Embedded Technology 2008

トレースログ可視化ツール TraceLogVisualizer（TLV） のご紹介

名古屋大学大学院 情報科学研究科
情報システム学専攻 高田・冨山研究室
博士課程前期課程2年 後藤 隼弐

PROJECT OCEAN

発表内容

Embedded Technology 2008

- 背景
- TLVの紹介
- TLVデモンストレーション
- おわりに

背景

Embedded Technology 2008

マルチプロセッサの必要性

- ▹クロックアップによる性能向上は限界
  - ✖消費電力，発熱の急増

➔*処理の並列性を高めることで性能向上を目指す*

組込みシステムにおけるマルチプロセッサ利用

- ▹従来：コストが上がるためなるべく使いたくない

➔*マルチコアプロセッサの登場により低コストで利用可能に*

➔**組込み分野でのマルチコアプロセッサ利用の増加**

マルチコアプロセッサ環境での課題

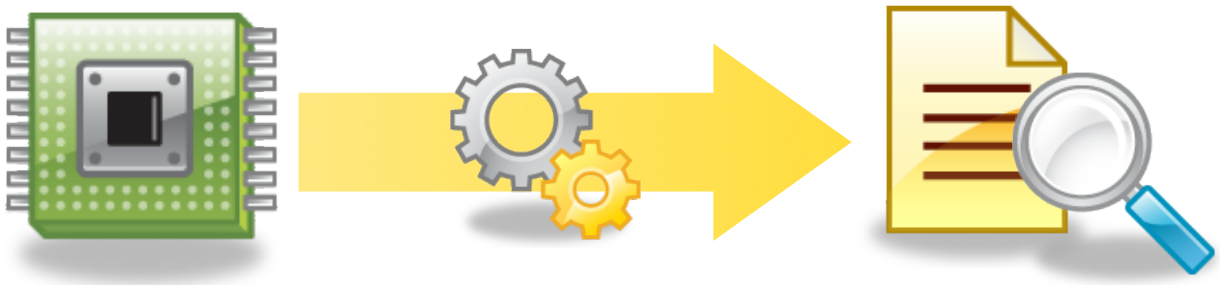
マルチコアプロセッサ環境でのデバッグ

- マルチコアプロセッサ環境では各コアが独立に並列動作
  - ×ブレークポイントやステップ実行を用いたデバッグが困難

➔*実行後のトレースログの解析によるデバッグが有効*

トレースログの解析によるデバッグ

- 実行後のトレースログを解析することによって動作を確認する
  - ×開発者がトレースログを直接扱うのは困難
  - ×トレースログのサイズや処理の複雑さによっては解析不可能



トレースログの例

RTOSのトレースログの例

- 4コア上でTOPPERS/FMPカーネルを実行したときのトレースログ

➔*右図*

- ×時系列に各コアの動作が分散していて把握するのが困難
- ×膨大な量なので所望の情報を探し出すのは至難の業

➔*トレースログの解析を支援するツールの要求*

```
.
.
[453665]:[1]: task 4 becomes RUNNABLE.
[453670]:[1]: dispatch to task5.
[454235]:[2]: leave from inh 1056.
[456997]:[3]: task 6 becomes RUNNABLE.
[457665]:[3]: dispatch to task6.
[457886]:[2]: task 4 becomes SUSPENDED.
[459345]:[4]: task 5 becomes RUNNABLE.
[459665]:[4]: dispatch to task5.
[460457]:[1]: task 5 becomes WAITTING.
[461007]:[1]: task 4 becomes RUNNABLE.
[463665]:[1]: dispatch to task4.
[469005]:[3]: task 6 becomes SUSPENDED.
.
.
.
```

4コア上で動くTOPPERS/FMPのトレースログの例

➔**トレースログを可視化表示するツールの開発へ**

TraceLogVisualizer（TLV）

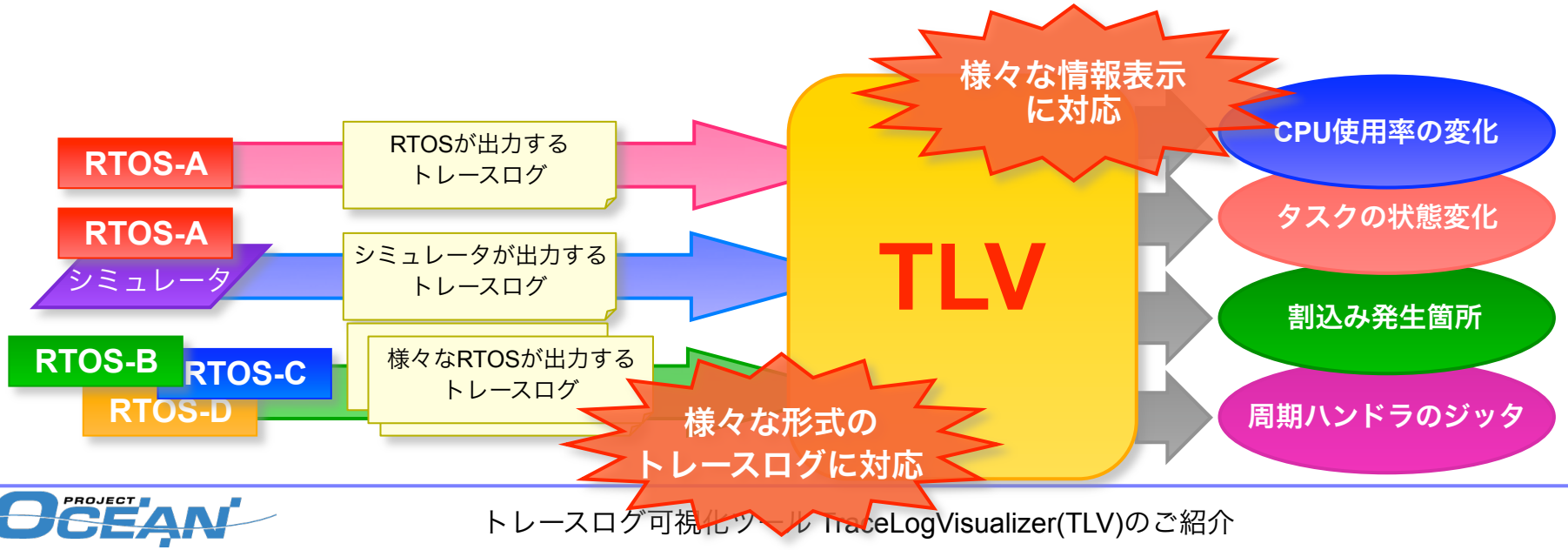
Embedded Technology 2008

TraceLogVisualizer（TLV）

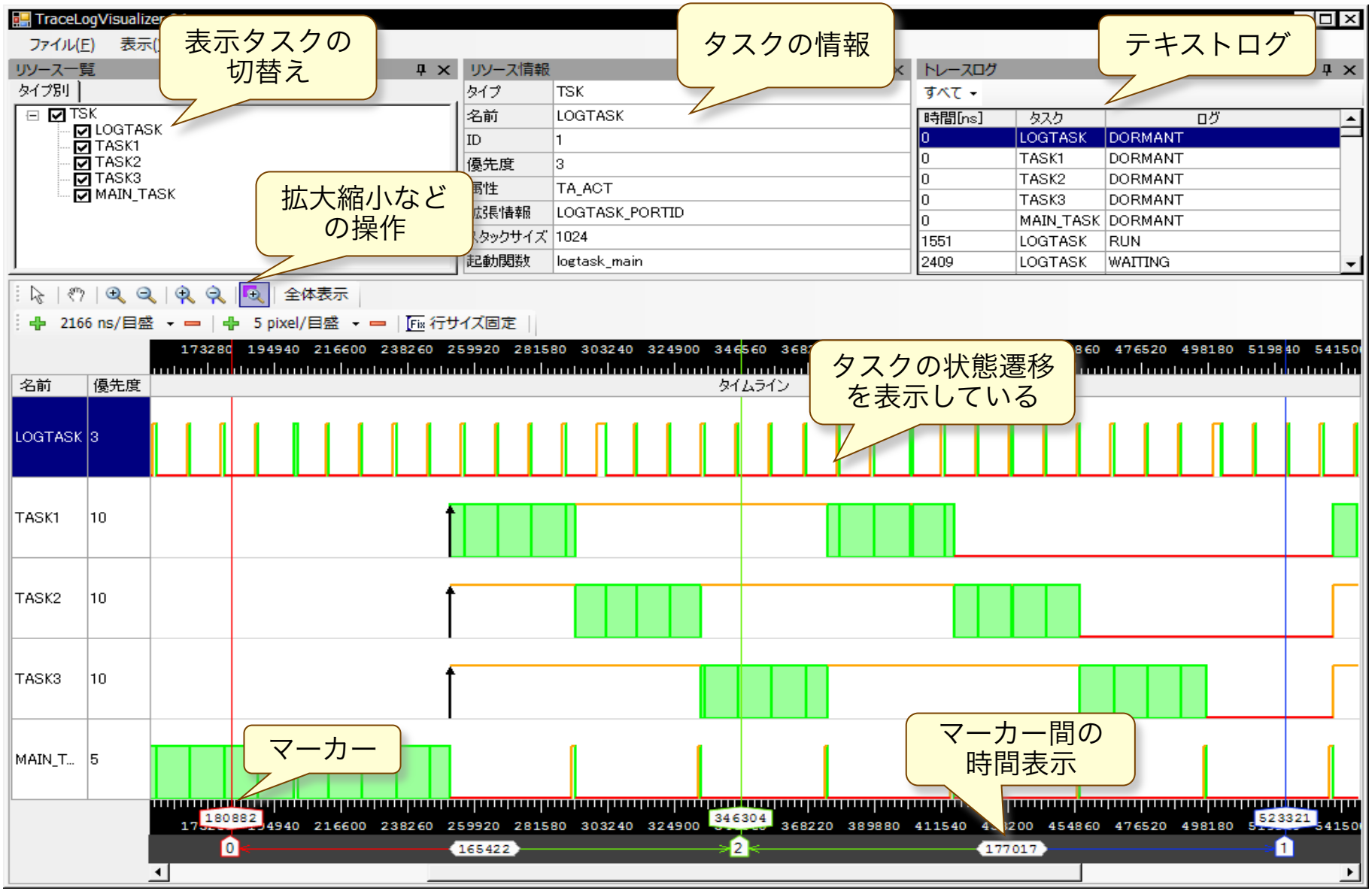
- トレースログを可視化表示するツール

特徴

- 様々な形式のトレースログに対応
- 様々な可視化表示に対応
- オープン化を予定
  - ➔ *将来的にTOPPERSプロジェクトからのリリースを検討*



TLVのスクリーンショット



TLVの使用例

TOPPERS/ASPカーネルを用いた例

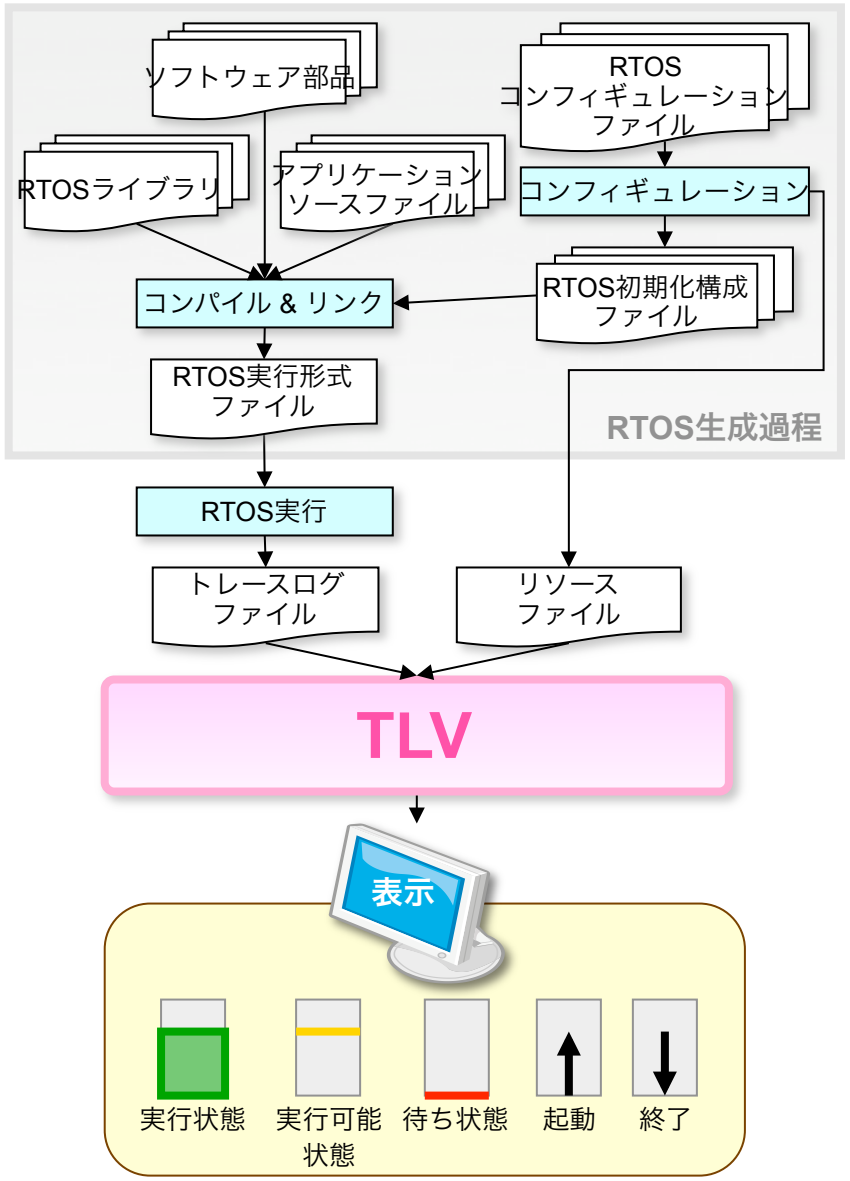
- TOPPERS/ASPカーネルは標準でトレースログを出力する機能を搭載している
  - タスクの状態遷移、システムコールの入口/出口、etc...

TLVへの入力ファイル

- トレースログファイル
  - 実行後に標準出力されるトレースログをファイル化したもの
- リソースファイル
  - 各タスクの情報を記述したファイル
  - カーネルのコンフィギュレーション段階で自動生成

可視化表示される情報

- タスクの状態遷移



TLVデモンストレーション

おわりに

Embedded Technology 2008

開発体制

- 文部科学省　先導的ITスペシャリスト育成推進プログラム

  プロジェクトOCEAN
  「OJLによる最先端技術適応能力をもつIT人材育成拠点の形成」

  において名古屋大学主導で開発

今後のロードマップ

- 2009年3月 ： TLV 1.0αのTOPPERS会員向けのリリース
  - 検索機能、統計情報出力機能の実装
- 2009年9月 ： TLV 1.0βのTOPPERS会員向けのリリース
  - テストケース入力によるデバッグ機能、ソースコードとの連携

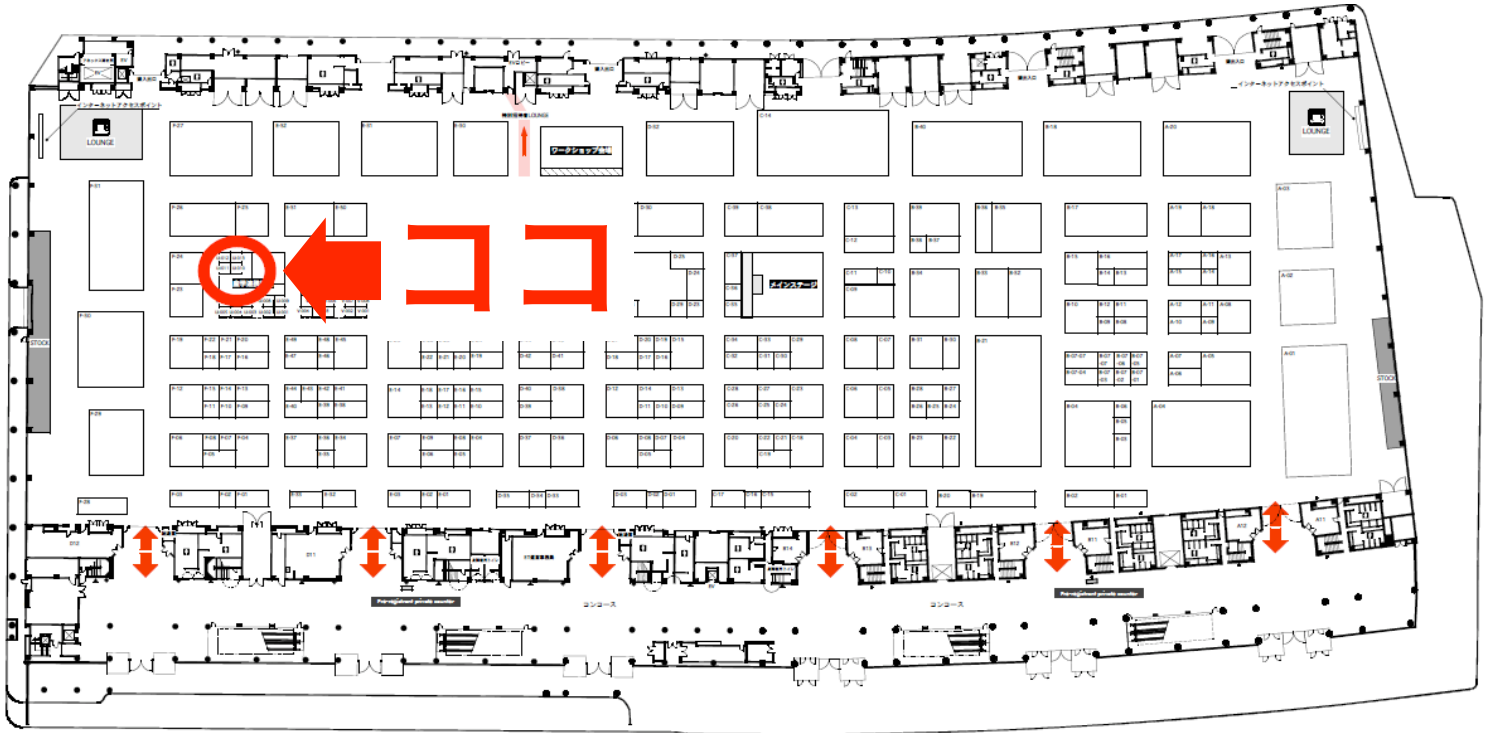
お問合せやご意見の送り先

- tlv@ertl.jp

ブース展示

ブース展示

- 産学連携推進パビリオンにて
  「名古屋大学 大学院情報科学研究科 高田・冨山研究室」
  として出展しています。
  - 展示ホール1F　ブース小間番号：UI-010



付録

付録内容

Embedded Technology 2008

- TLVが想定する可視化対象
- TLV全体像
- JSON形式
- 各ファイルの説明

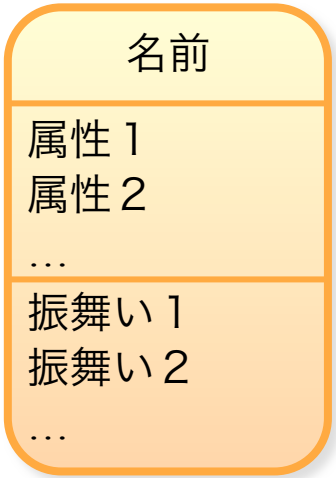
TLVが想定する可視化対象

Embedded Technology 2008

可視化情報を持つ単位

- ▹名前といくつかの属性/振舞いを持つもの

➔ ***リソース***と呼ぶ

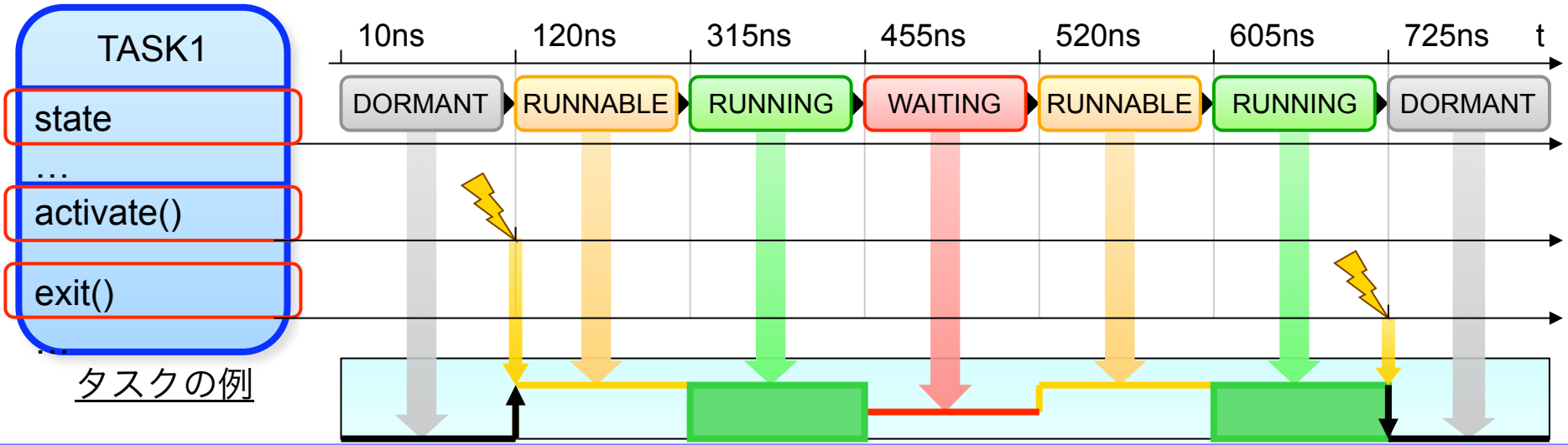


リソースの例

可視化対象

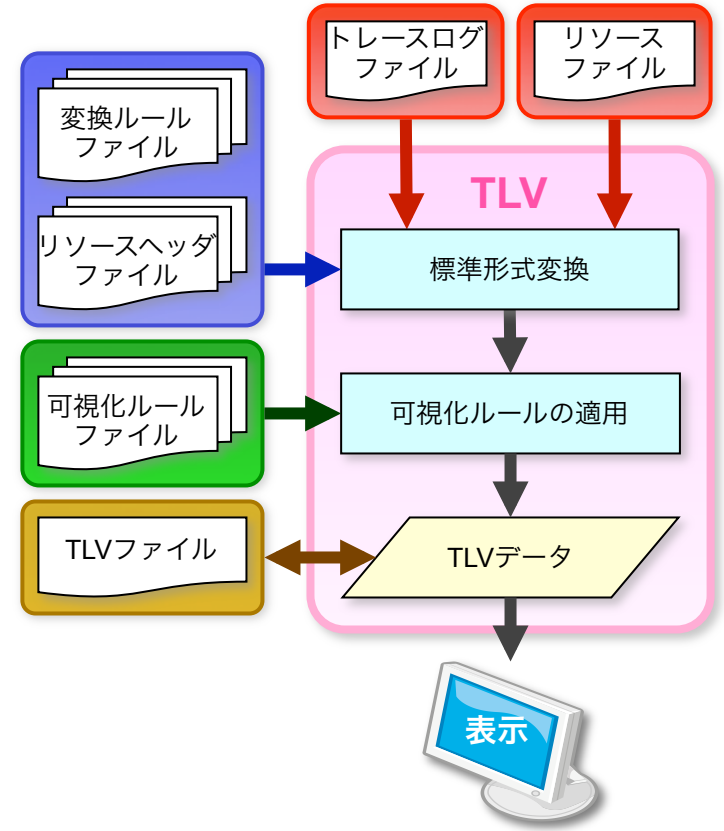
- ▹振舞いの発生
- ▹属性値の変化

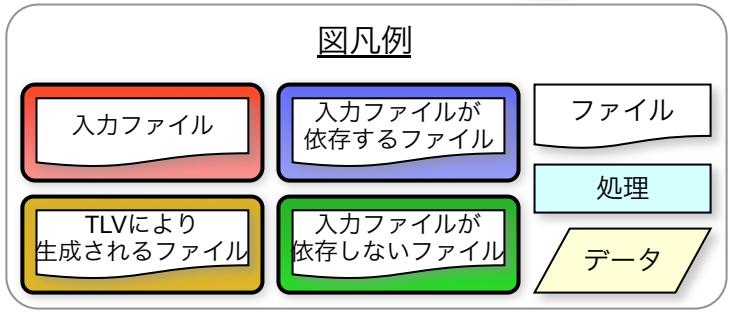
➔ ***イベント***と呼ぶ



タスクの例

TLV全体像





入力

- 可視化表示したいトレースログとリソースの情報

➔ *トレースログファイル*
➔ *リソースファイル*

標準形式への変換

- 様々な形式のトレースログに対応するため標準形式に変換する

➔ *リソースヘッダファイル*
➔ *変換ルールファイル*

可視化ルールの適用

- リソースの型に対応する可視化ルールを外部ファイルとして定義

➔ *可視化ルールファイル*

出力

- トレースログを可視化表示
- 可視化表示に必要なすべてのデータをまとめファイルとして出力

➔ *TLVファイル*

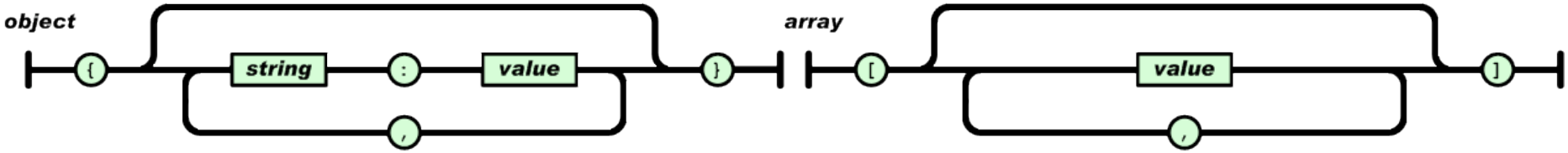
JSON形式

Embedded Technology 2008

JSON形式

- ユーザが記述する必要のあるすべてのファイルで**JSON形式**を採用
- RFC4627として仕様が規定されている
- 特徴 ： 単純。読み書きが容易。パースが簡単

➔*各ファイルの記述コスト、修得コストを低減*



value ::= string|number|bool|object|array|null

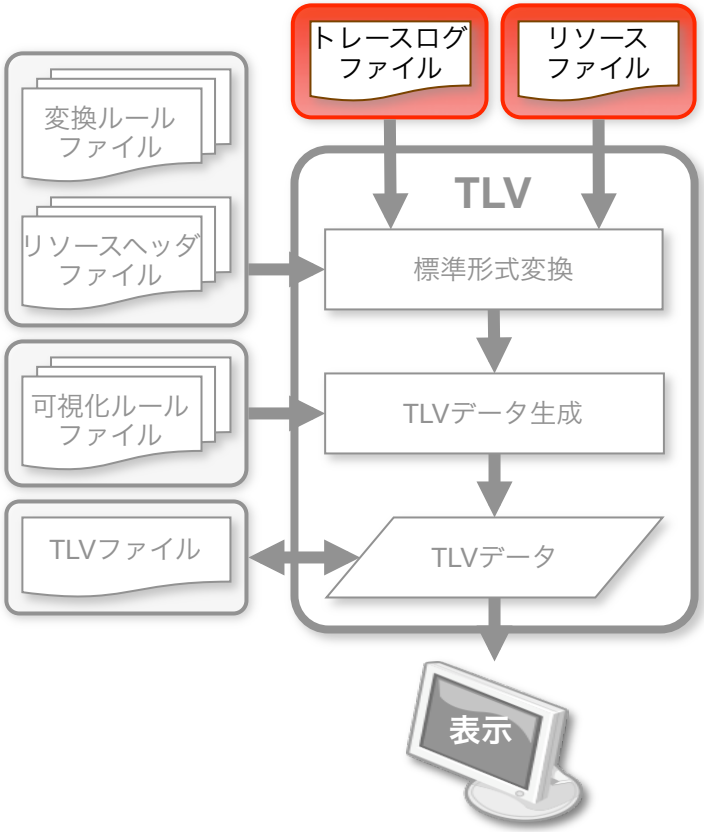
object例

“name”:{“attr1”:”val1”, “attr2”:1000, “attr3”:null, [1, 2, 3], {“key1”:”val1”,”key2”:”val2”}}

array例

“name”:[”val1”, true, 3.14, -10, “a\nb”, null, 23e-2, [1, 2, 3], {“key1”:”val1”, “key2”:”val2”}]

入力ファイル



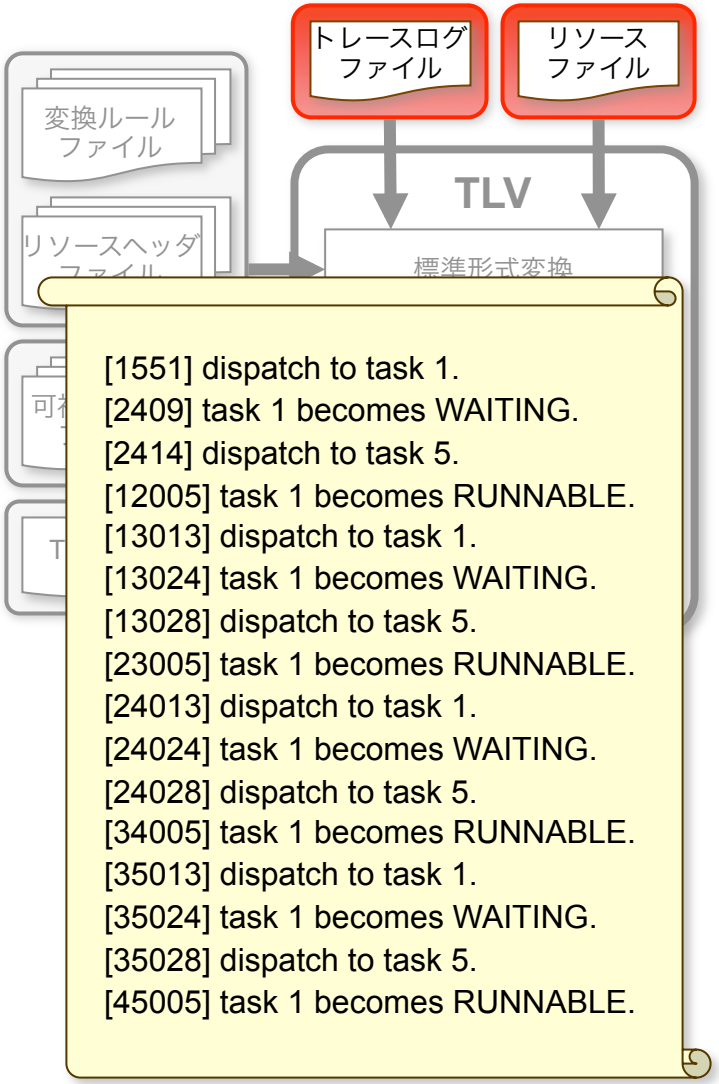
トレースログファイル

- OS、シミュレータ、エミュレータなどが出力するトレースログをファイル化したもの

リソースファイル

- リソースの情報、時間単位、変換ルール、リソースヘッダを記述
- ユーザが用意する
  - TOPPERSの場合
    - コンフィギュレータにより生成可能

入力ファイル



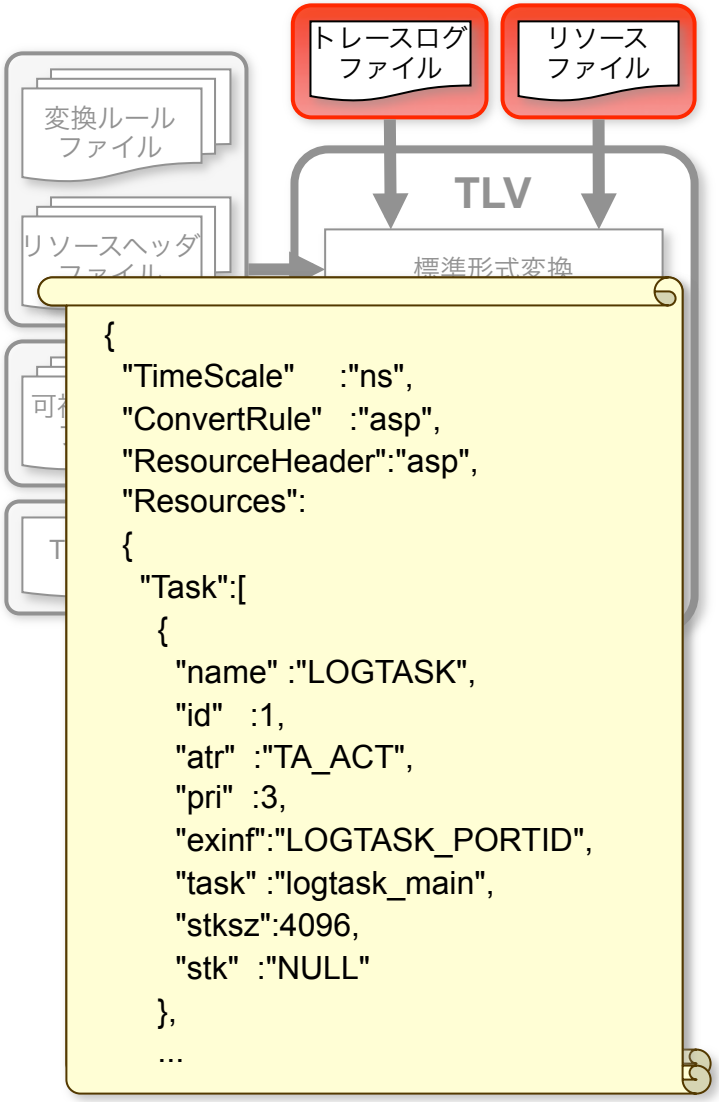
トレースログファイル

- OS、シミュレータ、エミュレータなどが出力するトレースログをファイル化したもの

リソースファイル

- リソースの情報、時間単位、変換ルール、リソースヘッダを記述
- ユーザが用意する
  - TOPPERSの場合
    - コンフィギュレータにより生成可能

入力ファイル



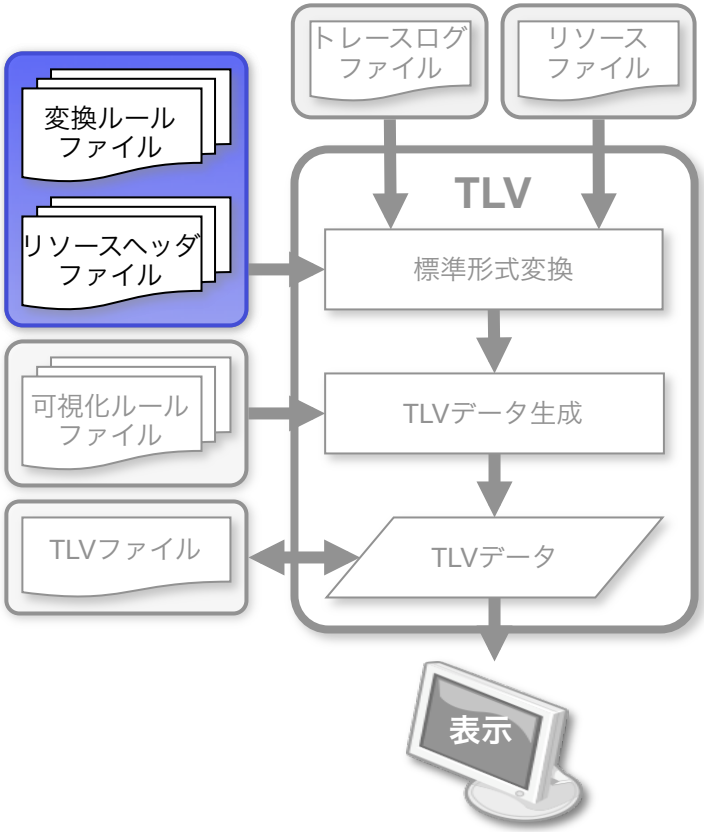
トレースログファイル

- OS、シミュレータ、エミュレータなどが出力するトレースログをファイル化したもの

リソースファイル

- リソースの情報、時間単位、変換ルール、リソースヘッダを記述
- ユーザが用意する
  - TOPPERSの場合
    - コンフィギュレータにより生成可能

入力ファイルが依存するファイル



- OSやシミュレータなどトレースログを出力するプラットフォームごとに用意する

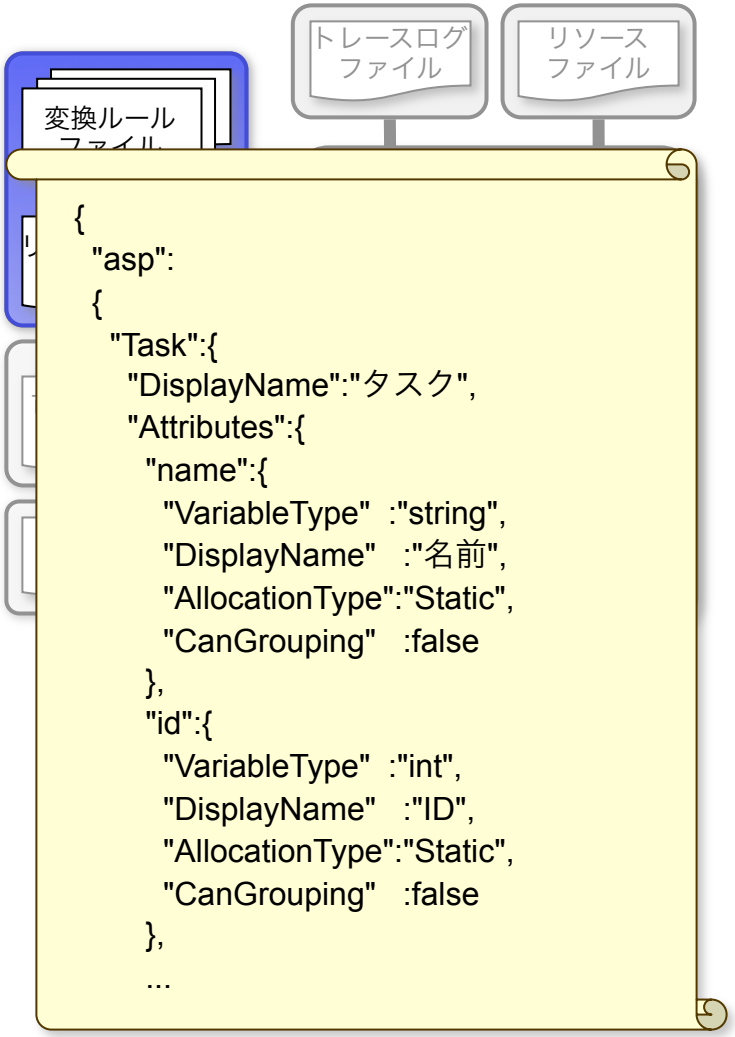
リソースヘッダファイル

- リソースの属性/振舞いを定義
- ユーザが記述
  - TOPPERS用は標準付属

変換ルールファイル

- 正規表現を用いた置換ルールを記述
- 置換を行う条件を記述出来る
  - あるリソースの属性値がXのとき
  - 属性値Xをもつリソースが存在するとき
  - 属性値Xをもつリソースが Y個のとき

入力ファイルが依存するファイル



- OSやシミュレータなどトレースログを出力するプラットフォームごとに用意する

リソースヘッダファイル

- リソースの属性/振舞いを定義
- ユーザが記述
  - TOPPERS用は標準付属

変換ルールファイル

- 正規表現を用いた置換ルールを記述
- 置換を行う条件を記述出来る
  - あるリソースの属性値がXのとき
  - 属性値Xをもつリソースが存在するとき
  - 属性値Xをもつリソースが Y個のとき

入力ファイルが依存するファイル

Embedded Technology 2008

検索する正規表現

文字列を抽出して利用できる

置換条件

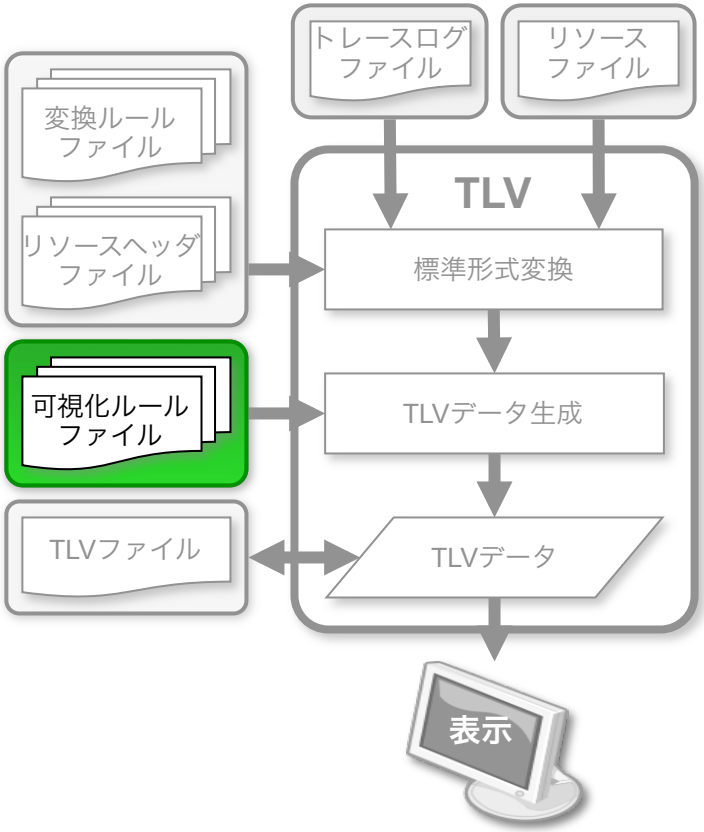
置換する文字列

```
{
  "asp":{
    "\[(?<time>\d+)\] dispatch to task (?<id>\d+)\.":[
      {"EXIST{Task(state==RUNNING)}":[
          "[${time}]Task(id==ATTR{Task(state==RUNNING).id}).preempt()",
          "[${time}]Task(id==ATTR{Task(state==RUNNING).id}).state=RUNNABLE"
        ]},
      "[${time}]Task(id==${id}).dispatch()",
      "[${time}]Task(id==${id}).state=RUNNING"
    ],
    "\[(?<time>\d+)\] task (?<id>\d+) becomes (?<state>[^\.]+)\.":[
      { "Task(id==${id}).state==DORMANT && ${state}==RUNNABLE" : "[${time}]Task(id==${id}).activate()",
        ...
      },
      "id":{
        "VariableType"  :"int",
        "DisplayName"   :"ID",
        "AllocationType":"Static",
        "CanGrouping"   :false
      },
      ...
```

変換ルールファイル

- 正規表現を用いた置換ルールを記述
- 置換を行う条件を記述出来る
  - あるリソースの属性値がXのとき
  - 属性値Xをもつリソースが存在するとき
  - 属性値Xをもつリソースが Y個のとき

可視化ルールファイル



可視化ルールファイル

- リソースの属性/振舞いと可視化表現を対応付けしたファイル
- 属性の型と対応付けすることによりリソースに依存しない汎用的な可視化ルールを作成可能

可視化ルールファイル


```
{ "VisualizeRules":{
    "Task.state":{
      "RUNNING" :"runningShapes",
      "RUNNABLE":"runnableShapes",
      "WAITING" :"waitingShapes",
      "DORMANT" :"dormantShapes",
      ...
    },
    "Task.activate()":"activateShapes",
    "Task.exit()"    :"exitShapes"
  },
  "Shapes":{
    "runningShapes":[{
       "Type":"Rectangle",
       "Area":["0,0","100%,80%"],
       "Pen" :{"Color":"0000ff00","Width":3.0},
       "Fill":"6600ff00"
     }],
    "runnableShapes":[{
       "Type":"Line",
       "Coordinates":["f(0),40%","t(0),40%"],
       "Pen":{"Color":"00ffff00","Width":3}
     }],
      ...
```

属性に対して指定

振舞いに対して指定

表示する図形の記述

可視化ルールファイル

- リソースの属性/振舞いと可視化表現を対応付けしたファイル
- 属性の型と対応付けすることによりリソースに依存しない汎用的な可視化ルールを作成可能