組込み RTOS 向けアプリケーション開発支援ツール TLV（トレース ログ ヴィジュアライザー）フェーズ 4 リファクタリング仕様書

2009 年 5 月 28 日

改訂履歴

版番	日付	更新内容	更新者
1.0	09/5/28	新規作成	水野洋樹

目次

1 **はじめに** 3

1.1 本書の目的 3

1.2 本書の適用範囲 3

1.3 用語の定義/略語の説明 3

1.4 概要 3

2 **概要説明** 4

2.1 リファクタリングを実施する理由 4

2.2 リファクタリングを実施する対象 4

3 **作業内容** 4

3.1 リファクタリング前と後の処理の流れ 4

3.2 実施内容 9

1 はじめに

1.1 本書の目的

本書の目的は、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「OJL による最先端技術適応能力を持つ IT 人材育成拠点の形成」プロジェクトにおける、OJL 科目ソフトウェア工学実践研究の研究テーマである「組込み RTOS 向けアプリケーション開発支援ツールの開発」に対して、その開発するソフトウェアに対する要求を記述することである。

本書は特に、フェーズ 4 におけるリファクタリング作業に関する記述を行う。

1.2 本書の適用範囲

本書は、組込み MPRTOS 向けアプリケーション開発支援ツールの開発プロジェクト（以下本プロジェクト）のフェーズ 4 におけるリファクタリング作業に関する記述を行う。

1.3 用語の定義/略語の説明

表 1　用語定義

用語・略語	定義・説明
TLV	Trace Log Visualizer
MPRTOS	マルチプロセッサ対応リアルタイムオペレーティングシステム
トレースログファイル	RTOS のトレースログ機能を用いて出力したトレースログや、シミュレータなどが出力するトレースログをファイルにしたもの
標準形式トレースログファイル	本ソフトウェアが扱うことの出来る形式をもつトレースログファイル。各種トレースログファイルは、この共通形式トレースログファイルに変換することにより本ソフトウェアで扱うことが出来るようになる。
変換ルール	トレースログファイルを標準形式トレースログファイルに変換する際に用いられるルール。
可視化ルール	標準形式トレースログファイルを可視化する際に用いられるルール。
TLV ファイル	本ソフトウェアが中間形式として用いるファイル。前述の標準形式トレースログファイルは、この TLV ファイルの一部である。

1.4 概要

本書では、組込み MPRTOS 向けアプリケーション開発支援ツールのソフトウェアの仕様を記述する。本書は特に、フェーズ 4 におけるリファクタリング作業に関する記述を行う。

2 概要説明

2.1 リファクタリングを実施する理由

フェーズ 4 では、変換・可視化ルールに外部スクリプトを利用できるように TLV を拡張する予定である。現在の TLV ままでは TLV ファイルの可搬性 (ポータビリティ) が問題になるため、リファクタリングを実施する。

現在の TLV は、標準形式トレースログを可視化ルールを用いた変換は、描画処理時に行なわれる。描画処理は TLV ファイルを開くたびに実行されるため、可視化ルールに外部スクリプトを利用した場合、TLV ファイルを別の環境で開くことが難しくなる。そのため、TLV ファイルの可搬性が損なわれる。

可視化ルールを用いた変換を、描画処理時でなくログファイル読み込みに行なうようにリファクタリングを実施する。ログファイル読み込みは、TLV を生成するときのみ実行されるため、TLV ファイルの可搬性が確保できる。

2.2 リファクタリングを実施する対象

リファクタリング対象は、TLV ファイルを生成するクラスと、TLV ファイルを用いて描画処理を行なうクラスである。

TLV ファイル生成時に可視化ルールを用いた変換を行なうように、TLV ファイルを生成するクラスを変更する。

TLV ファイル生成時に可視化ルールを用いた変換を行なうようにしたので、描画処理を行なうクラスでは可視化ルールを用いた変換を行なわないように変更する。

3 作業内容

3.1 リファクタリング前と後の処理の流れ

リファクタリング前の処理の流れを図 1,2 に示す。図 1 はファイル読み込み時の処理を示す。`TraceLogGenerator` を用いて、トレースログファイルを標準形式トレースログに変換している。図 2 は描画処理を示す。`TimeLineEvents` を用いて、標準形式トレースログを可視化ルールを用いて変換し、描画している。

リファクタリング後の処理の流れを図 3,4 に示す。図 3 はファイル読み込み時の処理を示す。`TraceLogGenerator` を用いてトレースログファイルを標準形式トレースログに変換し、`VisualizeShapesGenerator` を用いて、標準形式トレースログを可視化ルールを用いて変換している。図 4 は描画処理を示す。可視化処理済みの標準形式トレースログを描画している。

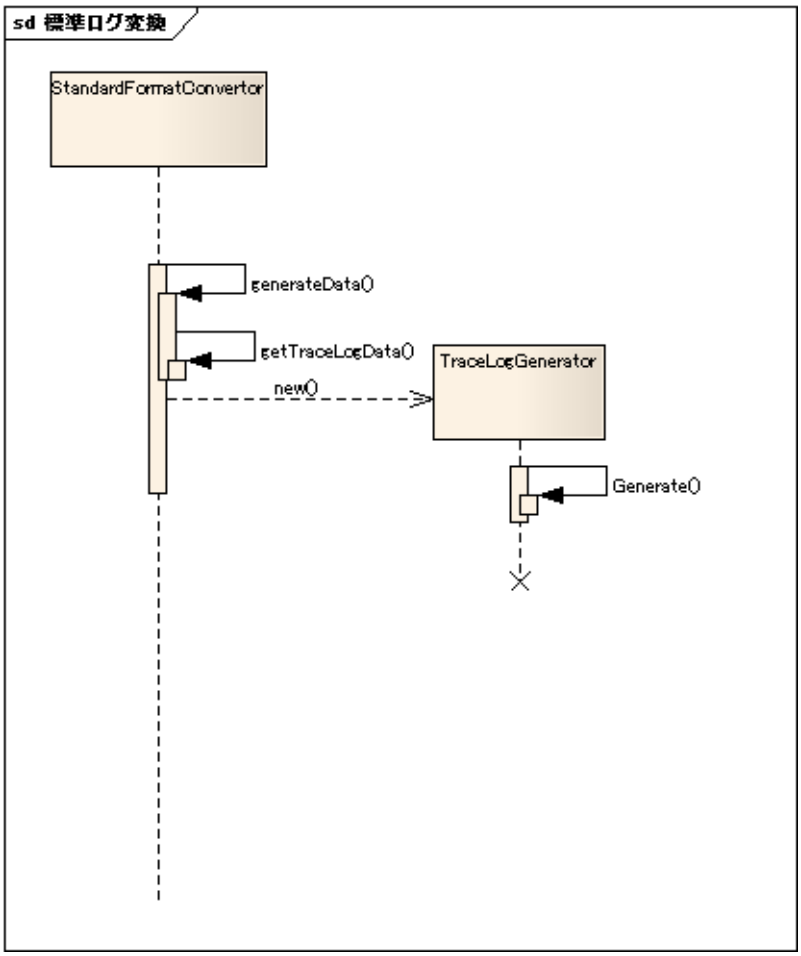

図１　リファクタリング前の標準ログ変換処理

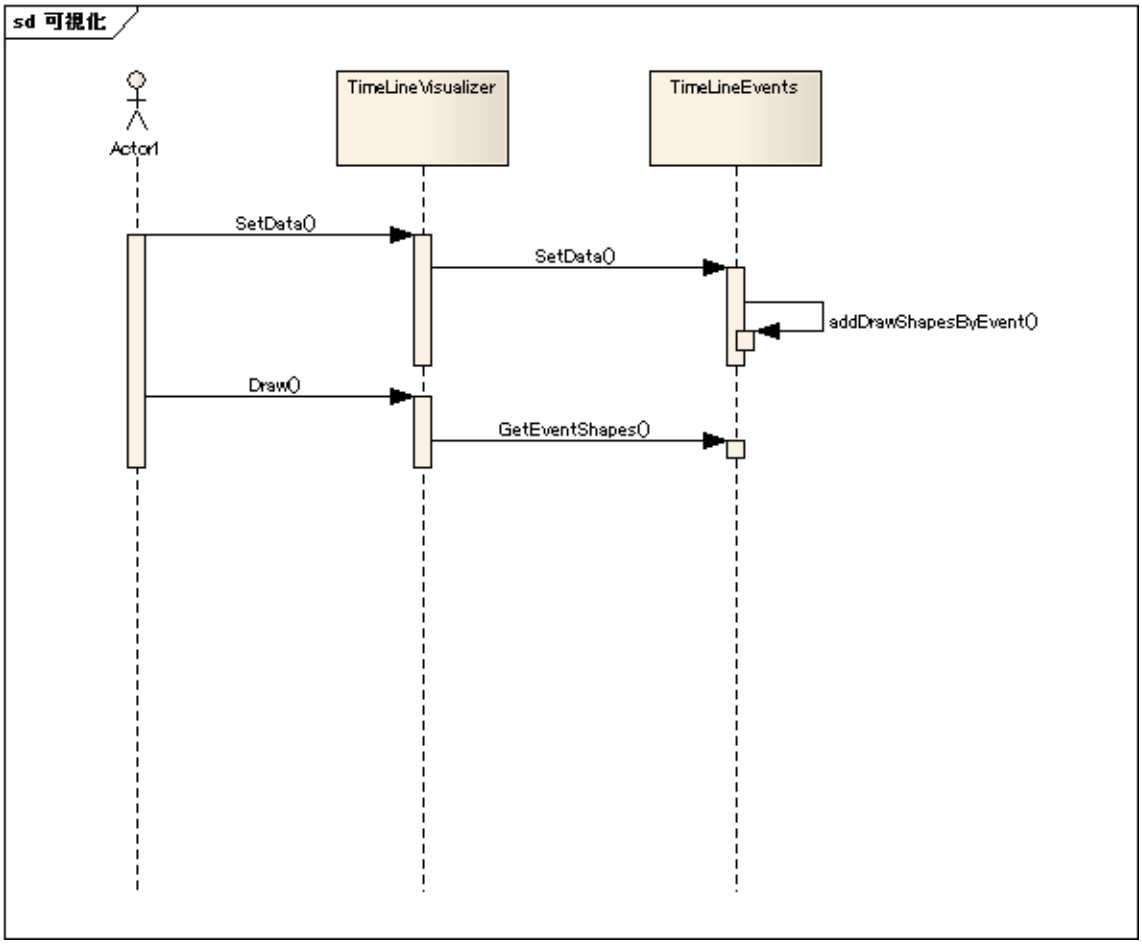

図２　リファクタリング前の可視化処理

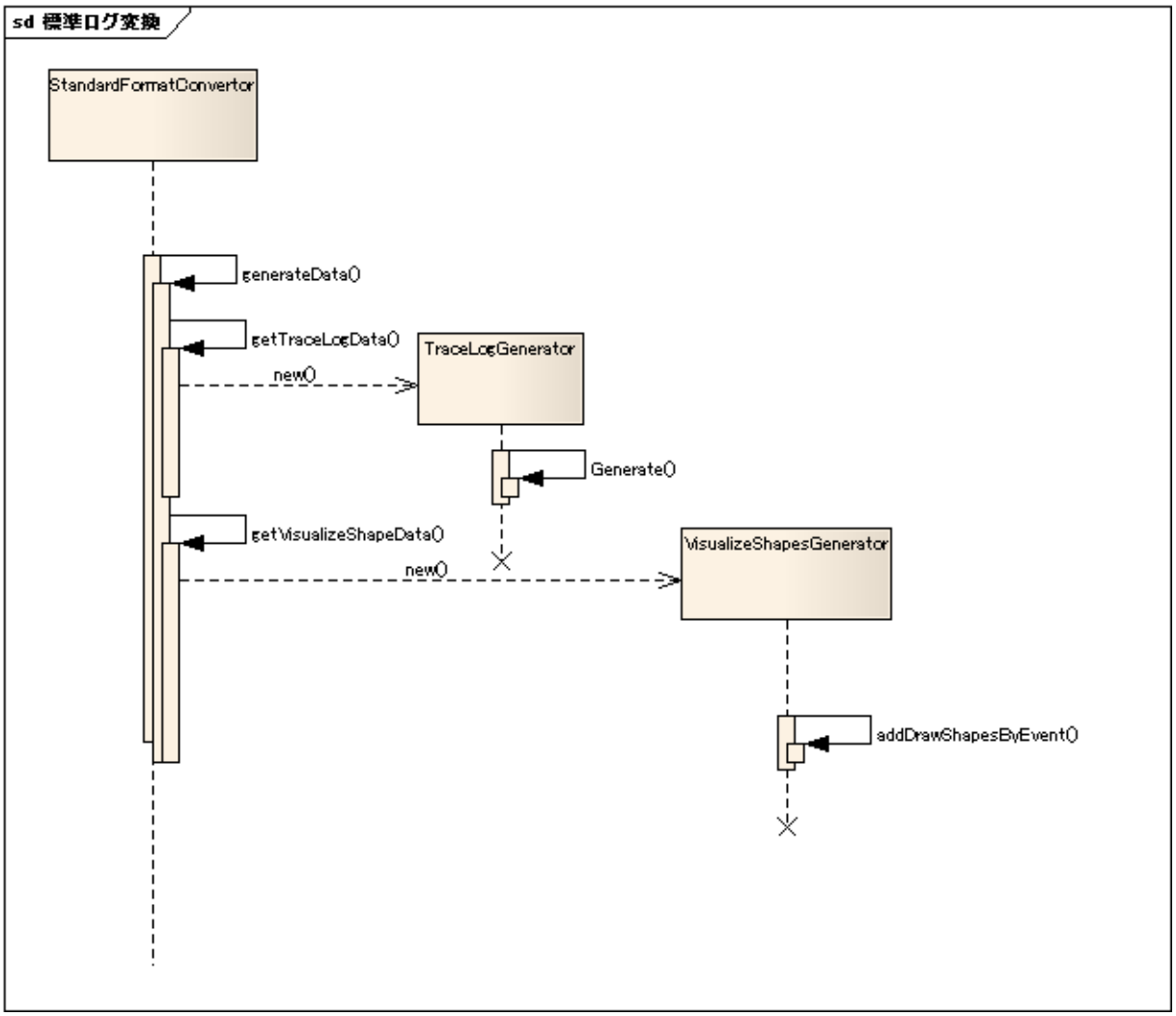

図３　リファクタリング後の標準ログ変換処理

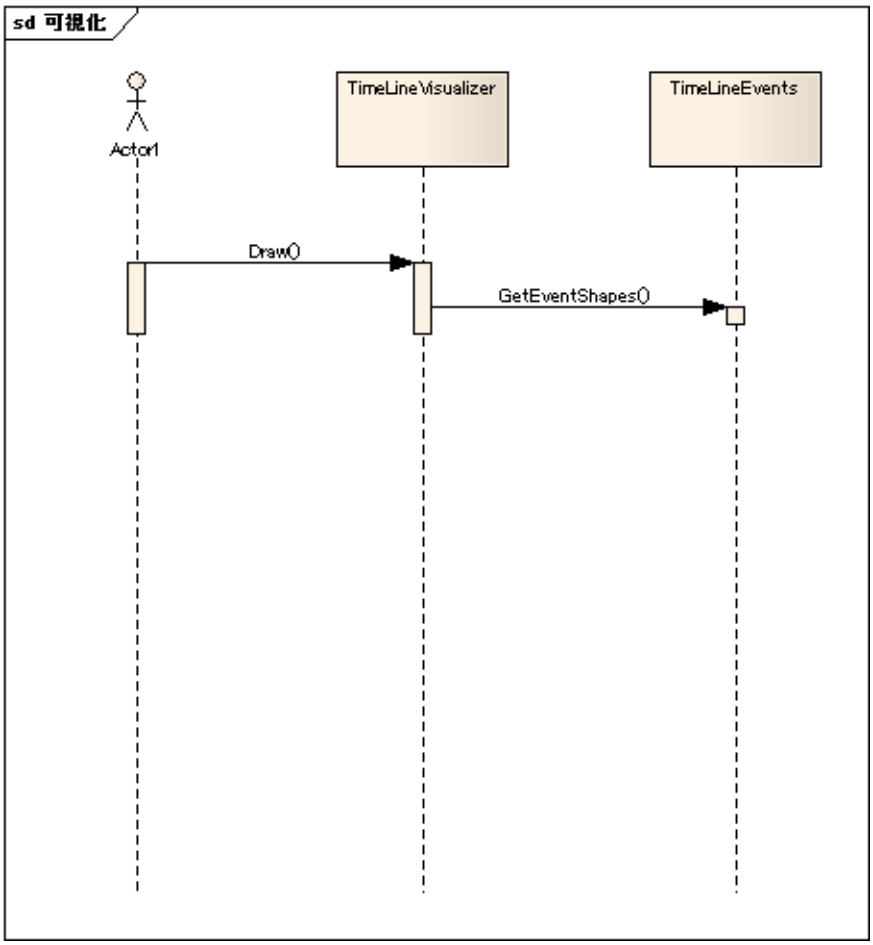

図４　リファクタリング後の可視化処理

3.2 実施内容

リファクタリングは以下の手順で行なう。

1. EventShapes のシリアライズ・デシリアライズ処理
2. 可視化ルールを用いた変換処理の移動
3. 描画処理の修正

3.2.1 EventShapes のシリアライズ・デシリアライズ処理

描画対象である `EventShapes` を JSON 形式で保存できるように、シリアイラズ・デシリアライズ処理を記述する。

1. `ShapeConvertor` を変更し、`Shape` クラスのシリアイラズ・デシリアライズ可能する。
2. `EventShapesConvertor` を作成し、`EventShapes` クラスのシリアイラズ・デシリアライズ可能する。

3.2.2 可視化ルールを用いた可視化処理の移動

`TimeLineEvents` 内にある可視化ルールを用いた可視化処理を独立したクラスに移動する。

1. `TimeLineEvents` の処理の大半を、`VisualizeShapeGenerator` に移動する。

3.2.3 描画処理の修正

`TimeLineEvents` を用いて描画処理を行なっていたクラスを修正する。修正対象は以下のクラスである。

- TimeLineVisualizer(10 箇所)
- TimeLineMacroViewer(2 箇所)
- TraceLogDisplayPanel(5 箇所)

参考文献