

UML 仕様からのコンポーネントソフトウェア 自動生成ツールの提案と実装*

4 J-4

片山 仁史† 松本 充広‡ 二木 厚吉†

北陸先端科学技術大学院大学情報科学研究科

1 はじめに

近年、ソフトウェア開発において部品化・再利用化、それにとまなう生産性の向上という理由からコンポーネントを組み合わせる手法によるコンポーネントソフトウェア開発法が注目されている。本稿で述べているコンポーネントソフトウェア開発とは、図 1 に示す通りコンポーネントライブラリに登録されているコンポーネントをコネクタを用いて組み合わせることによりソフトウェアを生成する手法である。この手法を有効に用いるには、コンポーネントがどのコンテキストで使われるかを示したモデルとコンポーネント自身の振る舞い等に関して記述されたモデル間で一貫性がとられる必要があるが、実際は、コンポーネント作成者とソフトウェア作成者との間の合意が取られるための手法・ツールが不十分である。そこで今後、設計と設計全体の検証を組み入れた高信頼コンポーネントソフトウェア管理ツールの重要性が高まることが考えられる。本論文では、UML 仕様に記述されたソフトウェアの設計図を基に、各コンポーネントを組み合わせるコネクタを自動生成することによって実用性・信頼性の高いソフトウェア生成を行うツールを議論する。

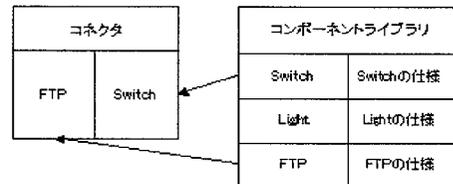


図 1: コンポーネントソフトウェア開発の例

2 システムの特徴・構成

本研究では、コンポーネントソフトウェア管理システム[§], 特にコンポーネントソフトウェアの自動生成部について議論する。本システムでは、ソフトウェアの基本仕様・詳細仕様、コンポーネントの基本仕様、コンポーネントライブラリを管理する。そして仕様間の一貫性検証、ソフトウェアの仕様からのソフトウェアの生成、ソフトウェアの配布を行う。各仕様はカテゴリシス法ベースのコンポーネント開発手法 [松本 01] をもとに設計された XML 表現の UML で記述されているものとする。本システムは主に以下の機能を持つ。

1. UML 図管理機能
XML で表現された UML 図を管理する。
2. UML 図間一貫性検証機能
検証には CafeOBJ 検証系を用いる。UML 図から検証に必要な情報を抽出し、CafeOBJ のコードに変換し検証を行う。
3. コネクタ生成機能
UML 図から生成に必要な情報を抽出し、コンポーネントをつなげるコネクタを生成する。
4. コンポーネントライブラリ管理機能
UML 図で記述されたコンポーネント仕様に対応するコンポーネント、コネクタを管理する。

* An Automatic Code Generator from UML to Component Software

†Yoshihito KATAYAMA(yoshi-k@jaist.ac.jp),
Michihiro MATSUMOTO,Kokichi FUTATSUGI
Japan Advanced Institute of Science and Technology
School of Information Science

‡On leave from First Division,PFU Limited.

§本システムは情報処理振興事業協会 (IPA) の委託により、財団法人ソフトウェア工学研究財団 (RISE) が実施した。平成 13 年度「高度情報化支援ソフトウェアシーズ育成事業」にて、(株)PFU が北陸先端大の協力のもと、作成したものである。なお、第二筆者は、このプロジェクトの研究開発リーダーである。

5. ユーザインタフェース機能

ユーザはこの機能を通じてのみ他の機能にアクセスすることができる。

3 コネクタの生成

本システムではUML図に記述されたクライアントの求める要求とそれを実現するコンポーネントソフトウェアの機能間で一貫性が保証された場合に、コネクタを生成しコンポーネントをつなげる機能を提供する。本研究におけるコンポーネント、コネクタの定義、生成原理を以下に示す。

3.1 コンポーネント

本研究でコンポーネントはJavaBeansとして実装しコンポーネントを組み上げて作られたものもコンポーネントとする。

3.2 コネクタ

本研究におけるコネクタとはコンポーネント間を連結する部分であり、コネクタは生成されるソフトウェアの種類(スタンドアロン型、クライアントサーバ型)によってJavaBeans,Servletを用いて実装する。

3.3 コネクタの生成原理

基本仕様はJavaBeansとなっているコンポーネントに対応し、コンポーネントの組み合わせ方を記述した詳細仕様は、コネクタに対応する(表1)。

詳細仕様における上位のコンポーネントの操作がこれを構成する下位コンポーネントの操作で実現するという構造は、JavaBeansにおける上位のコンポーネントのイベント処理が、これを構成する下位コンポーネントへ委譲することにより実現するという構造に対応する。図2の例で示すように、PUTコンポーネントの持つputというイベントが発生した場合、詳細仕様をもとに、このイベントに対応する下位のコンポーネントFTPに委譲し、FTP内のputを実行する。

詳細仕様とコンポーネントを入力にとると上記の構造の対応関係から詳細仕様をもとにコネクタが自動生成され出力する。

表 1: 仕様とコンポーネント・コネクタの対応表

基本仕様	コンポーネント
変数	プロパティ
オペレータ	メソッド, イベント
詳細仕様	コネクタ
オペレータ	イベント
等式	イベントディスパッチ

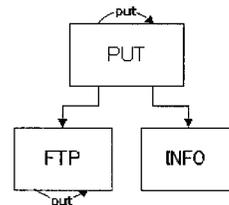


図 2: コンポーネントソフトウェアの構造例

4 まとめ

本稿では、仕様と実装されたコンポーネントとの対応関係を述べ生成原理に関して議論した。今後は、本稿で提案したシステムの実装を行う。コネクタ生成に関しては、同一のマシン上で動くスタンドアロン型の他にクライアント・サーバ型にも対応させていく。また、詳細仕様には、対応関係などデザインレベルが記述され、実装されたコンポーネントと比べると情報量が少ない。

そのため現在の段階では、生成されたコンポーネントソフトウェアは、要求通りにコンポーネントを組み合わせることはできるが、インタフェースの設計は不十分である。これらの点を改善していくことが今後の課題である。

参考文献

- [松本 01] 松本充広, 二木厚吉. コンポーネントソフトウェア開発用軽量フォーマルメソッド, 信学技法,(2001)
- [Krzysztof 99] Krzysztof Czarnecki and Ulrich W.Eisencke, Components and Generative Programming, Software Engineering Notes vol.24(6),pp.2-19,ACM,(1999)
- [Souza 99] Desmond Francis D'Souza and Alan Cameron Wills, Objects,Components,and Frameworks with UML:The Catalysis approach Addison-Wesley,(1999)