintUnit:
    or:
      -
        propertyIsEqualTo:
          matchCase: false
          literal: 5
      -
        propertyIsEqualTo:
          matchCase: false
          literal: 10
      -
        propertyIsEqualTo:
          matchCase: false
          literal: 15
      -
        propertyIsEqualTo:
          matchCase: false
          literal: 20
```

## 9 Appendix: サンプル

### 9.1 コンポーネント仕様

<b>RtcProfile</b>	
ID	RTC:SampleVender.SampleCategory.SampleComponent:1.0.0
version	0.2
<b>BasicInfo</b>	
name	SampleComponent
componentType	STATIC
activityType	PERIDIC
componentKind	DataFlowComponent
rtcType	Normal
category	SampleCategory
description	SampleDescription
executionRate	1000.0
executionType	PeriodicExecutionContext
maxInstances	1
vendor	SampleVender
version	1.0.0
abstract	SampleAbstract
hardwareProfil	SampleProfile
creationDate	2008-04-18T14:00:00
updateDate	2008-04-17T14:00:00
<b>doc_basic</b>	
algorithm	SampleAlgorithm
description	SampleBasicDescription
inout	SampleBasicInout
creator	SampleCreator
license	SampleLicense
reference	SampleReference
<b>VersionUpLog</b>	
	2008/04/18 14:00:00:Ver1.0
	2008/04/18 17:00:00:Ver1.1
<b>parameters</b>	
1	name param1
	defaultValue param def1
2	name param2
	defaultValue param def2

<b>actions</b>		
<b>action_status/doc_action</b>		
OnInitialize	implemented	true
	description	on initialize description
	PreCondition	on initialize Pre condition
	PostCondition	on initialize Post condition
OnFinalize	implemented	false
	description	on finalize description
	PreCondition	on finalize Pre condition
	PostCondition	on finalize Post condition
OnStartup	implemented	false
	description	on startup description
	PreCondition	on startup Pre condition
	PostCondition	on startup Post condition
OnShutdown	implemented	true
	description	on shutdown description
	PreCondition	on shutdown Pre condition
	PostCondition	on shutdown Post condition
OnActivated	implemented	true
	description	on activated description
	PreCondition	on activated Pre condition
	PostCondition	on activated Post condition
OnDeactivated	implemented	false
	description	on deactivated description
	PreCondition	on deactivated Pre condition
	PostCondition	on deactivated Post condition
OnExecute	implemented	false
	description	on execute description
	PreCondition	on execute Pre condition
	PostCondition	on execute Post condition
OnAborting	implemented	true
	description	on aborting description
	PreCondition	on aborting Pre condition
	PostCondition	on aborting Post condition
OnError	implemented	false
	description	on error description
	PreCondition	on error Pre condition
	PostCondition	on error Post condition
OnReset	implemented	false
	description	on reset description
	PreCondition	on reset Pre condition
	PostCondition	on reset Post condition
OnStateUpdate	implemented	false
	description	on state update description
	PreCondition	on state update Pre condition
	PostCondition	on state update Post condition
OnRateChanged	implemented	false
	description	on rate changed description
	PreCondition	on rate changed Pre condition
	PostCondition	on rate changed Post condition
<b>language</b>		
kind	Java	
arch	SampleArch	
compilerVersio	5.2	
os	UNIX	
path	library1	
version	1.0	

<b>configuration/doc_configuration</b>		
1	name	config1
	type	int
	defaultValue	10
	unit	config unit1
	constraint	100未満
	comment	Sample
	variableName	var1
	description	config Descl
	dataname	dataname1
	defaultValue	default1
	unit	config unit1
	range	config range1
	constraint	config constraint1
	2	name
type		String
default value		Sample
unit		(なし)
constraint		"up", "down", "left", "right"
<b>dataport/doc_dataport</b>		
1	portType	DataInPort
	name	inport1
	type	RTC: :TimedLong
	idlFile	DataPort1.idl
	interfaceType	CorbaPort
	dataflowType	Push, Pull
	subscriptionTyp	Periodic, New, Flush
	unit	IniUnit
	constraint	100以上200以下
	description	IniDescription
	type	IniType
	number	IniNumber
	semantics	IniSemantics
	unit	IniUnit
	occurrence	IniOccerrence
	operation	IniOperation
	comment	Sample
variableName	IniVar	
position	left	
2	portType	DataInPort
	name	inport2
	type	RTC: :TimedDouble
	interfaceType	CorbaPort
	dataflowType	Push, Pull
	subscriptionTyp	New, Periodic
	position	left
3	portType	DataOutPort
	name	outport1
	type	RTC: :TimedLong
	interfaceType	CorbaPort
	dataflowType	Push
	subscriptionTyp	New, Periodic
	description	Out1Description
	type	Out1Type
	number	Out1Number
	semantics	Out1Semantics
position	right	
4	portType	DataOutPort
	name	outport2
	type	RTC: :TimedDouble
	interfaceType	CorbaPort
	dataflowType	Push, Pull
	subscriptionTyp	New, Periodic
	position	right

<i>serviceport/doc_serviceport</i>		
1	name	SrvPort1
	transMethod::ki	CORBA
	description	ServicePort1 description
	ifdescription	ServicePort1 I/F description
	position	left
2	name	SrvPort2
	transMethod::ki	TCP
	position	left
	description	ServicePort2 description
	ifdescription	ServicePort2 I/F description
<i>serviceinterface/doc_serviceinterface</i>		
1	name	S1IF1
	type	IF1Type
	direction	Provided
	instanceName	IF1Instance
	idlFile	IF1Idlfile.idl
	path	IF1SearchPath
	description	if1 description
	docArgument	if1 Argument
	docReturn	if1 Return
	docException	if1 Exception
	docPreCondition	if1 PreCond
	docPostCondition	if1 PostCond
	variableName	IF1VarName
	2	name
type		IF2Type
direction		Required
instanceName		IF2Instance
path		IF2SearchPath

## 9.2 RTC.xml

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<rtc:RtcProfile rtc:version="0.2" rtc:id="RTC:SampleVender.SampleCategory.SampleComponent:1.0.0"
  xmlns:rtcExt="http://www.openrtp.org/namespaces/rtc_ext" xmlns:rtcDoc="http://www.openrtp.org/namespaces/rtc_doc"
  xmlns:rtc="http://www.openrtp.org/namespaces/rtc" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <rtc:BasicInfo xsi:type="rtcExt:basic_info_ext" rtc:updateDate="2008-04-18T14:00:00" rtc:creationDate="2008-04-18T14:00:00"
    rtc:hardwareProfile="SampleProfile" rtc:abstract="SampleAbstract" rtc:version="1.0.0" rtc:vendor="SampleVendor"
    rtc:maxInstances="1" rtc:executionType="PeriodicExecutionContext" rtc:executionRate="1000.0" rtc:description="SampleDescription"
    rtc:category="SampleCategory" rtc:rtcType="Normal" rtc:componentKind="DataFlowComponent" rtc:activityType="PERIODIC"
    rtc:componentType="STATIC" rtc:name="SampleComponent">
    <rtcDoc:Doc rtcDoc:reference="SampleReference" rtcDoc:license="SampleLicense" rtcDoc:creator="SampleCreator"
      rtcDoc:algorithm="SampleAlgorithm" rtcDoc:inout="SampleBasicInout" rtcDoc:description="SampleBasicDecription"/>
    <rtcExt:VersionUpLog>2008/04/18 14:00:00:Ver1.0.</rtcExt:VersionUpLog>
    <rtcExt:VersionUpLog>2008/04/18 17:00:00:Ver1.1.</rtcExt:VersionUpLog>
  </rtc:BasicInfo>
  <rtc:Actions>
    <rtc:OnInitialize xsi:type="rtcDoc:action_status_doc" rtc:implemented="true">
      <rtcDoc:Doc rtcDoc:preCondition="on_initialize Pre_condition" rtcDoc:postCondition="on_initialize Post_condition"
        rtcDoc:description="on_initialize description"/>
    </rtc:OnInitialize>
    <rtc:OnFinalize xsi:type="rtcDoc:action_status_doc" rtc:implemented="false">
      <rtcDoc:Doc rtcDoc:preCondition="on_finalize Pre_condition" rtcDoc:postCondition="on_finalize Post_condition"
        rtcDoc:description="on_finalize description"/>
    </rtc:OnFinalize>
    <rtc:OnStartup xsi:type="rtcDoc:action_status_doc" rtc:implemented="false">
      <rtcDoc:Doc rtcDoc:preCondition="on_startup Pre_condition" rtcDoc:postCondition="on_startup Post_condition"
        rtcDoc:description="on_startup description"/>
    </rtc:OnStartup>
    <rtc:OnShutdown xsi:type="rtcDoc:action_status_doc" rtc:implemented="true">
      <rtcDoc:Doc rtcDoc:preCondition="on_shutdown Pre_condition" rtcDoc:postCondition="on_shutdown Post_condition"
        rtcDoc:description="on_shutdown description"/>
    </rtc:OnShutdown>
    <rtc:OnActivated xsi:type="rtcDoc:action_status_doc" rtc:implemented="true">
      <rtcDoc:Doc rtcDoc:preCondition="on_activated Pre_condition" rtcDoc:postCondition="on_activated Post_condition"
        rtcDoc:description="on_activated description"/>
    </rtc:OnActivated>
    <rtc:OnDeactivated xsi:type="rtcDoc:action_status_doc" rtc:implemented="false">
      <rtcDoc:Doc rtcDoc:preCondition="on_deactivated Pre_condition" rtcDoc:postCondition="on_deactivated Post_condition"
        rtcDoc:description="on_deactivated description"/>
    </rtc:OnDeactivated>
    <rtc:OnAborting xsi:type="rtcDoc:action_status_doc" rtc:implemented="true">
      <rtcDoc:Doc rtcDoc:preCondition="on_aborting Pre_condition" rtcDoc:postCondition="on_aborting Post_condition"
        rtcDoc:description="on_aborting description"/>
    </rtc:OnAborting>
    <rtc:OnError xsi:type="rtcDoc:action_status_doc" rtc:implemented="false">
      <rtcDoc:Doc rtcDoc:preCondition="on_error Pre_condition" rtcDoc:postCondition="on_error Post_condition"
        rtcDoc:description="on_error description"/>
    </rtc:OnError>
    <rtc:OnReset xsi:type="rtcDoc:action_status_doc" rtc:implemented="false">
      <rtcDoc:Doc rtcDoc:preCondition="on_reset Pre_condition" rtcDoc:postCondition="on_reset Post_condition"
        rtcDoc:description="on_reset description"/>
    </rtc:OnReset>
    <rtc:OnExecute xsi:type="rtcDoc:action_status_doc" rtc:implemented="false">
      <rtcDoc:Doc rtcDoc:preCondition="on_execute Pre_condition" rtcDoc:postCondition="on_execute Post_condition"
        rtcDoc:description="on_execute description"/>
    </rtc:OnExecute>
    <rtc:OnStateUpdate xsi:type="rtcDoc:action_status_doc" rtc:implemented="false">
      <rtcDoc:Doc rtcDoc:preCondition="on_state_update Pre_condition" rtcDoc:postCondition="on_state_update Post_condition"
        rtcDoc:description="on_state_update description"/>
    </rtc:OnStateUpdate>
    <rtc:OnRateChanged xsi:type="rtcDoc:action_status_doc" rtc:implemented="false">
      <rtcDoc:Doc rtcDoc:preCondition="on_rate_changed Pre_condition" rtcDoc:postCondition="on_rate_changed Post_condition"
        rtcDoc:description="on_rate_changed description"/>
    </rtc:OnRateChanged>
  </rtc:Actions>
</rtc:RtcProfile>
```

```

<rtc:ConfigurationSet>
  <rtc:Configuration xsi:type="rtcExt:configuration_ext" rtcExt:variableName="var1" rtcExt:comment="Sample" rtc:unit="config_unit1"
    rtc:defaultValue="10" rtc:type="int" rtc:name="config1">
    <rtc:Constraint>
      <rtc:ConstraintUnitType>
        <rtc:propertyIsLessThan rtc:matchCase="false">
          <rtc:Literal>100</rtc:Literal>
        </rtc:propertyIsLessThan>
      </rtc:ConstraintUnitType>
    </rtc:Constraint>
    <rtc:Doc rtcDoc:constraint="config_constraint1" rtcDoc:range="config_range1" rtcDoc:unit="config_unit1"
      rtcDoc:description="config_Desc1" rtcDoc:defaultValue="default1" rtcDoc:dataname="dataname1"/>
  </rtc:Configuration>
  <rtc:Configuration xsi:type="rtcDoc:configuration_doc" rtc:defaultValue="Sample" rtc:type="String" rtc:name="config2">
    <rtc:Constraint>
      <rtc:ConstraintUnitType>
        <rtc:Or>
          <rtc:Constraint>
            <rtc:ConstraintUnitType>
              <rtc:propertyIsEqualTo rtc:matchCase="false">
                <rtc:Literal>up</rtc:Literal>
              </rtc:propertyIsEqualTo>
            </rtc:ConstraintUnitType>
          </rtc:Constraint>
          <rtc:Constraint>
            <rtc:ConstraintUnitType>
              <rtc:propertyIsEqualTo rtc:matchCase="false">
                <rtc:Literal>down</rtc:Literal>
              </rtc:propertyIsEqualTo>
            </rtc:ConstraintUnitType>
          </rtc:Constraint>
          <rtc:Constraint>
            <rtc:ConstraintUnitType>
              <rtc:propertyIsEqualTo rtc:matchCase="false">
                <rtc:Literal>left</rtc:Literal>
              </rtc:propertyIsEqualTo>
            </rtc:ConstraintUnitType>
          </rtc:Constraint>
          <rtc:Constraint>
            <rtc:ConstraintUnitType>
              <rtc:propertyIsEqualTo rtc:matchCase="false">
                <rtc:Literal>right</rtc:Literal>
              </rtc:propertyIsEqualTo>
            </rtc:ConstraintUnitType>
          </rtc:Constraint>
        </rtc:Or>
      </rtc:ConstraintUnitType>
    </rtc:Constraint>
  </rtc:ConfigurationSet>
  <rtc>DataPorts xsi:type="rtcExt:dataport_ext" rtcExt:position="left" rtcExt:variableName="In1Var" rtc:unit="In1Unit"
    rtc:subscriptionType="New,Periodic" rtc:dataflowType="Push,Pull" rtc:interfaceType="CorbaPort" rtc:idlFile="DataPort1.idl"
    rtc:type="RTC::TimedLong" rtc:name="inport1" rtc:portType="DataInPort">
    <rtc:Constraint>
      <rtc:ConstraintUnitType>
        <rtc:propertyIsBetween>
          <rtc:LowerBoundary>100</rtc:LowerBoundary>
          <rtc:UpperBoundary>200</rtc:UpperBoundary>
        </rtc:propertyIsBetween>
      </rtc:ConstraintUnitType>
    </rtc:Constraint>
    <rtc:Doc rtcDoc:operation="In1Operation" rtcDoc:occurrence="In1Occurrence" rtcDoc:unit="In1Unit"
      rtcDoc:semantics="In1Semantics" rtcDoc:number="In1Number" rtcDoc:type="In1Type" rtcDoc:description="In1Description"/>
  </rtc>DataPorts>
  <rtc>DataPorts xsi:type="rtcExt:dataport_ext" rtcExt:position="left" rtcExt:variableName="In2Var" rtc:subscriptionType="New,Periodic"
    rtc:dataflowType="Push,Pull" rtc:interfaceType="CorbaPort" rtc:type="RTC::TimedDouble" rtc:name="inport2"
    rtc:portType="DataInPort"/>

```

```

    <rtc:DataPorts xsi:type="rtcExt:dataport_ext" rtcExt:position="right" rtcExt:variableName="Out1Var"
      rtc:subscriprionType="New,Periodic" rtc:dataflowType="Push" rtc:interfaceType="CorbaPort" rtc:type="RTC::TimedLong"
      rtc:name="outport1" rtc:portType="DataOutPort"/>
    <rtcDoc:Doc rtcDoc:operation="Out1Operation" rtcDoc:occerrence="Out1Occerrence" rtcDoc:unit="Out1Unit"
      rtcDoc:semantics="Out1Semantics" rtcDoc:number="Out1Number" rtcDoc:type="Out1Type"
      rtcDoc:description="Out1Description"/>
  </rtc:DataPorts>
  <rtc:DataPorts xsi:type="rtcExt:dataport_ext" rtcExt:position="right" rtcExt:variableName="Out2Var"
    rtc:subscriprionType="New,Periodic" rtc:dataflowType="Push,Pull" rtc:interfaceType="CorbaPort" rtc:type="RTC::TimedDouble"
    rtc:name="outport2" rtc:portType="DataOutPort"/>
  <rtc:ServicePorts xsi:type="rtcExt:serviceport_ext" rtcExt:position="left" rtc:name="SrvPort1">
    <rtc:ServiceInterface xsi:type="rtcExt:serviceinterface_ext" rtcExt:variableName="IF1VarName" rtc:path="IF1SearchPath"
      rtc:type="IF1Type" rtc:idlFile="IF1Idlfile.idl" rtc:instanceName="IF1Instance" rtc:direction="Provided" rtc:name="S1IF1">
      <rtcDoc:Doc rtcDoc:docPostCondition="if1 PostCond" rtcDoc:docPreCondition="if1 PreCond"
        rtcDoc:docException="if1 Exception" rtcDoc:docReturn="if1 Return" rtcDoc:docArgument="if1 Argument"
        rtcDoc:description="if1 description"/>
    </rtc:ServiceInterface>
    <rtc:ServiceInterface xsi:type="rtcExt:serviceinterface_ext" rtc:path="IF2SearchPath" rtc:type="IF2Type" rtc:idlFile="IF2Idlfile.idl"
      rtc:instanceName="IF2Instance" rtc:direction="Required" rtc:name="S1IF2"/>
    <rtc:TransMethods rtc:kind="CORBA"/>
    <rtcDoc:Doc rtcDoc:ifdescription="ServicePort1 I/F description" rtcDoc:description="ServicePort1 description"/>
  </rtc:ServicePorts>
  <rtc:ServicePorts xsi:type="rtcExt:serviceport_ext" rtcExt:position="left" rtc:name="SrvPort2">
    <rtc:TransMethods rtc:kind="TCP"/>
    <rtcDoc:Doc rtcDoc:ifdescription="ServicePort2 I/F description" rtcDoc:description="ServicePort2 description"/>
  </rtc:ServicePorts>
  <rtc:Parameters rtc:defaultValue="param_def1" rtc:name="param1"/>
  <rtc:Parameters rtc:defaultValue="param_def2" rtc:name="param2"/>
  <rtc:Language xsi:type="rtcExt:language_ext" rtc:kind="Java">
    <rtcExt:targets rtcExt:version="1.0" rtcExt:os="UNIX" rtcExt:compilerVersion="5.2" rtcExt:arch="SampleArch">
      <rtcExt:libraries rtcExt:path="library1"/>
    </rtcExt:targets>
  </rtc:Language>
</rtc:RtcProfile>

```



### 9.3 YAML 形式

```

---
rtcProfile:
  actions:
    onAborting:
      "rtcDoc::doc":
        description: on_aborting description
        postCondition: on_aborting Post_condition
        preCondition: on_aborting Pre_condition
        implemented: true
    onActivated:
      "rtcDoc::doc":
        description: on_activated description
        postCondition: on_activated Post_condition
        preCondition: on_activated Pre_condition
        implemented: true
    onDeactivated:
      "rtcDoc::doc":
        description: on_deactivated description
        postCondition: on_deactivated Post_condition
        preCondition: on_deactivated Pre_condition
        implemented: false
    onError:
      "rtcDoc::doc":
        description: on_error description
        postCondition: on_error Post_condition
        preCondition: on_error Pre_condition
        implemented: false
    onExecute:
      "rtcDoc::doc":
        description: on_execute description
        postCondition: on_execute Post_condition
        preCondition: on_execute Pre_condition
        implemented: false
    onFinalize:
      "rtcDoc::doc":
        description: on_finalize description
        postCondition: on_finalize Post_condition
        preCondition: on_finalize Pre_condition
        implemented: false
    onInitialize:
      "rtcDoc::doc":
        description: on_initialize description
        postCondition: on_initialize Post_condition
        preCondition: on_initialize Pre_condition
        implemented: true
    onRateChanged:
      "rtcDoc::doc":
        description: on_rate_changed description
        postCondition: on_rate_changed Post_condition
        preCondition: on_rate_changed Pre_condition
        implemented: false
    onReset:
      "rtcDoc::doc":
        description: on_reset description
        postCondition: on_reset Post_condition
        preCondition: on_reset Pre_condition
        implemented: false
    onShutdown:
      "rtcDoc::doc":
        description: on_shutdown description
        postCondition: on_shutdown Post_condition
        preCondition: on_shutdown Pre_condition
        implemented: true
    onStartup:
      "rtcDoc::doc":
        description: on_startup description
        postCondition: on_startup Post_condition
        preCondition: on_startup Pre_condition
        implemented: false
    onStateUpdate:
      "rtcDoc::doc":
        description: on_state_update description
        postCondition: on_state_update Post_condition
        preCondition: on_state_update Pre_condition
        implemented: false
  basicInfo:
    abstract: SampleAbstract
    activityType: PERIODIC
    category: SampleCategory
    "rtcExt::comment": SampleComment
    componentKind: DataFlowComponent
    componentType: STATIC
    createDate:
      day: 18
      hour: 14
      minute: 0
      month: 4
      second: 0
      year: 2008
    description: SampleDescription
    "rtcDoc::doc":
      algorithm: SampleAlgorithm
      creator: SampleCreator
      description: SampleBasicDescription
      inout: SampleBasicInout
      license: SampleLicense
      reference: SampleReference
    executionRate: 1000.0
    executionType: PeriodicExecutionContext
    hardwareProfile: SampleHardwareProfile
    maxInstances: 1
    name: SampleComponent
    "rtcExt::property":
      -
        name: SampleName
        value: SampleValue
    rtcType: SampleRtcType
    "rtcExt::saveProject": SampleSaveProject
    updateDate:
      day: 17
      hour: 14
      minute: 0
      month: 4
      second: 0
      year: 2008
    vendor: SampleVendor
    version: 1.0.0
    "rtcExt::versionUpLog":
      - "2008/04/18 14:00:00:Ver1.0"
      - "2008/04/18 17:00:00:Ver1.1"
  configurationSet:
    configuration:
      -
        "rtcExt::comment": SampleComment
        constraint:
          constraintUnit:
            propertyIsLessThan:
              matchCase: true
              literal: 100
          defaultValue: "10"
          unit: config_unit1
        "rtcDoc::doc":
          constraint: config_constraint1
          dataname: dataname1
          defaultValue: default1
          description: config_Desc1
          range: config_range1
          unit: config_unit1

```

```

name: config1
"rtcExt::property":
-
  name: SampleName
  value: SampleValue
  type: int
  "rtcExt::variableName": var1
-
constraint:
  constraintUnit:
    or:
      -
        propertyIsEqualTo:
          matchCase: true
          literal: "up"
      -
        propertyIsEqualTo:
          matchCase: true
          literal: "down"
      -
        propertyIsEqualTo:
          matchCase: true
          literal: "left"
      -
        propertyIsEqualTo:
          matchCase: true
          literal: "right"
  defaultUnit: Sample
  name: config2
  type: String
dataPorts:
-
  "rtcExt::comment": SampleComment
  constraint:
    constraintUnit:
      propertyIsBetween:
        lowerBoundary: 100
        upperBoundary: 200
  dataflowType: "Push,Pull"
  "rtcDoc::doc":
    description: In1Description
    number: In1Number
    occurrence: In1Occurrence
    operation: In1Operation
    semantics: In1Semantics
    type: In1Type
    unit: In1Unit
  idlFile: DataPort1.idl
  interfaceType: CorbaPort
  name: inport1
  portType: DataInPort
  "rtcExt::position": &58 LEFT
  subscriptionType: "Periodic,New,Flush"
  type: "RTC::TimedLong"
  unit: SampleUnit
  "rtcExt::variableName": In2Var
-
  dataflowType: "Push,Pull"
  interfaceType: CorbaPort
  name: inport2
  portType: DataInPort
  "rtcExt::position": *58
  subscriptionType: "New,Periodic"
  type: "RTC::TimedDouble"

```

```

-
  dataflowType: Push
  "rtcDoc::doc":
    description: Out1Description
    number: Out1Number
    semantics: Out1Semantics
    type: Out1Type
  interfaceType: CorbaPort
  name: outport1
  portType: DataOutPort
  "rtcExt::position": &62 RIGHT
  subscriptionType: "New,Periodic"
  type: "RTC::TimedLong"
-
  dataflowType: "Push,Pull"
  interfaceType: CorbaPort
  name: outport2
  portType: DataOutPort
  "rtcExt::position": *62
  subscriptionType: "New,Periodic"
  type: "RTC::TimedDouble"
id: "RTC:SampleVendor.SampleCategory.SampleComponent:1.0.0"
language:
  kind: Java
  targets:
  -
    arch: SampleArch
    compilerVersion: SampleCompilerVersion
    os: SampleOS
    libraries:
    -
      path: library1
      version: SampleVersion
parameters:
-
  defaultValue: param_def1
  name: param1
-
  defaultValue: param_def2
  name: param2
servicePorts:
-
  "rtcDoc::doc":
    description: ServicePort1 description
    ifdescription: ServicePort1 I/F description
  name: SrvPort1
  "rtcExt::position": *58
  transMethod:
  -
    kind: CORBA
  serviceInterface:
  -
    direction: Provided
  "rtcDoc::doc":
    description: if1 Description
    docArgument: if1 Argument
    docException: if1 Exception
    docPostCondition: if1 PostCond
    docPreCondition: if1 PreCond
    docReturn: if1 Return
  idlFile: IF1Idlfile.idl
  instanceName: IF1Instance
  name: S1IF1
  path: IF1SearchPath
  type: IF1Type
  "rtcExt::variableName": IF1VarName

```

```
-  
  direction: Required  
  idlFile: IF2Idlfile.idl  
  instanceName: IF2Instance  
  name: S1IF2  
  path: IF2SearchPath  
  type: IF2Type  
-  
  "rtcDoc::doc":  
    description: ServicePort2 description  
    ifdescription: ServicePort2 I/F description  
    name: SrvPort2  
    "rtcExt::position": *62  
    transMethod:  
  -  
    kind: TCP  
  version: "0.2"
```

## 10 Appendix: 仕様記述利用例

ここでは、RTC仕様記述方式で作成された仕様の実際の使用例を示す。

### 10.1 RTC 仕様記述の利用例

RTC仕様記述方式で記述されたRTC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、RTC仕様記述はRTシステムを構成する最も基本的な構成要素であるため、ツールチェーンの出発点となるデータフォーマットである。

図 4にRTC仕様ファイルを利用するツール群の相互関係を示す。

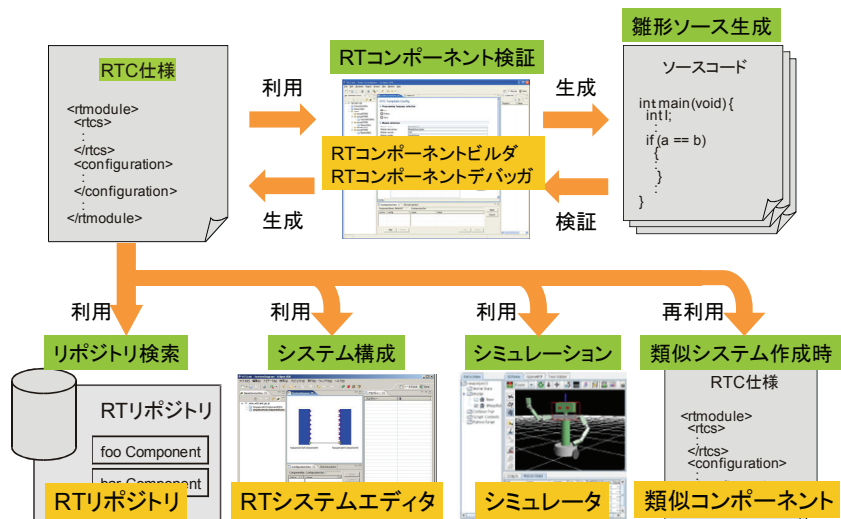


図 4 OpenRTP ツールチェーンにおける RTC 仕様の利用

## 10.2 RTC 仕様記述フォーマットとしての利用

システム設計時に、機能要素をモジュール分割し各要素の詳細設計を行う際の、各モジュールの記述方式として利用する方法が考えられる。たとえば、UMLでコンポーネントのモデルを記述し、その設計情報をXML形式に変換し出力することで後述のコード生成の情報として利用する方法などが考えられる。

産総研が提供するRTC設計ツール「RTCビルダ」はEclipse上でモジュールの仕様を入力することで、XML形式のRTC仕様ファイルを生成することができる。RTCビルダではさらにこのXMLファイルから、OpenRTM-aist用のRTCひな型コードを生成することができる。

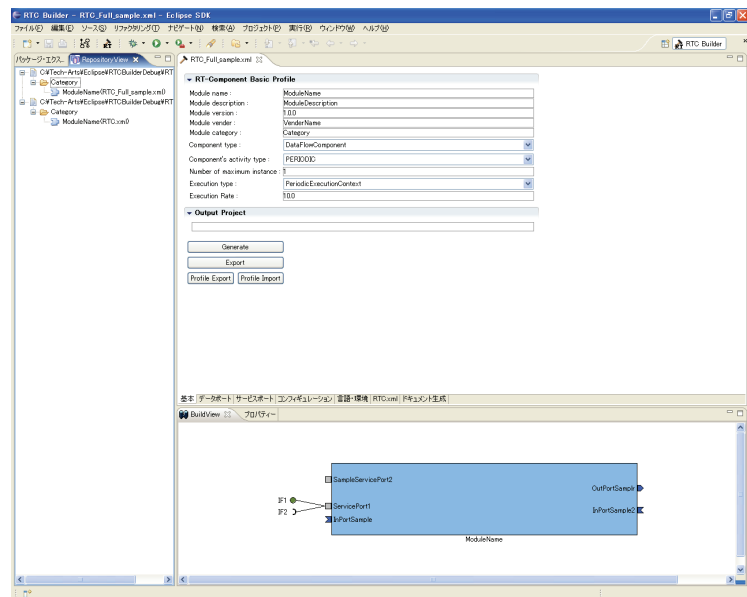


図 5 RTC ビルダの入力画面

## 10.3 コード生成のための情報として利用

RTC仕様記述ファイルにはRTCのモデルに必要な情報がすべて含まれているため、この情報を元にRTCのひな型コードを生成することができる。新たな言語に対応したRTミドルウェアが実装された場合でも、RTCモデルは共通であるので、この仕様記述方式に対応した新たなコードジェネレータを実装するだけで、既存のRTC仕様から新言語に対応したコード生成を容易に行うことができる。

本仕様記述方式はOMG RTC Specification の PIM で定義されている要素をすべて含んでいる。したがって、OMG RTC Specification の PIM に基づき、CORBA以外のプラットフォーム上に構築されたミドルウェアが実装された場合でも、本仕様記述方式で記述されたRTCのモデルは有効であり、コードジェネレータなども容易に実装可能である。

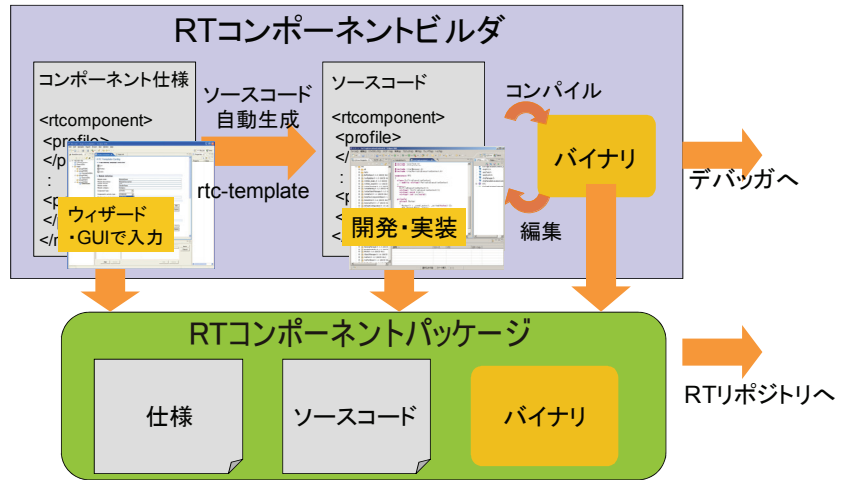


図6 RTC ビルダによるコード生成と開発  
産総研が提供するRTCビルダでは、RTCの仕様入力によるRTC仕様記述ファイル（XML形式）の生成と同時に、C++、Java、Python用OpenRTM-aistのコード生成を行うことができる（図6）。

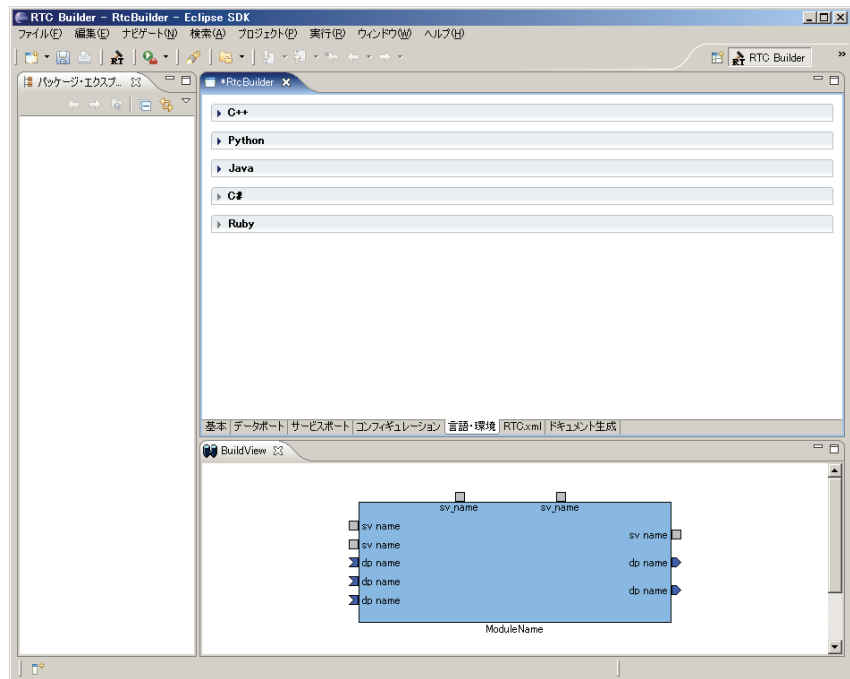


図7 RTC ビルダの言語選択画面  
RTCビルダは、RTC仕様記述に基づき、OpenRTM-aistのC++、JavaおよびPythonのRTコンポーネントのひな型コードを生成することができる（図7）。

## 10.4 システム構成時に利用

RTC仕様を利用して、システム構成を行うことができる。図はRTシステムエディタのオフラインエディタ画面である。オフラインエディタでは、実際にコンポーネントが起動していない状態でも、RTC仕様を読み込みエディタ

インタ画面にRTCを表示、RTCを接続しシステムを構築することができる。RTC仕様には、システム構築に必要なRTCプロファイルの情報が含まれているので、エディタにはRTCのデータポートやサービスポートの数やそれぞれのプロファイル情報とともに、RTCをアイコンで表示し、ポートの接続や、コンフィギュレーションパラメータの設定などの操作を行い、希望するシステム構成をオフラインで編集することが可能である。

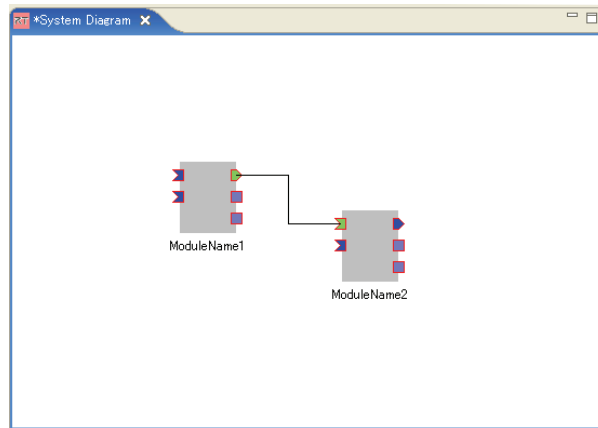


図8 オフラインシステムエディタの編集画面

## 10.5 RTC リポジトリにおける検索情報として利用

RTCリポジトリは、RTCをサーバ上に蓄積し、必要な時に必要なコンポーネントを検索・ダウンロード・配置を行うための機能を提供するサーバである。RTCリポジトリにおいては、各コンポーネントはRTC仕様に基づいて登録・管理・分類される。

RTC仕様とともに登録されたコンポーネントは、利用者がコンポーネントを検索する際に入力する検索キーとマッチングを行い、リポジトリが適切なコンポーネントを利用者に提示する。

また、システム運用時には、コンポーネントの配置およびコンフィギュレーションを行うある種のアプリケーションプログラムが、システム仕様記述ファイルを解釈し、コンポーネントを選択しリポジトリサーバから適切なコンポーネントをダウンロード・配置・設定してシステムを起動する。適切なコンポーネントを選択する際にはRTC仕様が利用される。

## 10.6 既存仕様の再利用

すでに存在するコンポーネントと類似のコンポーネントを作成する際には、既存のRTC仕様を参照することで、より再利用性の高いコンポーネントを作成することができる。

例えば、ある種のセンサをコンポーネントしたものがすでにあり、そのデータポート、サービスポート、コンフィギュレーションパラメータなどが再利用性の高い設計となっているとする。同種のより高性能なセンサが発売された場合、既存のセンサコンポーネントと同等の仕様でコンポーネントを作成することにより、すでにこのセンサコンポーネントが利用されているシステムにおいて、新たなセンサコンポーネントに置き換えることが容易にできるようになる。

## 10.7 ドキュメントとして利用

モジュールの再利用性を向上させるには、インターフェースの適切な定義と、十分なドキュメントを提供することが肝要である。モジュールの開発者と利用者が異なる場合や、モジュールのソースコードが提供されていない場合などを含め、モジュール利用者がモジュールの内部を詳細に調べなくても利用できるだけの十分な情報を提供することで、モジュールの再利用性は向上する。

RTC仕様記述方式にはプロファイル情報を記述するRTC **Basic Profile**とともに、RTC **Basic Profile** には記述しきれないセマンティックな情報を含めてRTCのドキュメント記述を支援する RTC **Document Profile** が定義されている。たとえば`doc_baisc::algorithm`には、そのコンポーネントが提供する機能やそれを実現するアルゴリズムを記述することにより、利用者に対してモジュールがどういったアルゴリズムに基づき動作するかといった意図を伝える。また、`doc_action::{description, precondition, postCondition}`はそれぞれ、コンポーネントのLifecycle Stateにおけるそれぞれのアクションでどういった動作が行われるか、そのときの事前条件・事後条件といった意図を記述する。データポートやコンフィギュレーションパラメータのBasic Profileではそれらの名前や型に関する情報が定義されるが、Document Profile ではそれらのデータの単位や意味といった情報を記述することができる。このように、RTC仕様記述方式では、モジュール再利用のために利用者に対して必要かつ十分な情報を提供するための要素が数多く定義されている。

Document Profile を含むRTC仕様のXMLファイルを、適当なテキストプロセッサに通すことで、人間が読みやすい形式に変換することも可能である。たとえば、RTC仕様XMLファイルを、ドキュメント生成ツールであるdoxygenフォーマットに変換することで、HTMLやLaTeXあるいはman形式に変換しモジュール利用者に提供するなどの利用法も考えられる。



## 11 参考文献

---

### 11.1 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](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>