

OCL記述を用いた コンポーネント設計の検証

キヤノン株式会社

和田 明日可

wada.asuka@canon.co.jp

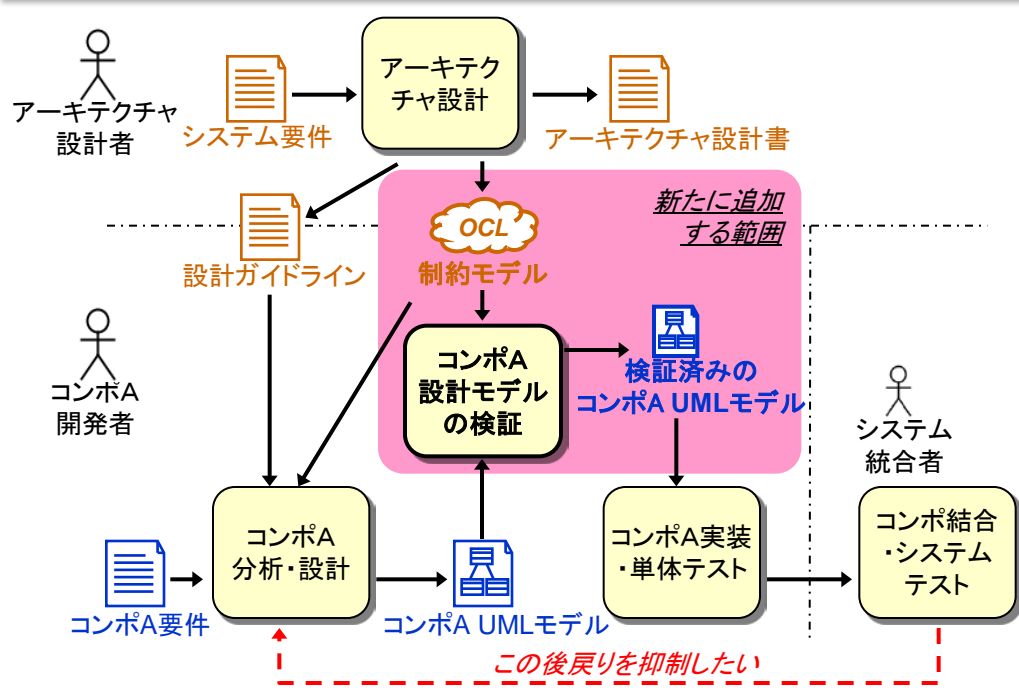
開発における問題点

コンポーネントの着脱によって機能拡張を実現するシステムでは、各コンポーネントに相互の接続を担保するための設計上の制約を課す場合がある。この制約を満たしているか否かを確認せずに開発を進めると、コンポーネント結合テストの段階で大きな後戻りが発生する。

手法・ツールの適用による解決

UMLモデル内の要素に対して制約を与える制約記述言語OCLを用いて、各コンポーネントに課される共通の制約をモデル化する。さらに、コンポーネント毎の設計結果に応じてコンポーネント固有の制約モデルを自動生成し、これを用いて各コンポーネントの設計段階で制約に対する違反の有無を検証する。

目標とする開発プロセスと実現手段



1. コンポーネントの制約のモデル化

要求: 従来ドキュメント(設計ガイドライン)で記述していたコンポーネントの設計上の制約を、厳密に正しく伝達可能な形でモデル化したい。

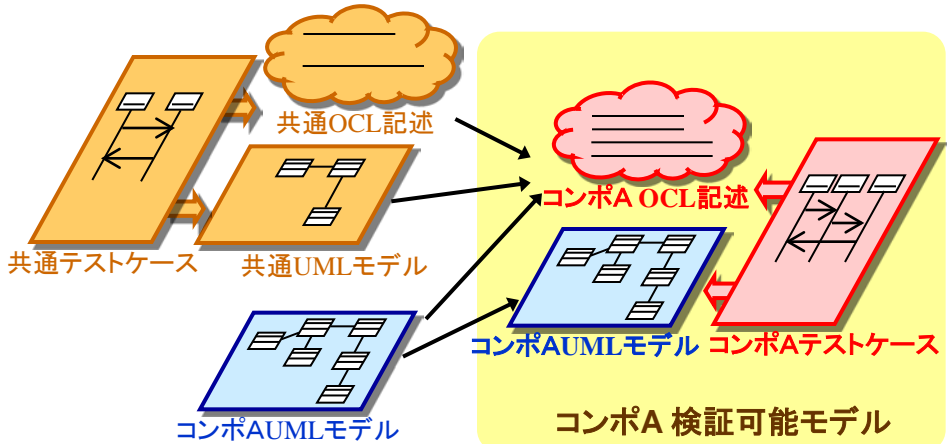
共通UMLモデル & 共通OCL記述 & 共通テストケースによって制約のモデル化を達成した。

2. 設計モデルの検証

要求: コンポーネント開発者の作成した設計モデル(UMLモデル)が設計上の制約を満たしているかを早期・簡易に確認したい。

コンポーネント固有OCL記述 & 固有テストケース生成ツールを作成し、コンポーネント開発者の負担を増加させることなく設計モデルの半自動検証を可能とした。

検証可能モデルの生成



- コンポーネントAの詳細設計結果を検証するために必要な要素
 - 1. コンポーネントA UMLモデル : コンポーネントAの開発者が設計
 - 2. コンポーネントA固有のOCL記述
 - 3. コンポーネントA固有のテストケース } : 自動生成
- 検証にはOCLツールUSE(Univ. Bremen製)を利用

制約モデルの変換例

~設計ガイドラインへの制約記述例~
 ・すべてのストリームはユニークなIDを保持しなければならない(不変条件 : inv1)
 ・ストリーム数は、常に最大ストリーム数以下でなければならない(不変条件 : inv2)
 ・Sleep()を受け付けた後はストリームが存在してはならない(事後条件 : SleepPost1)

