

段階的詳細化をサポートする UML 開発環境

設計書

日本電気株式会社

海津 智宏

2008/03/07

1. ユースケース

本システムのユースケースを 図 1 のユースケース図で示す。

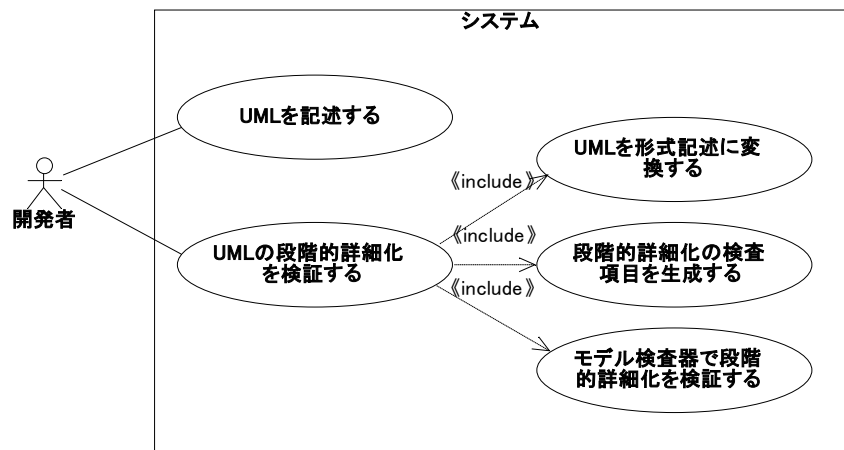


図 1 ユースケース図

このうち、「UML を記述する」は SystemDirector Application Modeler（以下 SDAM）で、「モデル検査器で段階的詳細化を検証する」は FDR2 で実現する。「UML を形式記述に変換する」と「段階的詳細化の検査項目を生成する」を新規に開発する。

2. 全体構成

本システムの全体構成を 図 2 に示す。形式記述には FDR2 の入力である CSP を用いる。
「UML を形式記述に変換する」を実現するための「CSP 変換機能」と「段階的詳細化の検査項目を生成する」を実現するための「CSP 合成機能」を、SDAM と同じ Eclipse 開発環境上に実装する。

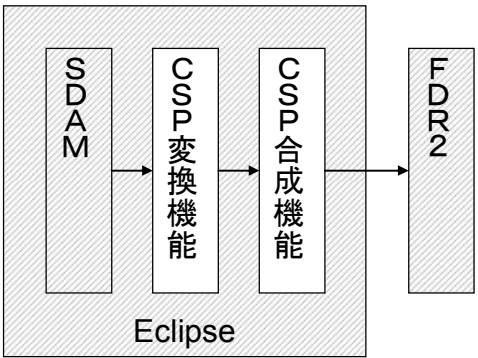


図 2 システムの全体構成

CSP 変換機能の入力は OMG で定められた XMI ファイル、CSP 変換機能と CSP 合成機能の出力は CSP ファイルとする（図 3）。機能仕様は修了製作レポートを参照。

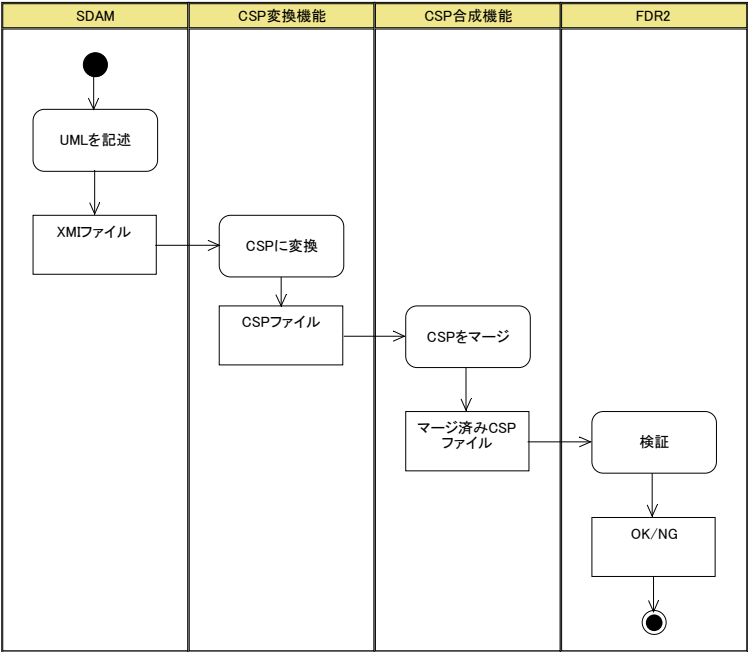


図 3 データフロー

3. クラス設計

本システムのクラス設計を 図 4 のクラス図に示す。実装は 3 個のパッケージと 10 個のクラスからなる。

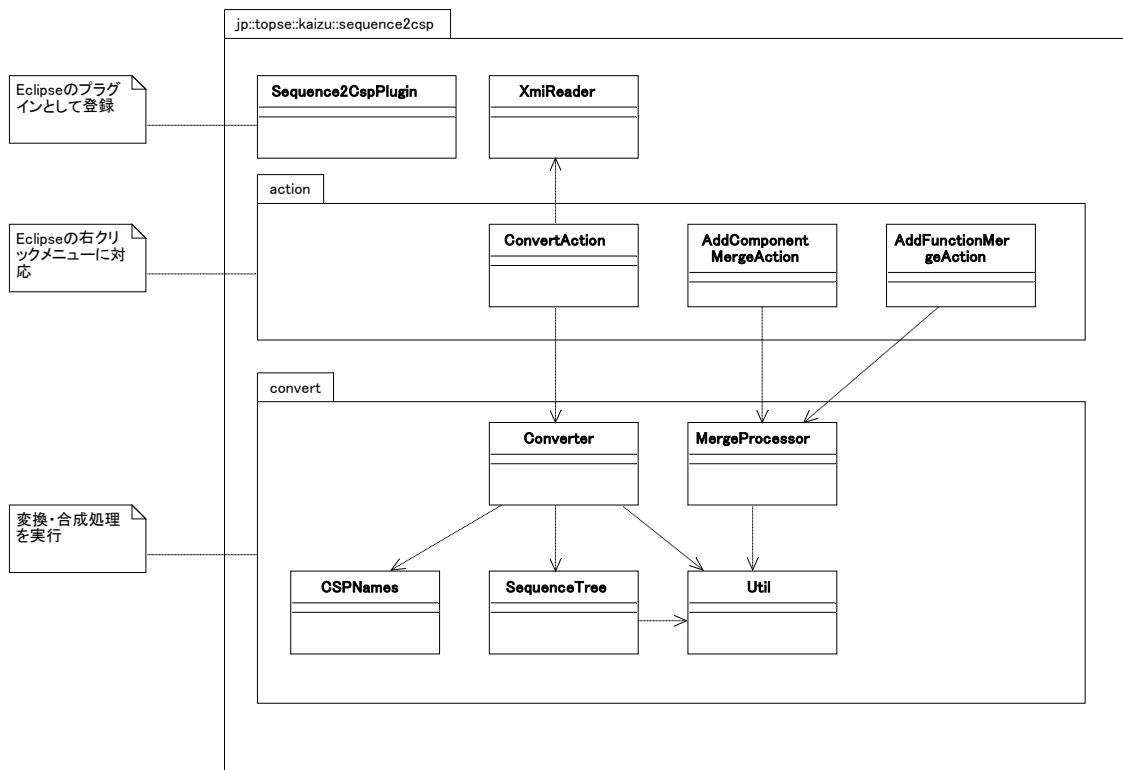


図 4 クラス図

それぞれのクラスの内容を簡単に説明する。

- **jp.topse.kaizu.sequence2csp パッケージ**
システム全体をこのパッケージ内に定義する。パッケージ直下に Eclipse のプラグインクラスを置く。
- **Sequence2CspPlugin クラス**
Eclipse のプラグインとして CSP 変換機能・CSP 合成機能を管理する。
- **XmiReader クラス**
Eclipse Foundation で提供されている UML2 プラグインの機能を利用して XMI ファイルを読み込む。
- **action パッケージ**

Eclipse の右クリックメニューに対応する action を定義する。

- **ConvertAction クラス**

「CSP に変換」メニューを定義する。Converter クラスを呼び出す。

- **AddComponentMergeAction クラス**

「CSP をマージ (コンポーネント分割)」メニューを定義する。MergeProcessor クラスを呼び出す。

- **AddFunctionMergeAction クラス**

「CSP をマージ (機能追加)」メニューを定義する。サンプル実装である。

- **convert パッケージ**

実際の変換・合成処理を実行するためのクラスは convert パッケージ内に置く。

- **Converter クラス**

XMI から CSP への変換を行う。

- **MergeProcessor クラス**

2つの CSP ファイルを入力として合成した CSP ファイルへ変換する。

- **CSPNames クラス**

XMI ファイル中のオブジェクト名やメッセージ名を CSP ファイルの中で利用できる名前に変換する。

- **SequenceTree クラス**

プロセスのふるまいの変換に用いる、メッセージのツリーを保持する。また、ツリーをたどって CSP に変換する機能を持つ。

- **Util クラス**

文字列や集合に関する演算を定義する。

以上