**RTWファイルからの制御周期抽出**

**使用説明書**

株式会社NEC情報システムズ

2015年12月21日

**目次**

[1. はじめに 2](#_Toc437444322)

[2. ソフトウェア構成 3](#_Toc437444323)

[2.1. 実行環境 4](#_Toc437444324)

[2.2. ファイル一覧 5](#_Toc437444325)

[3. ビルド手順 6](#_Toc437444326)

[3.1. ビルド準備 6](#_Toc437444327)

[3.1.1. JDK8のインストール 6](#_Toc437444328)

[3.1.2. Apache Antのインストール 6](#_Toc437444329)

[3.1.3. CUPの配置 6](#_Toc437444330)

[3.1.4. SimulinkModel.xsdの配置 6](#_Toc437444331)

[3.2. ビルド 6](#_Toc437444332)

[3.3. javadocの生成(オプショナル) 6](#_Toc437444333)

[4. 実行 7](#_Toc437444334)

[4.1. GetRateコマンド仕様(javaコマンド使用時) 7](#_Toc437444335)

[4.2. GetRate.shコマンド仕様(shellスクリプト使用時) 7](#_Toc437444336)

[4.3. SetRateコマンド仕様(javaコマンド使用時) 8](#_Toc437444337)

[4.4. SetRate.shコマンド仕様(shellスクリプト使用時) 8](#_Toc437444338)

[5. モジュール仕様 9](#_Toc437444339)

[5.1. GetRateのクラス 9](#_Toc437444340)

[5.1.1. GetRateクラス 10](#_Toc437444341)

[5.2. SetRateのクラス 11](#_Toc437444342)

[5.2.1. SetRateクラス 14](#_Toc437444343)

[5.2.2. Rateクラス 16](#_Toc437444344)

[5.2.3. BlockInfoクラス 19](#_Toc437444345)

**図目次**

[図 2‑1 プログラムと入出力ファイル 3](#_Toc437438506)

[図 2‑2　ディレクトリ構成 5](#_Toc437438507)

**表目次**

[表 2‑1 ソフトウェア構成 3](#_Toc437438508)

[表 2‑2 ツール・ライブラリ一覧 4](#_Toc437438509)

[表 5‑1 GetRateクラス 9](#_Toc437438510)

[表 5‑2 SetRateクラス 11](#_Toc437438511)

[表 5‑3 setRateクラス内クラスの一覧 12](#_Toc437438512)

**参考資料**

|  |  |  |  |
| --- | --- | --- | --- |
| 項番 | 仕様書名 | 版数 | 発行元 |
|  |  |  |  |
|  |  |  |  |

**用語一覧**

|  |  |  |
| --- | --- | --- |
| 項番 | 用語 | 意味 |
| 1 | ブロックレベル構造XML  BLXML | Simulinkモデルから抽出したブロック構造を持つXMLファイル(名大様提供) |
|  |  |  |

# はじめに

本書では、コード生成可能なSimulink モデルおよびMathWorks 社コード生成ツールによって出力されるrtw ファイルを読み、制御周期情報を名古屋大学様のモデルベース並列化フローのBLXMLに入力する機能について説明します。

# ソフトウェア構成

　ソフトウェア構成は以下の通りです

表 2‑1 ソフトウェア構成

|  |  |  |
| --- | --- | --- |
| 項番 | 名称 | 説明 |
| 1 | GetRate | RTWファイル中のブロックの周期(サンプル時間)を抽出したCSVファイルを出力する |
| 2 | SetRate | 周期を抽出したCSVファイルと入力となる処理対象のBLXMLファイルを入力し、ブロックの周期を更新したBLXMLファイルを出力する |
| 3 | RTWファイル | MathWorks社のコード生成ツールが出力する拡張子rtwが付いているファイル |
| 4 | 抽出周期CSVファイル | RTWファイルからブロックの周期を抽出、出力したCSVファイル |
| 5 | 入力BLXML | 処理対象のBLXMLファイル |
| 6 | 出力BLXML | RTWファイルから抽出したブロックの周期を反映したBLXMLファイル |

RTWファイル

GetRate

抽出周期

CSVファイル

SetRate

出力BLXML

入力BLXML

Cコード

図 2‑1 プログラムと入出力ファイル

## 実行環境

　本プログラムはJava SE(JDK8)が実行可能なプラットフォームで動作します。また、ビルドおよび実行については表 2‑2のツール・ライブラリが必要です。

表 2‑2 ツール・ライブラリ一覧

|  |  |  |  |
| --- | --- | --- | --- |
| No. | 名称 | 説明 | 入手先 |
| 1 | JDK8 | Java開発環境 | <http://www.oracle.com/technetwork/jp/java/javase/overview/index.html> |
| 2 | CUP  (0.11b) | Java用LALRパーサー | <http://www2.cs.tum.edu/projects/cup/>  (本ドキュメントでは0.11bを使用) |
| 3 | JFlex  (1.6.1) | Java用字句解析ルーチン生成 | <http://jflex.de/> |
| 4 | Apache Ant | ビルドツール | <http://ant.apache.org/> |

## ファイル一覧

　以下は本ドキュメント記載の機能に関わるディレクトリ構成とファイルの一覧です。

Simulink\_XSD/java

│ build.xml

│ SetRate.sh

│ GetRate.sh

│

├─src

│ └─jp

│ └─ertl

│ ├─simulink

│ │ ├─BLXML

│ │ │ SchemaDefine.java

│ │ │

│ │ └─rtw

│ │ RTWSampleTime.java

│ │ RTWParam.java

│ │ RTWSectionBase.java

│ │ RTWBlock.java

│ │ RTWSystem.java

│ │ RTWCompiledModel.java

│ │

│ └─utils

│ Getopts.java

│ SimpleCSV.java

│

├─grammer

│ RTWScanner.flex

│ RTWParser.cup

│

├─lib

│ readme-cup.txt

│

├─GetRate

│ GetRate.java

│

├─SetRate

│ SetRate.java

│

└─schema

readme-schema.txt

jaxb-prop.xjb

図 2‑2　ディレクトリ構成

* build.xmlはantでビルドするための設定ファイルです。ただし、本ドキュメント外の機能と共用しているため、本ドキュメント外の機能のビルド設定も含まれます
* 実際にビルドする際には、libにCUPのライブラリを配置する必要があります。lib\readme-cup.txtはその説明です
* 実際にビルドする際にはSimulinkModel.xsdをschemaに配置する必要があります。schema\readme-schema.txtはその説明です
* jaxb-prop.xjbはJAXBにより生成されるJavaコードのパッケージ名の指定と、JAXBで衝突が発生するXMLの属性の”value”をValueAttributeに変更するために用います。なおパッケージ名はjp.ertl.simulink.SimulinkModelです

# ビルド手順

## ビルド準備

### JDK8のインストール

最新のJDK8をダウンロードサイトからダウンロードし、インストールしてください。

<http://www.oracle.com/technetwork/java/javase/downloads/index.html>

* インストールしたJDKのパスをJAVA\_HOME環境変数に適切に設定してください
* 環境変数PATHにjavac,xjcが実行可能なように設定してください。具体的には $JAVA\_HOME/binをPATHに追加設定します

### Apache Antのインストール

最新のantをダウンロードし、インストールしてください。

<https://ant.apache.org/bindownload.cgi>

antが実行できるようantコマンドがあるディレクトリを環境変数にPATHに追加設定してください。

### CUPの配置

以下からCUPをダウンロードしてlibに展開します。

<http://www2.cs.tum.edu/projects/cup/>

展開されたjarファイルがlib直下にある必要があります。

### SimulinkModel.xsdの配置

最新の状態のSimulink\_XSD/SimulinkModel.xsdをschemaにコピーしてください

## ビルド

Simulink\_XSD/javaのディレクトリにて以下のようにantを実行してください。

$ ant

ただし、本ドキュメント外の機能のビルドも含まれるため、本ドキュメント外の機能のビルドも同時に行われます。

## javadocの生成(オプショナル)

ドキュメント(javadoc)を生成するには以下のようにantを実行してください。

$ ant javadoc

javadoc以下にHTMLファイルが生成されますので、内容については、任意のWeb/HTMLブラウザで参照してください。なお、本ドキュメント外の機能のビルドも含まれるため、本ドキュメント外の機能のビルドも同時に行われます。

# 実行

## GetRateコマンド仕様(javaコマンド使用時)

　javaコマンドにて実行する場合、以下のように指定します。

$ java –cp “*クラスパス*” GetRate [-v] [-o 出力CSVファイル] *RTWファイル*

* ***クラスパス***は次のものを指定する必要があります。
  + lib/java-cup-11b-runtime.jarの相対または絶対パス
  + GetRate.classがあるディレクトリ
* **-v**オプションは実行時に様々な情報を標準出力に表示します。このため、出力ファイルを指定しない(**-o**オプションを指定しない)場合には指定していても無視します
* **-o**オプションは抽出したブロックの周期を出力するCSVファイルを指定します。**-v**オプションを指定する場合は、**-o**オプションを指定してください
* ***RTWファイル***はMathWorksのコード生成ツールが出力した拡張子がrtwのファイルを指定します

## GetRate.shコマンド仕様(shellスクリプト使用時)

GetRate.shは4.1GetRateコマンド仕様(javaコマンド使用時)を簡易に実行するためのスクリプトなので、オプションは同じです。

$ GetRate.sh [-v] [-o 出力CSVファイル] *RTWファイル*

なお、本コマンドは、実行時にコマンドが配置されているディレクトリから相対的なファイル参照を行うため、次のようなディレクトリ構造である必要があります。

<GetRate.shのあるディレクトリ>

├ GetRate.sh

├ build

│ └ GetRate.class

└ lib

└ java-cup-11b-runtime.jar

## SetRateコマンド仕様(javaコマンド使用時)

　javaコマンドにて実行する場合、以下のように指定します。

$ java –cp “*クラスパス*” SetRate [-v] -I 入力BLXML [-o 出力BLXML] [*CSVファイル*]

* ***クラスパス***は次のものを指定する必要があります。
  + JAXBにより生成され、コンパイルされたSimulinkModelのclassがあるディレクトリ
  + SetRate.classがあるディレクトリ
* **-v**オプションは実行時に様々な情報を標準出力に表示します。このため、出力ファイルを指定しない(**-o**オプションを指定しない)場合には指定していても無視します
* **-i**オプションは処理対象の入力BLXMLファイルを指定します。**これは必須オプションです**
* **-o**オプションは周期を反映した出力するBLXMLファイルを指定します。**-v**オプションを指定する場合は、**-o**オプションの使用を推奨します
* ***CSVファイル***はGetRateにより、rtwファイルから抽出されたブロックの周期が出力されたCSVファイルを指定します。***CSVファイル***の指定が省略された場合、標準入力から読み込みを行います

## SetRate.shコマンド仕様(shellスクリプト使用時)

SetRate.shは4.3SetRateコマンド仕様(javaコマンド使用時)を簡易に実行するためのスクリプトなので、オプションは同じです。

$ SetRate.sh [-v] -I 入力BLXML [-o 出力BLXML] [*CSVファイル*]

なお、本コマンドは、実行時にコマンドが配置されているディレクトリから相対的なファイル参照を行うため、次のようなディレクトリ構造である必要があります。

<SetRate.shのあるディレクトリ>

├ SetRate.sh

└ build

├ GetRate.class

　 └ (SimulinkModelのclassファイル)

# モジュール仕様

## GetRateのクラス

GetRate.java内で定義されるクラスは表 5‑1の通りです。

表 5‑1 GetRateクラス

|  |  |  |
| --- | --- | --- |
| クラス名 | 内容 | |
| GetRate | RTWファイルからブロックの周期を抽出しCSVファイルへ出力する | |
| メンバ | 説明 |
| verbose | 実行時のverboseフラグ |
| メソッド | 説明 |
| dumpSampleTime() | 抽出したブロックの周期をCSVファイルへ出力する |
| usage() | コマンドの使用方法を出力しexit(1)で終了する |
| main() | GetRateのメインの処理 |

### GetRateクラス

RTWファイルからブロックの周期を抽出しCSVファイルへ出力する主処理のクラスです。

#### 静的メンバverbose

private static boolean verbose = false;

プログラム実行時のverboseメッセージの表示有無のフラグです。-vオプションを指定するとtrueになります。

#### メソッドdumpSampleTime()

private static void dumpSampleTime(PrintStream out, RTWBlock block)

ブロック***block***にサンプル時間が指定されていれば、***out***にCSV形式でブロック名と周期を出力します。

#### メソッドusage()

private static void usage()

コマンドの使用方法を表示し、終了ステータス1で終了します。

#### メソッドmain()

public static void main(String args[])

GetRateの主処理です。コマンドラインのオプション処理、RTWファイルの入力処理を行い、周期が指定されているブロックを抽出し、抽出したブロックの周期をCSVファイルへ出力します。

## SetRateのクラス

SetRate.java内で定義されるクラスは表 5‑2の通りです。また、クラス内クラスは、表 5‑3の通りです

表 5‑2 SetRateクラス

|  |  |  |
| --- | --- | --- |
| クラス名 | 内容 | |
| SetRate | CSVファイルのブロックの周期をBLXMLに反映する | |
| メンバ | 説明 |
| verbose | 実行時のverboseフラグ |
| do\_inherit | 周期の継承を解決するかどうかのフラグ |
| do\_setBaseRate | 周期が未決定の場合、ベースレートを設定するかどうかのフラグ |
| blockRateMap | ブロック名をキーとしたRateの連想配列 |
| blockInfoMap | ブロック名をキーとしたBlockInfoの仮想配列 |
| メソッド | 説明 |
| loadCSV() | ブロックの周期が出力されたCSVファイルを読み込み、blockRatemapを構築する |
| makeBlockInfo() | 入力したXMLからblockInfoMapを構築する |
| correctBlockInfo() | XMLのマージ情報を元にBlockInfoのマージ情報を設定する |
| process() | 読み込んだ全ての情報から、ブロックに周期を反映する |
| inheritSampleTime() | 周期の継承を処理する試験実装 |
| setBaseRate() | 周期が未決定の場合ベースレートを設定する |
| usage() | コマンドの使用方法を出力しexit(1)で終了する |
| main() | SetRateのメインの処理 |
| 内部クラス | 説明 |
| Rate | ブロックの周期を格納するクラス |
| BlockInfo | ブロックの情報を保持するクラス |

表 5‑3 setRateクラス内クラスの一覧

|  |  |  |
| --- | --- | --- |
| クラス名 | 内容 | |
| Rate | ブロックの周期情報を格納するクラス | |
| コンストラクタ | 説明 |
| Rate(String,String) | ブロック名と周期(文字列)を設定するコンストラクタ |
| 静的メンバ | 説明 |
| baseRate | ベースレート(文字列) |
| baseRateValue | ベースレート(値) |
| メンバ | 説明 |
| name | BLXMLでのブロック名 |
| rate | ブロックの周期(文字列) |
| rateValue | ブロックの周期(値) |
| intRateValue | ブロックの周期のベースレート比 |
| used | 使用したかどうかのフラグ |
| 静的公開メソッド | 説明 |
| setBaseRate() | ベースレートを設定する |
| getBaseRate() | ベースレート(文字列)を取得する |
| getBaseRateValue() | ベースレート(値)を取得する |
| 公開メンバ | 説明 |
| getName() | BLXMLのブロック名を取得する |
| getRate() | ブロックの周期(文字列)を取得する |
| getRateValue() | ブロックの周期(値)を取得する |
| getIntRateValue() | 周期のベースレート比を取得する |
| isUsed() | 使用されたか調べる |
| setUsed() | 使用されたことを設定する |
| equals(Rate) | 周期が一致するか調べる(Rate) |
| equals(doube) | 周期が一致するか調べる(double) |
| equals(String) | 周期が一致するか調べる(String) |

|  |  |  |
| --- | --- | --- |
| BlockInfo | ブロックの情報を保持するクラス | |
| コンストラクタ | 説明 |
| BlockInfo(BlockT,Blockinfo) | XMLのブロック(BlockT)と上位階層のBlockInfoを設定するコンストラクタ |
| メンバ | 説明 |
| block | XMLのブロック(BlockT) |
| upperInfo | 上位階層のブロック(BlockInfo) |
| mergeOwnerInfo | このブロックをマージしているブロック |
| mergeBlocks | マージしているブロックのリスト |
| rateFixed | このブロックの周期が固定されたかどうかを示す |
| rate | このブロックのRateインスタンス |
| 公開メソッド | 説明 |
| getBlock() | XMLのブロックを取得 |
| getUpperInfo() | 上位階層のブロックを取得 |
| setMergeOwnerInfo() | このブロックをマージしているブロックを設定する |
| getMergeOwnerInfo() | このブロックをマージしているブロックを取得する |
| addMergeBlock() | マージするブロックを設定する |
| getMergeBlocks() | マージしているブロックのリストを取得する |
| rateFixed() | このブロックの周期を決定(固定)する |
| isRateFixed() | このブロックの周期が固定されているか調べる |
| setRate() | Rateインスタンスを設定する |
| getRate() | Rateインスタンスを取得する |
| getRateInMergeBlocks() | マージしているブロックに周期が決定しているものがあればそのRateインスタンスを返す |
| correctRate() | 自分自身の周期が決定しているか、またはマージしているブロックに周期が決定しているブロックがあれば、その周期をマージしているブロックに設定する |

### SetRateクラス

CSVファイルのブロックの周期をBLXMLに反映するクラスです。

#### 静的メンバverbose

private static boolean verbose = false;

プログラム実行時のverboseメッセージの表示有無のフラグです。-vオプションを指定するとtrueになります。

#### 静的メンバdo\_inherit

private static boolean do\_inherit = false;

周期(サンプル時間)の継承を解決するかどうかのフラグです。-Iオプションを指定するとtrueになります。**なお、この機能は実験的機能です。**

#### 静的メンバdo\_setBaseRate

private static boolean do\_setBaseRate = false;

周期が未決定の場合にベースレートに設定するかどうかのフラグです。-Bオプションを指定するとtrueになります。

#### 静的メンバblockRateMap

private static HashMap<String, Rate>

blockRateMap = new HashMap<String, Rate>();

ブロック名をキーとしたRateクラスのインスタンスの連想配列です。

#### 静的メンバblockInfoMap

private static HashMap<String, BlockInfo>

blockInfoMap = new HashMap<String, BlockInfo>();

ブロック名をキーとしたBlockInfoクラスのインスタンスの連想配列です。

#### メソッドloadCSV ()

private static void loadCSV(String csvFile)

csvFileで指定したCSVファイル(またはcsvFileがnullの場合は標準入力)からRTWファイルから抽出したブロックの周期を読み込み、ベースレートの設定と、blockRateMapの構築を行います。

#### メソッドmakeBlockInfo ()

private static void makeBlockInfo(BlocksT blocks, BlockInfo upper)

読み込んだBLXMLのブロックの情報からblockInfoMapの構築を行います。なお、このブロックが連想配列blockRateMapのキーであれば、その値のRateインスタンスをBlockInfoの周期情報として設定し、Rateインスタンスに使用済みとします。

#### メソッドcorrectBlockInfo ()

private static void correctBlockInfo(BlocksT blocks)

BLXMLのブロックのマージ情報から、BlockInfo間のマージ情報を構築します。

#### メソッドprocess ()

private static void process(BlocksT blocks)

プログラムに読み込んだ全ての情報から、BLXMLのブロックに周期を反映します。

#### メソッドinheritSampleTime ()

private static void inheritSampleTime(BlocksT blocks, String rate)

ブロックの周期が”-1”で継承となっているものに対し、他のブロックとの関係から継承の解決を試みます。**なお、この機能は実験的機能です。**

#### メソッドsetBaseRate ()

private static void setBaseRate(BlocksT blocks)

ブロックの周期が”-1”で継承となっていて、継承が解決されていない場合に、ブロックの周期をベースレートにします。

#### メソッドusage()

private static void usage()

コマンドの使用方法を表示し、終了ステータス1で終了します。

#### メソッドmain()

public static void main(String args[])

SetRateのmain関数です。コマンドラインのオプション処理、入力のBLXMLとCSVファイルを読み込み、周期を反映したBLXMLを出力します。

### Rateクラス

RTWファイルから抽出され、CSVに出力されているブロックの周期を格納するクラスです。

#### Rateコンストラクタ

public Rate(String b, String r)

ブロック名***b***と文字列の周期***r***を指定してRateクラスのインスタンスを生成します。

#### 静的メンバbaseRate

private static String baseRate;

ベースレートの値(文字列)を保持します。

#### 静的メンバbaseRateValue

private static double baseRateValue;

ベースレートの値(浮動小数点)を保持します。

#### メンバname

private String name;

BLXML内のブロック名を保持します。

#### メンバrate

private String rate;

ブロックの周期を文字列で保持します。

#### メンバrateValue

private double rateValue;

ブロックの周期を浮動小数点で保持します。

#### メンバintRateValue

private int intRateValue;

ブロックの周期をベースレートとの整数比で保持します。

#### メンバused

private boolean used;

このブロックの周期を使用したかどうかのフラグです。

#### 静的メソッドsetBaseRate ()

public static void setBaseRate(String b)

***b***で指定した文字列のベースレートを設定します。

#### 静的メソッドgetBaseRate()

public static String getBaseRate()

文字列のベースレートを取得します。

#### 静的メソッドgetBaseRateValue ()

public static double getBaseRateValue()

浮動小数点のベースレートを取得します。

#### メソッドgetName ()

public String getName()

このRateインスタンスのBLXMLでのブロックの名前を取得します。

#### メソッドgetRate ()

public String getRate()

このRateインスタンスの文字列の周期を取得します

#### メソッドgetRateValue ()

public double getRateValue()

このRateインスタンスの浮動小数点の周期を取得します。

#### メソッドgetIntRateValue ()

public int getIntRateValue()

このRateインスタンスの周期の整数のベースレート比を取得します。

#### メソッドisUsed ()

public boolean isUsed()

このRateインスタンスが使用されたかどうかを真理値で返します。trueならば使用されあ事を示します。

#### メソッドsetUsed ()

public void setUsed ()

このRateインスタンスを使用した事を設定します。

#### メソッドequals ()

public boolean equals(Rate r)

public boolean equals(double d)

public boolean equals(String s)

このRateインスタンスと等しいかどうかを判定します。

### BlockInfoクラス

BLXMLのブロックの情報を内部処理用に保持するクラスです。

#### BlockInfoコンストラクタ

public BlockInfo(BlockT b, BlockInfo u)

BLXMLのブロック***b***と上位階層(SubSystem)の***u***を指定してBlockInfoクラスのインスタンスを生成します

#### メンバblock

private BlockT block;

BLXMLのブロックの<block>に対応するblock\_Tを保持します。

#### メンバupperInfo

private BlockInfo upperInfo = null;

このブロックの上位階層(SubSystem)のBlockInfoを保持します。

#### メンバmergeOwnerInfo

private BlockInfo mergeOwnerInfo = null;

このブロックをマージしているブロックのBlockInfoを保持します。

#### メンバmergeBlocks

private List<BlockInfo> mergeBlocks = null;

このブロックがマージしているブロックのBlockInfoのリストを保持します。

#### メンバrateFixed

private boolean rateFixed = false;

このブロックの周期が決定(固定)されているかどうかを示すフラグです。trueの場合は固定化され、継承に対して変化しません。

#### メンバrate

private Rate rate;

このブロックの周期を格納したRateクラスのインスタンスを保持します。

#### メソッドgetBlock ()

public BlockT getBlock()

BLXMLの<block>に対応するBlock\_Tを取得します。

#### メソッドgetUpperInfo ()

public BlockInfo getUpperInfo()

このブロックの上位階層(SubSystem)を取得します。

#### メソッドsetMergeOwnerInfo ()

public void setMergeOwnerInfo(BlockInfo o)

このブロックをマージしているブロックを設定します。

#### メソッドgetMergeOwnerInfo ()

public BlockInfo getMergeOwnerInfo()

このブロックをマージしているブロックを取得します。

#### メソッドaddMergeBlock ()

public void addMergeBlock(BlockInfo b)

このブロックがマージしているブロックをリストに追加します。

#### メソッドgetMergeBlocks ()

public List<BlockInfo> getMergeBlocks()

このブロックがマージしているブロックのリストを取得します。

#### メソッドrateFixed ()

public void rateFixed()

このブロックの周期を決定(固定)します。

#### メソッドisRateFixed ()

public boolean isRateFixed()

このブロックの周期が決定(固定)しているか調べます。trueならこのブロックの周期は決定(固定)しています。

#### メソッドsetRate ()

public void setRate(Rate r)

このブロックの周期(Rateインスタンス)を設定します。

#### メソッドgetRate ()

public Rate getRate()

このブロックの周期(Rateインスタンス)を取得します。

#### メソッドgetRateInMergeBlocks ()

public Rate getRateInMergeBlocks()

このブロックがマージしているブロックに周期が決定(固定)しているブロックがあれば、最初に見つかったブロックのRateインスタンスを取得します。

#### メソッドcorrectRate ()

public boolean correctRate()

このブロックまたはマージしているブロックに決定(固定)している周期があれば、マージしているブロック全体にその周期を設定します。