SimuLike マニュアル

最終更新日：2010/1/14

目次

1 SimuLike について

SimuLike は，異なる機能を持つコンポーネント群である．以下の 16 個のコンポーネントから成る．

- Converter コンポーネント
- Multiplexer コンポーネント
- DeMultiplexer コンポーネント
- Duplicator コンポーネント
- Selecter コンポーネント
- Gainer コンポーネント
- Summater コンポーネント
- Constant コンポーネント
- Integrator コンポーネント
- Derivativer コンポーネント
- Limitter コンポーネント
- AND コンポーネント
- OR コンポーネント
- NOT コンポーネント
- FileReader コンポーネント
- FileWriter コンポーネント

2 動作環境等

2.1 動作環境について

本コンポーネントが動作を確認している動作環境は以下の通りである．

OS	Debian GNU/Linux 4.0r5 + ART-Linux 2.6.18
RT ミドルウェア	OpenRTM-aist 0.4.2 C++ 版
コンパイラ	gcc 4.1.2
依存ライブラリ	OpenRTM-aist に関するもの

本コンポーネントはソースファイルとして配布している．利用する際は，適宜コンパイルを行ってほしい．

また，本コンポーネントは ART 環境で使用することを前提として作成されている．ART 環境でのコンポーネントの利用方法は OpenRTM の公式サイトを参照 [1].

2.2 ビルドに関して

特に特殊な操作は必要ない．ファイルを展開したディレクトリに移動し，`make` を行う．

例：Converter の場合

```
make -f Makefile.Converter
```

3　すべてのコンポーネントに共通する仕様

まず，すべてのコンポーネントに共通する仕様について述べる．どのコンポーネントも RTCLink 上に配置した時点ではデータポートを持たず，ConfigurationView，もしくはサービスポートを使ってポートのデータ型を指定してから Activate することでデータポートが生成される．多入力，もしくは多出力を持つコンポーネントについてはポートの数も指定する必要がある（図 1 参照）．指定できるデータ型はコンポーネントによって異なるが，だいたい short, long, float, double, boolean と各配列の 10 通りが用意されている．また，指定できるポート数の最大は 10 個になっている．Activate 後に Deactivate し，データ型やポート数を変更して再度 Activate しても，変更は反映されない．データ型やポート数を変更する際は，必ず Finalize を行いコンポーネントを一度 RTCLink 上から削除する必要がある．

コンポーネントによっては，データポートのパラメータ以外にも設定する項目がある．それについては次章から詳しく説明する．

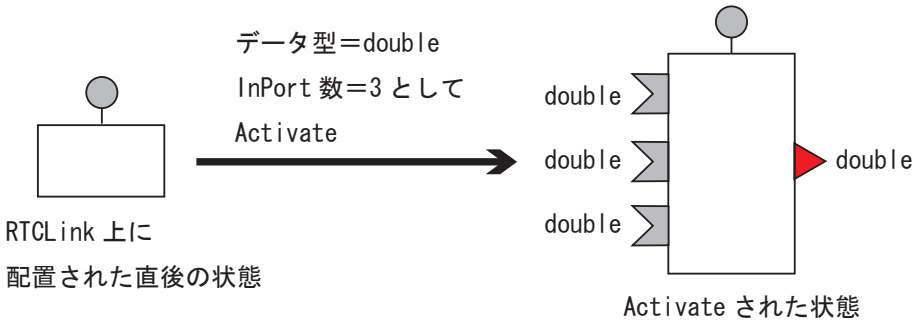

図 1　ポート生成の例

4　Converter コンポーネント

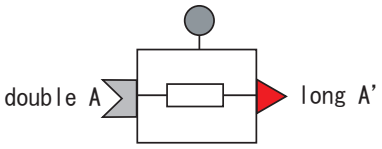

図 2　Converter コンポーネント

Converter コンポーネント（図 2）は，1 入力・1 出力のコンポーネントである．InPort から入ってきたデータを，データ型を変換して OutPort から出力することができる．指定できるデータ型は short, long, float, double, boolean と各配列の 10 通りになっており，InPort 側と OutPort 側でそれぞれ指定する必要がある．

変数から配列，また配列から変数への変換も可能である．変数から配列への変換の場合，配列の要素数はあらかじめ Activate 前に ConfigurationView 内の Extra_num で設定した数になり，また配列の中身はすべて同一の数値になる．配列から変数への変換の場合，あらかじめ Activate 前に Extra_num で配列の何番目を使用するか設定し，その数値が出力の変数になる．Extra_num に設定した数値が入力配列の要素数よりも大きかった場合，コマンドラインに”DataLength ERROR.”と表示し，コンポーネントはエラー状態になる．

5　Multiplexer コンポーネント, DeMultiplexer コンポーネント

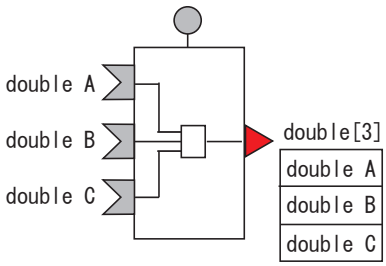

図 3　Multiplexer コンポーネント

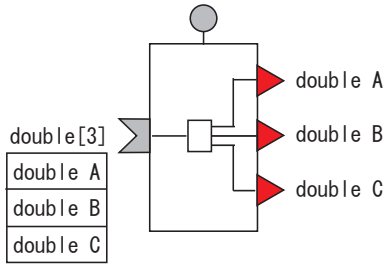

図 4　DeMultiplexer コンポーネント

Multiplexer コンポーネント（図 3）は，多入力，1 出力のコンポーネントである．複数の InPort から入ってきた個々の変数を，1 つの OutPort から配列として出力することができる．

DeMultiplexer コンポーネント（図 4）は，1 入力，多出力のコンポーネントである．1 つの InPort から入ってきた配列を，複数の OutPort から個々の変数として出力することができる．

指定できるデータ型は short, long, float, double, boolean の 5 通りになっている（In,Out のどちらかが配列であるのは明確なので，配列が含まれることは指定しなくてもいい）．

Multiplexer は各周期で各 InPort を監視，新しいデータが来ていればそれを内部に保存する．その後，すべての InPort からの新しいデータがそろっていればそれを結合して配列として OutPort に流し，保存したデータはリセットされる．そろっていなければ，保存したデータはそのままにその周期は終了する．出力される配列の要素数は InPort の数と同じになる．

DeMultiplexer は各周期で InPort を監視，新しい配列が来ていればそれを分解して変数として OutPort に流し，来ていなければ何もせずにその周期は終了する．ただし，指定した OutPort の数と入力配列の要素数が一致している必要があり，一致していない場合はコマンドラインに”data.length ERROR.”と表示し，その周期は終了する．

6　Duplicator コンポーネント

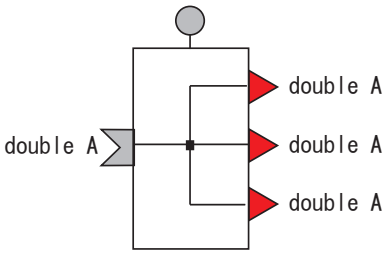

図 5　Duplicator コンポーネント

Duplicator コンポーネント（図 5）は，1 入力・多出力のコンポーネントである．1 つの InPort から入ってきたデータを，複数の OutPort から出力することができる．

指定できるデータ型は short, long, float, double, boolean と各配列の 10 通りになっている．

7　Selecter コンポーネント

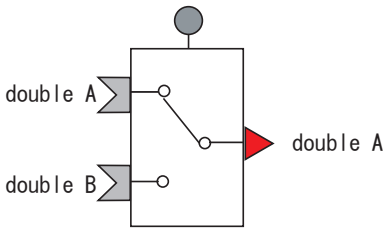

図 6　Selecter コンポーネント

Selecter コンポーネント（図 6）は，多入力，1 出力のコンポーネントである．選択された InPort から入ってきたデータを，1 つの OutPort から出力することができる．

指定できるデータ型は short, long, float, double, boolean と各配列の 10 通りになっている．

SelecterIIIO は各周期で各 InPort を監視，選択された InPort に新しいデータが来ていればそれを OutPort に流し，来ていなければ何もせずにその周期は終了する．

ポートの選択は Activate 前に一度 ConfigurationView 内の Select_Num で行うことになるが，その後もサービスポートを通じて現在選択されているポートを取得したり，また異なるポートを選択することができる．

8　Gainer コンポーネント

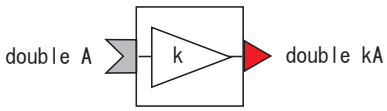

図 7　Gainer コンポーネント

Gainer コンポーネント（図 7）は，1 入力・1 出力のコンポーネントである．InPort から入ってきたデータを，定数倍して OutPort から出力することができる．

指定できるデータ型は short, long, float, double と各配列の 8 通りになっている．ゲイン値は double 型であり，Activate 前に一度 ConfigurationView 内の Gain で設定することになるが，その後もサービスポートを通じて現在のゲイン値を取得したり，また新たなゲインを設定することができる．

9　Summater コンポーネント

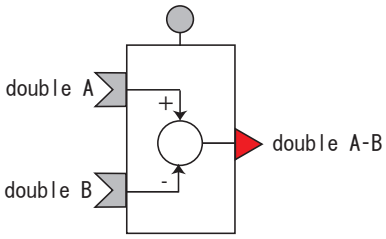

図 8　Summater コンポーネント

Summater コンポーネント（図 8）は，多入力・1 出力のコンポーネントである．複数の InPort から入ってきたデータを加減算し，結果を OutPort から出力することができる．配列は各要素ごとに加減算される．

指定できるデータ型は short, long, float, double と各配列の 8 通りになっている．Activate 前にデータ型と InPort の数だけでなく，＋として接続されるポート数も指定する必要がある．例えば，データ型を double，InPort の数を 4，＋の数を 2 と指定して Activate した場合，生成される InPort の名前は上から順に doublein0+，doublein1+, doublein0-, douoblein1-となり，上から 2 つが＋として接続されるポートに，それ以外がーとして接続されるポートになる．

Summater は各周期ごとに各 InPort を監視，新しいデータが来ていればそれを内部に保存する．その後，すべての InPort からの新しいデータがそろっていればそれを加減算して OutPort に流し，保存したデータはリセットされる．そろっていなければ，保存したデータはそのままにその周期は終了する．

要素数が異なる配列を InPort に繋ぐこともできる．その場合，出力配列の要素数は入力側で最大の要素数と同じになり，入力配列で要素数が足りないものは，足りない部分に 0 が入っているとして計算される．

10　Constant コンポーネント

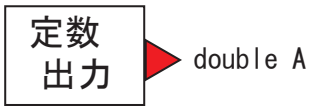

図 9　Constant コンポーネント

Constant コンポーネント（図 9）は，無入力・1 出力のコンポーネントである．設定された値を OutPort から出力し続けることができる．

指定できるデータ型は short, long, float, double の 4 通りになっている．出力する数は Activate 前に一度 ConfigurationView 内の Output で設定することになるが，その後もサービスポートを通じて現在の値を取得したり，また異なる値を設定することができる．ConfigurationView 内には他にも Start_Flag というパラメータが存在しており，これを 1 に変更することでデータの出力が開始される．これも，サービスポートから出力の開始・終了を切り替えることができる．

11　Derivativer コンポーネント, Integrator コンポーネント

図 10　Derivativer コンポーネント

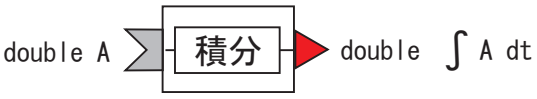

図 11　Integrator コンポーネント

Derivativer（図 10），Integrator コンポーネント（図 11）は，1 入力・1 出力のコンポーネントである．1 つの InPort から入ってきたデータをそれぞれ微分，積分して，1 つの OutPort から出力する．

指定できるデータ型は short, long, float, double と各配列の 8 通りになっている．ConfigurationView 内に ART_Cycle というパラメータが存在するが，これはコンポーネントの実行周期を秒単位で記述するためのものである．ただし，ここで周期を変更したからといって，コンポーネントの実行周期が変更されるわけではない．あくまでコンポーネント内部の計算に使われる数値である．

12　Limitter コンポーネント

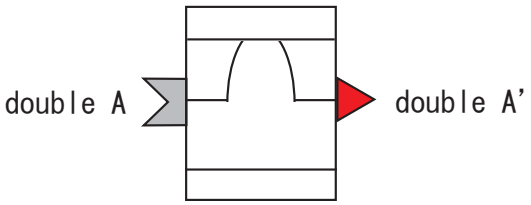

図 12　Limitter コンポーネント

Limiter コンポーネント（図 12）は，1 入力・1 出力のコンポーネントである．1 つの InPort から入ってきたデータを設定された上下限と照らし合わせ，超過していればそれを修正して OutPort から出力する．

指定できるデータ型は short, long, float, double と各配列の 8 通りになっている．データの上下限は一度 ConfigrationView 内の Limit_MAX・Limit_MIN で設定する必要があるが，その後もサービスポートを通じて現在の上下限を取得したり，また新たな上下限を設定することができる．

13　And コンポーネント, Or コンポーネント, Not コンポーネント

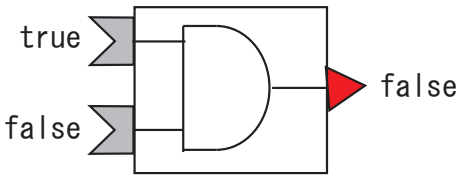

図 13　And コンポーネント

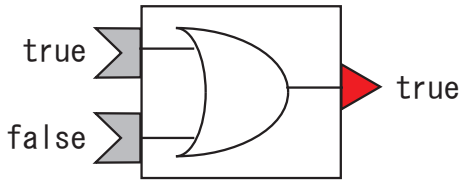

図 14　Or コンポーネント

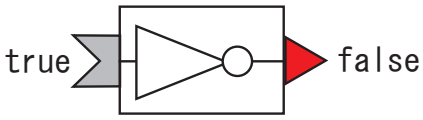

図 15　Not コンポーネント

And（図 13），Or（図 14），Not（図 15）コンポーネントは，多入力・1 出力のコンポーネントである（Not コンポーネントのみ 1 入力）．InPort から入ってきた boolean 型のデータに対して，それぞれ And，Or，Not の演算を行い，結果を OutPort から出力する．

指定できるデータ型は boolean のみになっている．And，Or は各周期ごとに各 InPort を監視，新しいデータが来ていればそれを内部に保存する．その後，すべての InPort からの新しいデータがそろっていればそれを処理して OutPort に流し，保存したデータはリセットされる．そろっていなければ，保存したデータはそのままにその周期は終了する．

14　FileReader コンポーネント

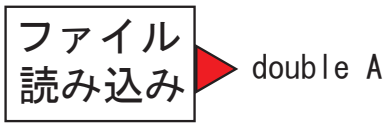

図 16　FileReader コンポーネント

FileReader コンポーネントは，無入力・1 出力のコンポーネントである．ファイルに記述されているデータを，毎周期ごとに OutPort から出力する．

指定できるデータ型は short, long, float, double と各配列の 8 通りになっている．データを取得するファイル名は，ConfigurationView 内の FilrName や，サービスポートから設定できる．図 17 に示すように，ファイルにデータが複数列に渡って記述されている場合，OutPort に配列を選択していればそれらのデータを配列として出力することができる．その場合，配列の要素数を ConfigurationView 内の Array_Size か，サービスポートで設定する必要がある．ConfigurationView 内には他にも Start_Flag というパラメータが存在しており，これを 1 に変更することでデータの出力が開始され，0 に変更することで出力が停止する．これも，サービスポートから出力の開始・終了を選択することができる．

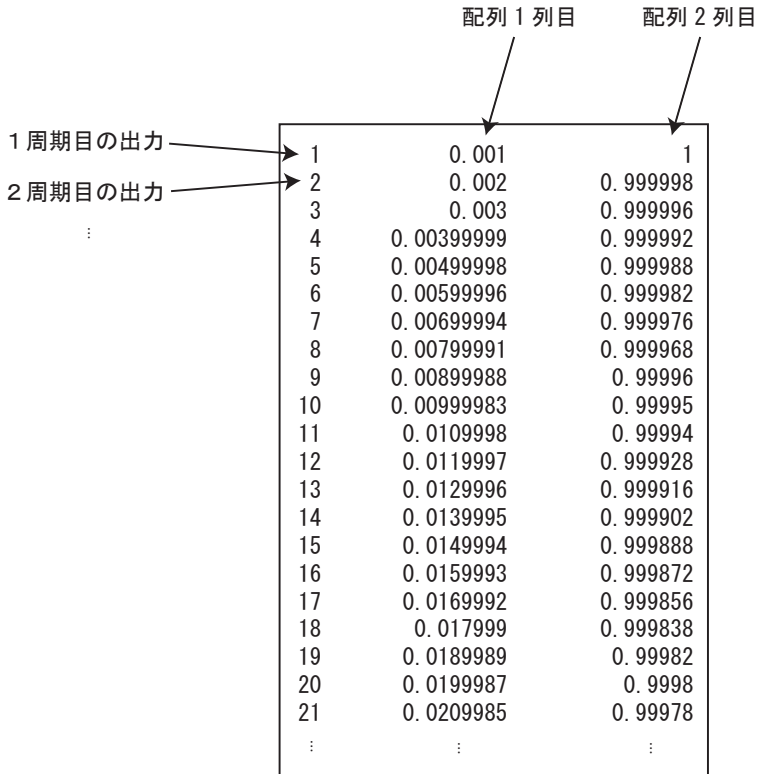

図 17　データファイルの記述例

15　FileWriter コンポーネント

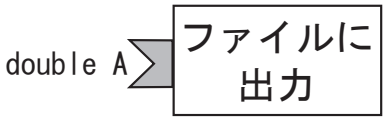

図 18　FileWriter コンポーネント

FileWriter コンポーネントは，1 入力・無出力のコンポーネントである．InPort から入力されたデータを配列に保存しながら一定時間取得し続け，一定時間後にファイルに書き出す．

指定できるデータ型は short, long, float, double と各配列の 8 通りになっている．データを取得する時間は ConfigurationView やサービスポートで設定が可能である．また，書き出すファイル名も ConfigurationView で設定できる．ConfigurationView 内には他にも Start_Flag というパラメータが存在しており，これを 1 に変更することでデータの取得が開始され，設定した時間の終了後にデータがファイルに書き出される．これも，サービスポートから取得の開始・終了を選択することができる．

ConfigurationView 内に ART_Cycle というパラメータが存在するが，これはコンポーネントの実行周期を秒単位で記述するためのものである．ただし，ここで周期を変更したからといって，コンポーネントの実行周期が変更されるわけではない．あくまでコンポーネント内部の計算に使われる数値である．

参考文献

[1] ARTLinux 用実行コンテキスト -OpenRTM-aist-： http://www.is.aist.go.jp/rt/OpenRTM-aist/html/TIPS2FARTLinuxE794A8E5AE9FE8A18CE382B3E383B3E38386E382ADE382B9E38388.html, 2010/01/14 閲覧