
TOPPERSプロジェクト プレス発表
2009年11月18日

トレースログ可視化ツールTraceLogVisualizer(TLV) の一般公開について

名古屋大学 大学院情報科学研究科
附属組込みシステム研究センター(NCES)

開発の背景

マルチコア環境でのデバッグ

- マルチコア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

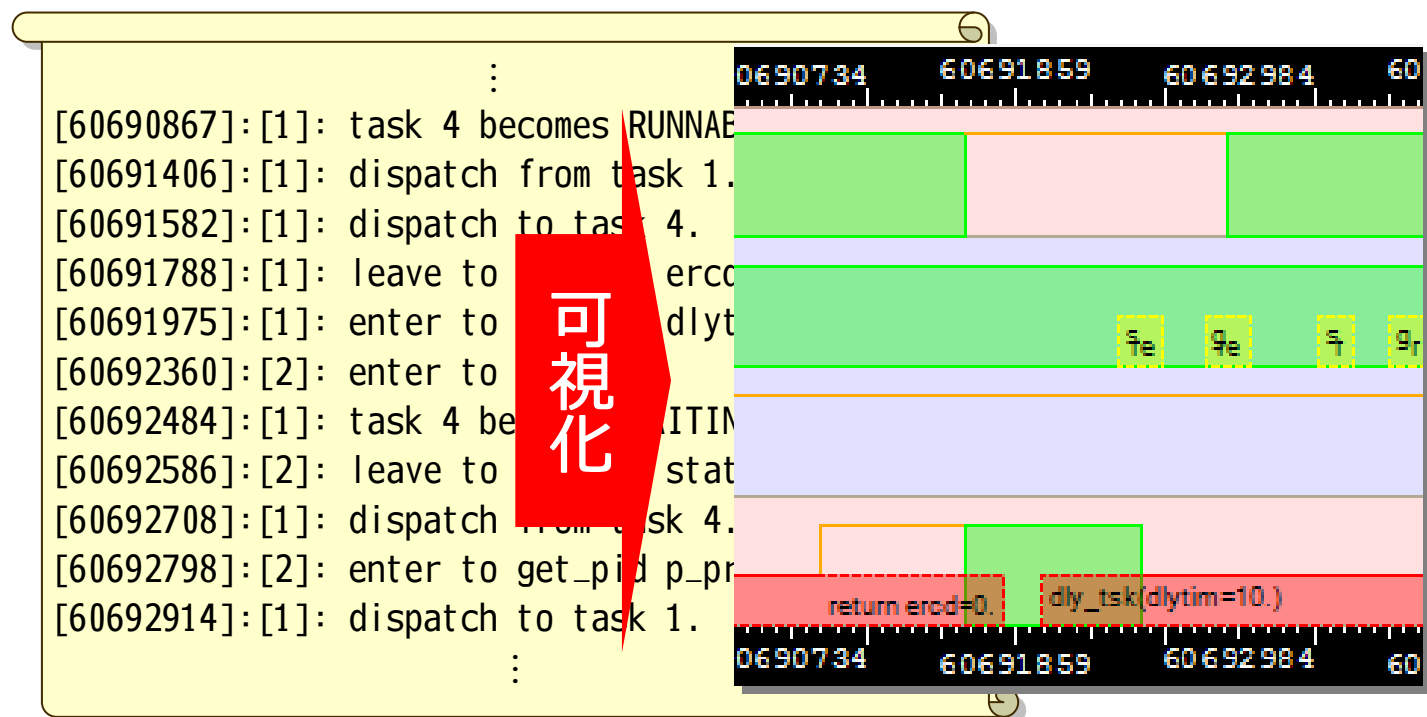
トレースログの解析によるデバッグ

- RTOSやシミュレータ、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には限界がある
 - × 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
 - トレースログの解析を支援するツールの要求

→ トレースログを可視化表示するツールの開発へ

2コア上で動くRTOSのトレースログの例

- × 時系列に各コアの動作が分散
- × 膨大な量
- ➡ 約1msの間に11個のログ



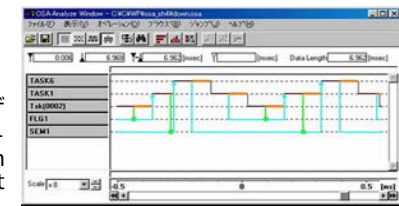
➡ トレースログを可視化表示し解析を支援

既存の可視化ツール

組込みシステム向けデバッグソフトウェア

- PARTNER-Jet イベントトラッカー
- WatchPoint OSアナライザ

WatchPoint OSアナライザ
<https://www.sophia-systems.co.jp/ice/products/watchpoint>



組込みシステム向け統合開発環境

- EvenTrek
- QNX System Profiler



EvenTrek
http://www.esol.co.jp/embedded/eb_multicore2.html

Unix系OSのトレースログプロファイラ

- Dtrace-Chime (Solaris)
- LTTV(Linux)

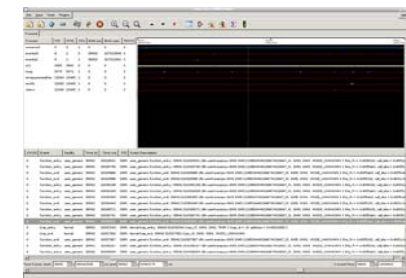
✗ ログの形式が標準化されていない

➡ 汎用性に乏しい

✗ 可視化表示項目が提供されているものに限られる

➡ 拡張性に乏しい

LTTV
<http://ltt.polymtl.ca/screenshots/>



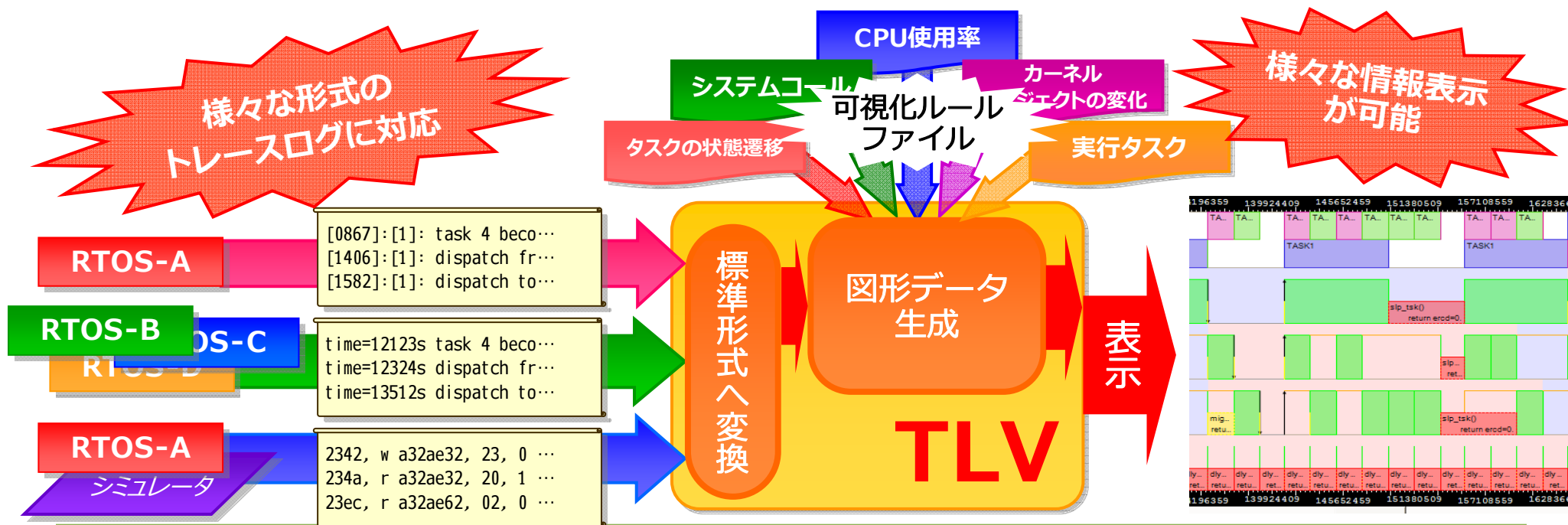
TraceLogVisualizer (TLV)

トレースログを可視化表示するツール

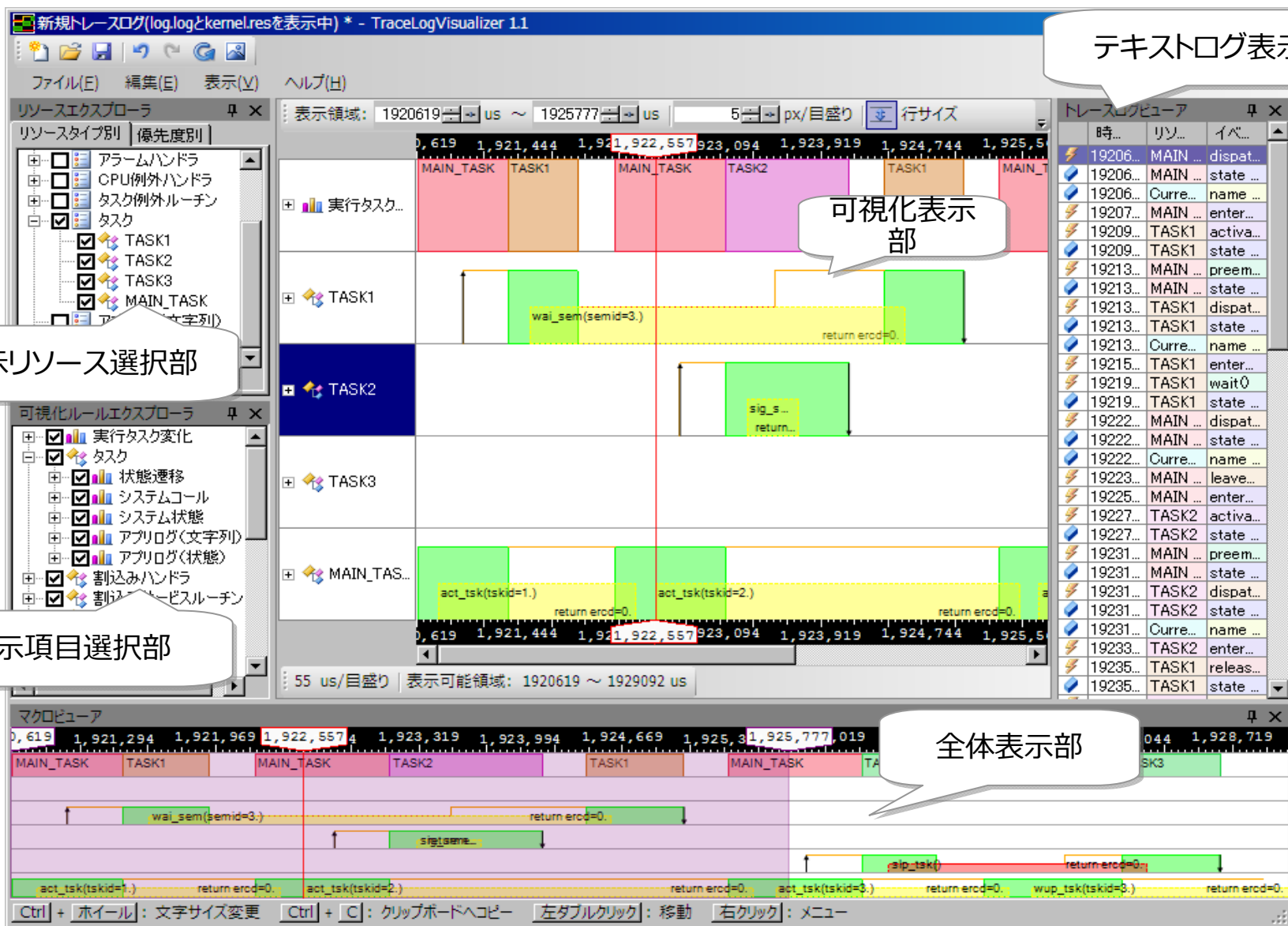
開発目標: 汎用性と拡張性

- ✓ 2009年5月 1.0rc リリース (TOPPERSメンバ向け)
- ✓ 2009年11月 1.1 リリース (一般公開)
- ✓ NCESなどで利用

Windows環境で動作



TLVのスクリーンショット



新規トレースログ(log.logとkernel.resを表示中) * - TraceLogVisualizer 1.1

ファイル(F) 編集(E) 表示(V) ヘルプ(H)

リソースエクスプローラ

リソースタイプ別 優先度別

- アラムハンドラ
- CPU例外ハンドラ
- タスク例外ルーチン
- タスク
 - TASK1
 - TASK2
 - TASK3
 - MAIN_TASK

表示リソース選択部

可視化ルールエクスプローラ

- 実行タスク変化
- タスク
 - 状態遷移
 - システムコール
 - システム状態
 - アプリログ(文字列)
 - アプリログ(状態)
 - 割り込みハンドラ
 - 割り込みサービスルーチン

表示項目選択部

表示領域: 1920619 ~ 1925777 us 5 px/目盛り 行サイズ

実行タスク...

TASK1

TASK2

TASK3

MAIN_TAS...

可視化表示部

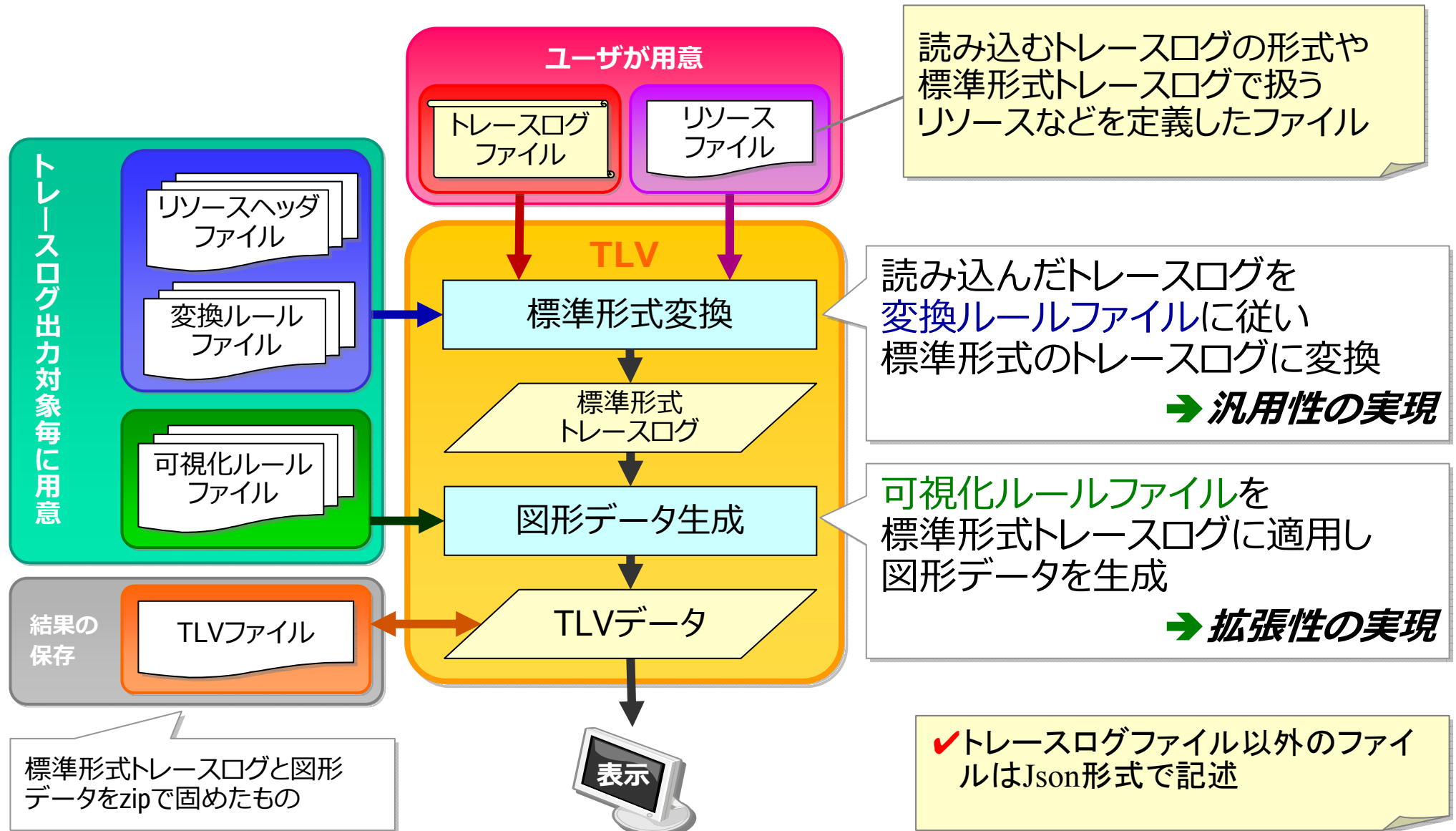
テキストログ表示部

マクロビューア

全体表示部

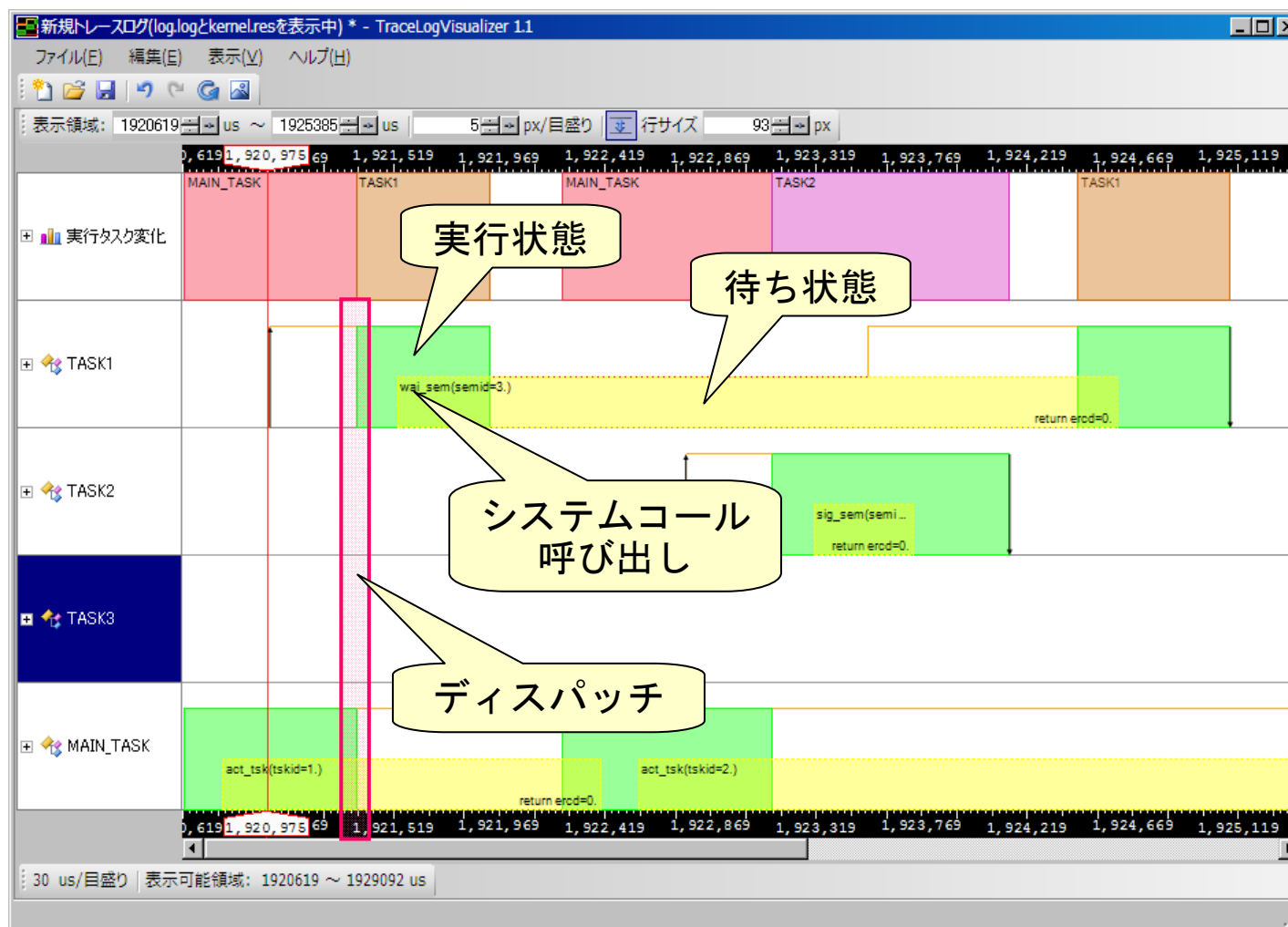
Ctrl + ホイール: 文字サイズ変更 Ctrl + C: クリップボードへコピー 左ダブルクリック: 移動 右クリック: メニュー

TLV全体像



TOPPERS/ASPカーネルのカーネルログ表示

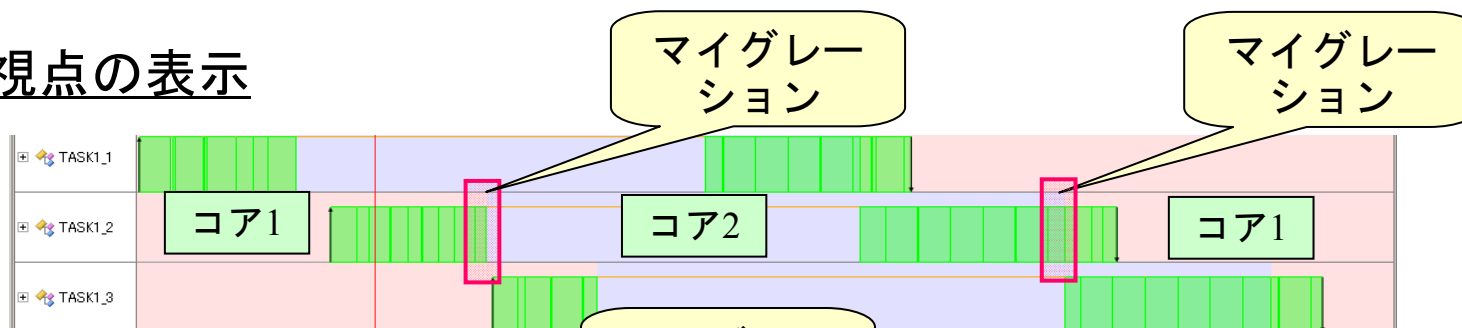
- タスクの状態変化, システムコール呼び出し, ディスパッチ等を可視化



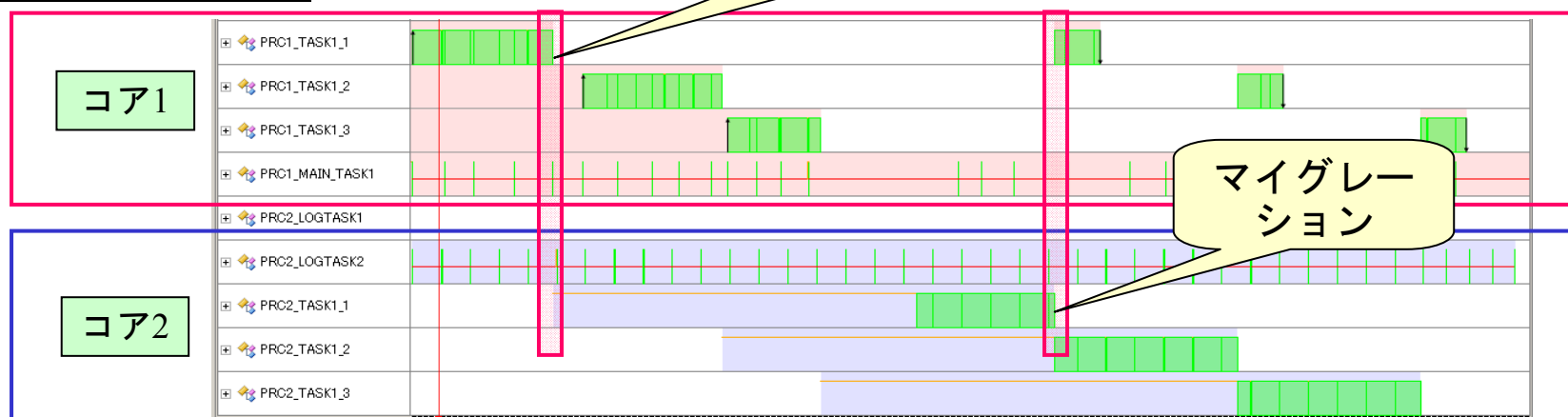
TOPPERS/FMPカーネルのカーネルログ表示

- TOPPERS/FMPカーネル
 - マルチプロセッサ対応のRTOS, マイグレーションをサポート
- マイグレーションの表現方法で2種類用意

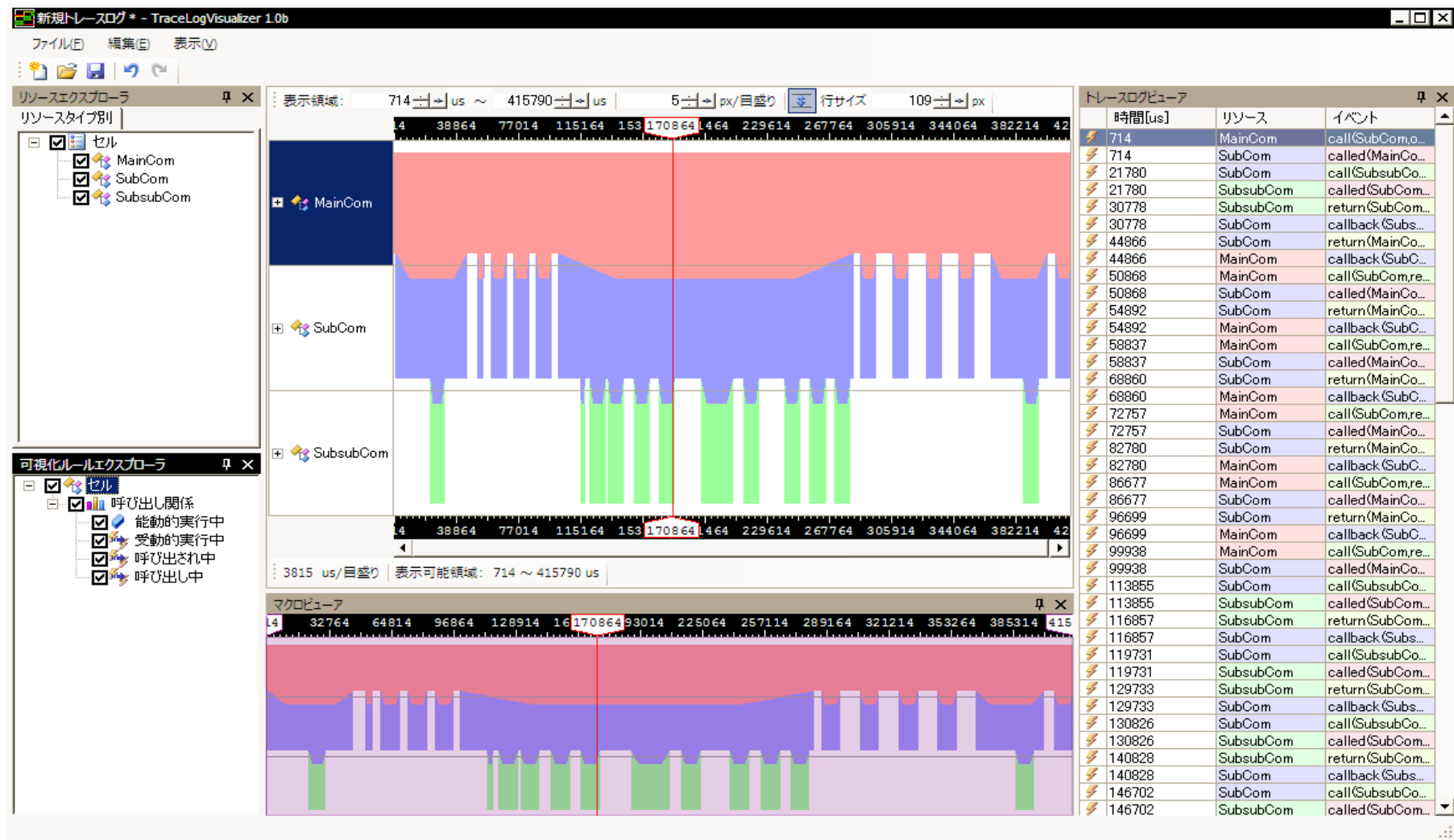
タスク視点の表示



コア視点の表示



TOPPERS組込みコンポーネントシステムの可視化

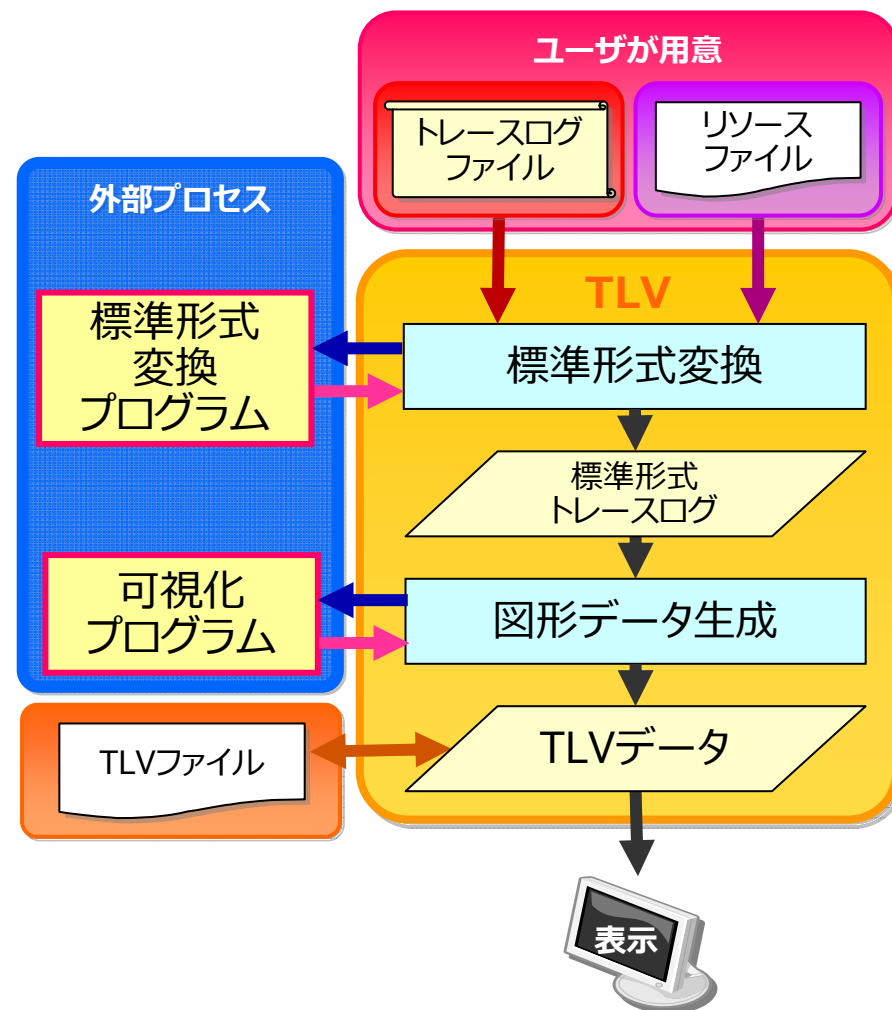


TLV 1.1 における改良点：スクリプト拡張

- 各変換に任意の言語を使用可能に
- 変換を外部プロセスで実行する
 - パイプを通じて通信する
- 既存のルールと併用可能

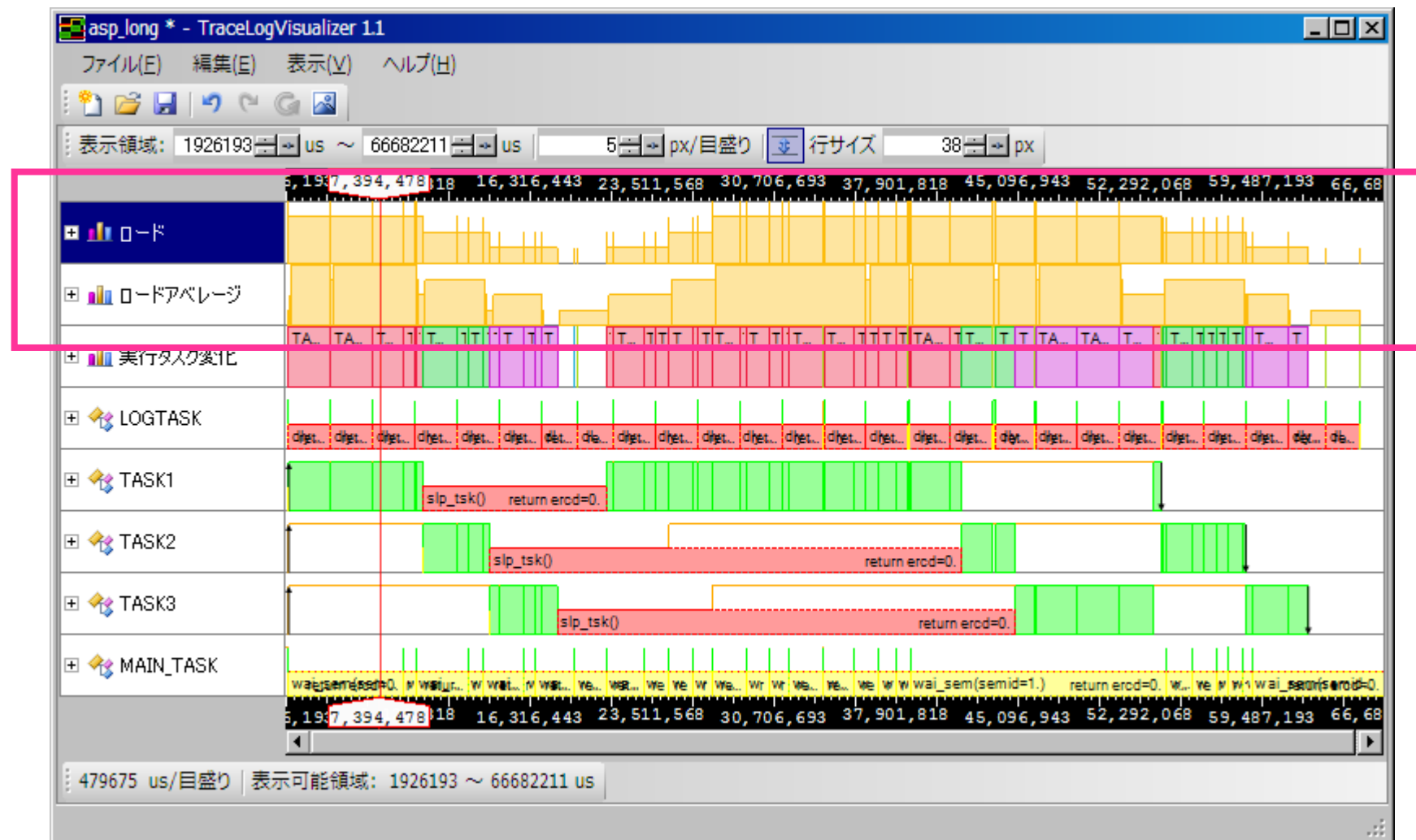
メリット

- ログから計算した結果を可視化可能
- 使い慣れた言語が利用できる
- 言語によっては高速化が期待できる
- 外部ツールとの連携が可能
 - 例: モデル検査機と連携させ、エラーを自動チェックする



スクリプト拡張：ロードとロードアベレージの可視化

- ・スクリプト拡張により可視化(Rubyで269行)



TLV 1.1 における改良点 : アプリログ

アプリ(ユーザー)レベルで使用頻度が
高い可視化項目を整備する

変換項目例

- 変数の値の変化の可視化
- 文字列出力の内容の可視化
- ユーザー定義状態の変化の可視化
- グラフ表示(要スクリプト拡張)

変換ルールの記述なしに可視化が可能

- リソースファイルは一部ユーザーが記述する必要がある
- syslog(sprintf)の出力フォーマットを定める

既存の可視化と併用可能

- ASP, FMPの可視化と併用可能

アプリログ：変数の値の変化と文字列出力の可視化

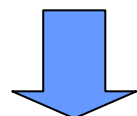
アプリ記述

文字列出力の可視化

```
syslog("applog strtask : TASK %d : %s.", 3 "pro..")
```

変数の値の変化の可視化

```
syslog("applog str : ID %d : %d.", 1, a)
```



実行して取得

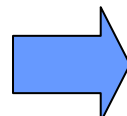
出力ログ

文字列出力の可視化

```
[time] applog strtask : TASK 3: pro..
```

変数の値の変化の可視化

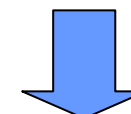
```
[time] applog str : ID 1 : 1
```



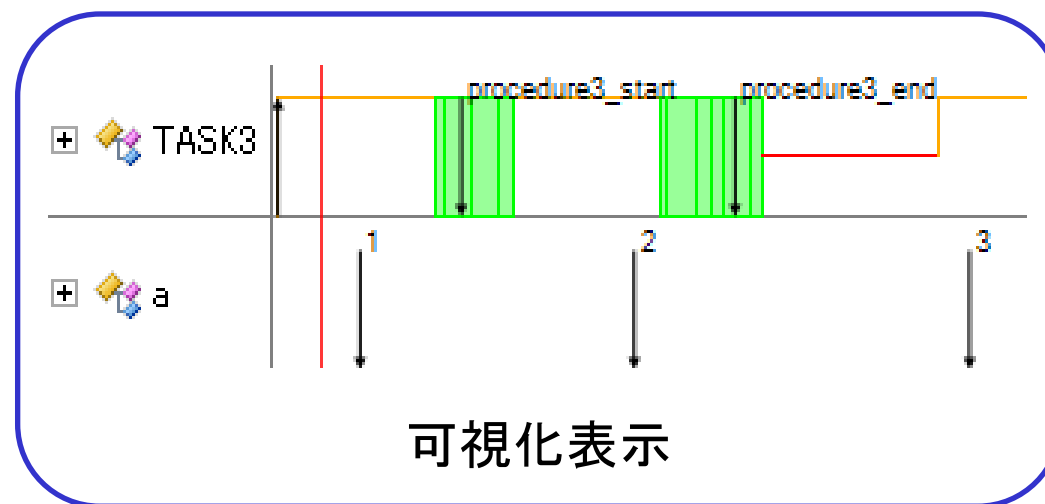
入力

リソースファイル

```
"a":{  
  "Type":"ApplogStr",  
  "Attributes":{  
    "id":"1"  
    "str":""  
  }  
}
```



入力



TLVの開発体制



開発体制

- 名古屋大学 大学院情報科学研究科 ITスペシャリストコースにおけるOJL (On the Job Learning) の開発テーマとして, 配属された大学院生らによって開発
- 2008年度: 修士2年生 1名が配属
 - 情報処理学会 山下記念研究賞/学生奨励賞 を受賞
- 2009年度: 修士2年生 2名が配属
- 2010年度: 修士2年生 2名が配属予定

ITスペシャリストコース

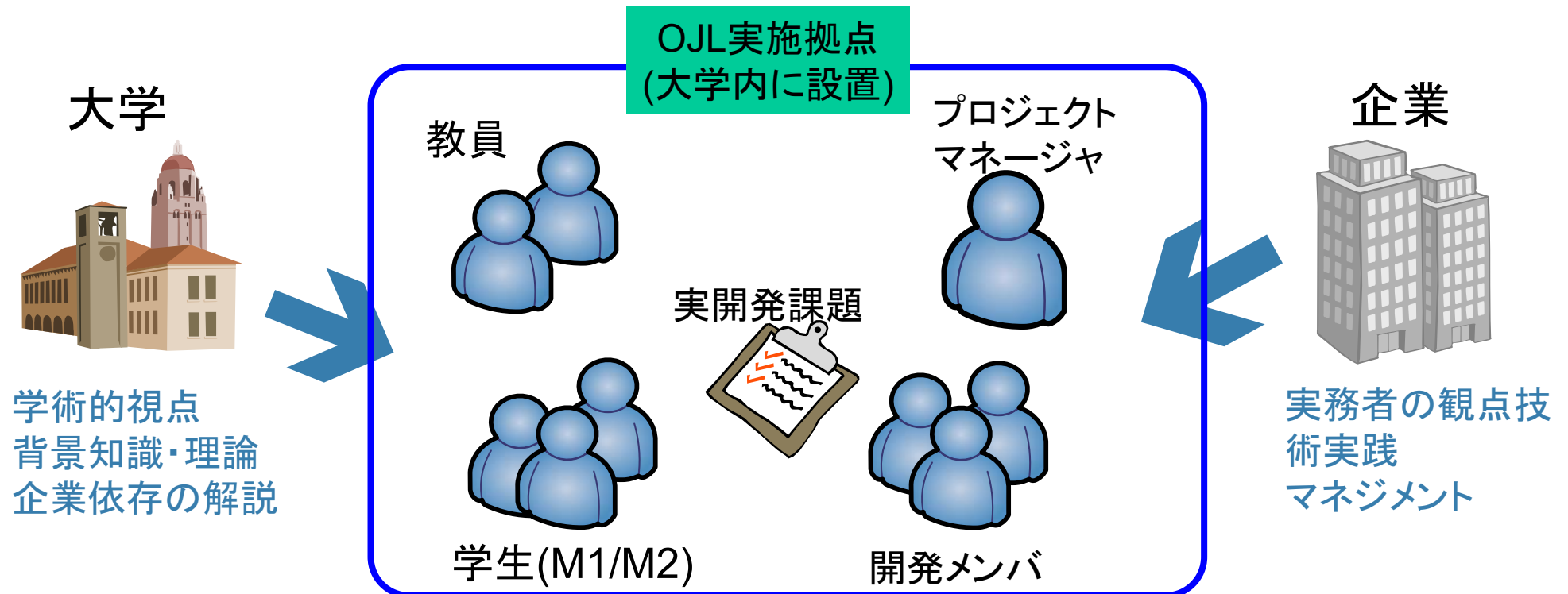
- 文部科学省の「先導的ITスペシャリスト育成推進プログラム」に対して, 名古屋大学・南山大学・愛知県立大学・静岡大学・6社の企業の共同で提案し採択された「OJLによる最先端技術適応能力をもつ人材育成拠点形成」(OCEAN プロジェクト)により, 名古屋大学 大学院情報科学研究科に設置された履修コース

OJL: On the Job Learning

OJL (On the Job Learning)



- 企業で行われているソフトウェア開発プロジェクトを教材とする実践教育
- 教育担当，業務担当の双方が指導
- 実開発課題を大学で長期間実施
 - 学生の教育だけでなく，社会人教育も



OJLテーマ一覧

テーマ名	連携企業	実施大学
システム構成管理技術の研究	トヨタ自動車株式会社	名古屋大学 愛知県立大学
ラベルライターの外国語向けソフトウェア開発	ブラザー工業株式会社	名古屋大学 愛知県立大学
次世代運転支援システムの情報表示系機能を検証するツールの開発	株式会社デンソー	名古屋大学
次世代LANプログラム開発	株式会社オートネットワーク技術研究所	名古屋大学
半導体製造装置のFA通信ソフトウェア開発	東京エレクトロンソフトウェア・テクノロジーズ株式会社	南山大学
自動販売機制御ソフトウェアの再開発	富士電機リテイルシステムズ株式会社	南山大学
組み込みソフトウェアに向くコンポーネント仕様の研究	アイシン精機株式会社※	名古屋大学
JavaソースコードのCDI(Code Inspection)の開発	(株)キャナリーリサーチ※	南山大学
大学事務システムの開発	大学主導	南山大学
Multicore対応リアルタイムOSの開発	大学主導	名古屋大学

※印は 連携6社以外からのテーマ

OJL実施の特徴 (修論・共同研究との違い)

- 企業PMのプロジェクト管理
- 専用の作業室 (機密管理のため)
- 週例ミーティングで進捗管理
- ドキュメント作成(週報, 各種開発成果物等)
- 品質管理活動(テスト/レビュー)
- 半年ごとに全OJLテーマの発表会

