

アスペクト指向ソフトウェアにおける アーキテクチャの文書化に関する研究

— 静的構造図検査ツールの設計と実現 —

2004MT040 加藤 敏夫 2004MT061 宮田 和季 2004MT093 柴田 将志

指導教員 野呂 昌満

1 はじめに

本研究室では、組み込みソフトウェアのためのアスペクト指向ソフトウェアアーキテクチャスタイルとして、E-AoSAS++ を提案している。

現在 E-AoSAS++ ではアーキテクチャの記述方法としていくつか手段があるが現在 UML の記法を基準としたアーキテクチャの記述法が提案されている。

UML に記法をもちいている理由として、世界的に広く普及しており直感的な理解が得られやすく、UML のツールも豊富にあるので実用性が高いということが挙げられる。

しかし、E-AoSAS++ の記述法においてセマンティクスの整理が現在行われてなく、また、セマンティクスを検査する方法も現在確立していない。

本研究の目的は、E-AoSAS++ の静的構造図における E-AoSAS++ のセマンティクスの整理、E-AoSAS++ に基づいたアーキテクチャの静的構造図におけるセマンティクス検査を行うツールの開発である。

本研究では、UML ツールで記述した E-AoSAS++ に基づいたアーキテクチャから中間形を生成するツールを作成した、このツールを用いて静的構造図のシンタックスを検査するツールの作成を行い、セマンティクスを検査するツールの作成を行う。

この論文は 2 章・E-AoSAS++、3 章・ツールについて、4 章・考察という構成になっている。

2 E-AoSAS++

本章では E-AoSAS++ に付いて述べる。

2.1 E-AoSAS++

E-AoSAS++ は、組み込みソフトウェアのアスペクト指向ソフトウェアアーキテクチャスタイルである。E-AoSAS++ では、組み込みソフトウェアアーキテクチャを、並行に動作する状態遷移機械の集合として規定している。E-AoSAS++ では、並行に動作する状態遷移機械が互いにメッセージを送り、協調動作を行う。また、E-AoSAS++ では UMP を用いてモジュール化を行っている。

Universal Modularization Pattern(UMP)

UMP とはモジュール化を行う方法であり、1 つの UMP が 1 つの Policy を持ち、Policy が複数の UMP や CSTM を管理、協調動作を行っている。

並行状態遷移機械

並行に動作する状態遷移機械 (以下、CSTM) の構造を表したのが図 1 である。

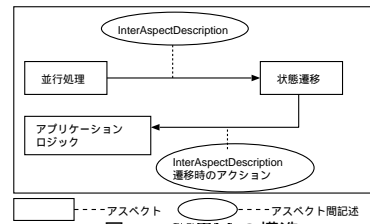


図 1 :CSTM の構造

- 並行処理アスペクト
並行処理に関する処理を行う。
- 状態遷移アスペクト
状態遷移機械を持ち、状態遷移に関する処理を行う。
- アプリケーションロジックアスペクト
状態遷移時のアクション処理中の、データアクセスに関する処理を行う。
- アスペクト間記述 (InterAspectDescription, 以下 IAD)
各アスペクト間の関連を記述する。
- アスペクト間記述、遷移時のアクション状態遷移アスペクトが状態遷移するときのアクションを記述する。

並行状態遷移機械では並行処理アスペクト、状態遷移アスペクト、アプリケーションロジックアスペクトがアスペクト間記述によりメッセージ通信をおこない、協調動作することにより機能を実現している。

2.2 アーキテクチャ記述

E-AoSAS++ のアーキテクチャの記述についての説明を行う。

定義されているステレオタイプ

E-AoSAS++ のアーキテクチャを記述する際に定義されているステレオタイプを以下の表 1 に示す。

以下のステレオタイプは Policy に付与するステレオタイプである。

表 1 定義されているステレオタイプ

BaseMetaModel	ステレオタイプ
Component	<<Aggregation>>
Component	<<IAD>>

ステレオタイプ<<Aggregation>>は Policy がいくつか

の UMP, CSTM に対し管理, 協調動作を行う場合に用いられる。

ステレオタイプ<<IAD>>は Policy が外部の UMP, CSTM にたいしてメッセージを送ることで協調動作を行う場合に用いられる。

また, Policy が<<Aggregation>>, <<IAD>>両方の動作を行う場合は両方のステレオタイプが付与することになる。

2.3 静的構造図の記法

E-AoSAS++ ではアーキテクチャの静的な記述法として, コンポーネント図を用い記述を行っている。図 2 は E-AoSAS++ で用いられているアーキテクチャの例である。

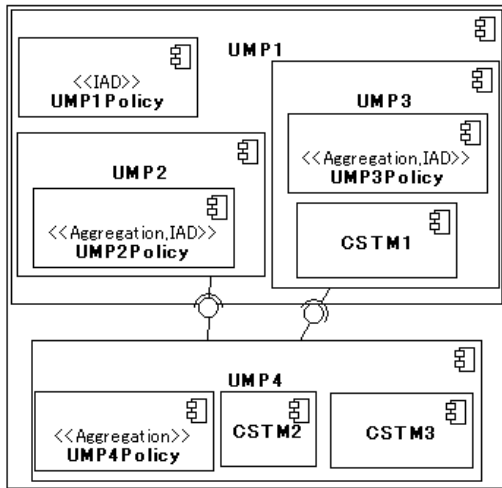


図 2 アーキテクチャの例

UMP 内にステレオタイプ<<IAD>>, <<Aggregation>>が付与しているコンポーネントは Policy である。また UMP 内に記述されているコンポーネントは UMP の Policy が管理しているコンポーネントである。アセンブリコネクタで接続されているコンポーネントの関係は要求インターフェース側が提供インターフェース側のコンポーネントを Policy を用いて管理しているものである。また, Policy が管理する UMP, CSTM の関係を表した図をクラス図として各 Policy ごとに一つのダイアグラムを作成する。Policy の関連を表すクラス図の例が図 3 である。

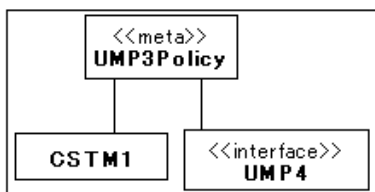


図 3 Policy の関連を表すクラス図 (例)

Policy の関連を表すクラス図では, 各 UMP 内における Policy が管理しているコンポーネントを一つのクラスとして考え, Policy が管理している各コンポーネントの関係を示すものである。

クラス図を用いている理由は関連の表記方法が多くコンポーネント図では明記出来ない関連や集約, 汎化等の図式経路が明記可能だからである。

3 ツールの設計と実現

本研究ではセマンティクス検査ツールを作成する手順として以下の手順をふまえ作成を行うことにした。

1. 入力形式の決定
2. 中間形生成ツールの設計と実現
3. シンタックス検査ツールの設計と実現
4. セマンティクス検査ツールの設計と実現

3.1 入力形式の決定

本研究で作成するセマンティクス検査ツールは, UML ツールに依存しないツールの作成を目的としており, 2 つのファイル形式を入力として考えた。

- XMI ファイル
- XML ファイル

XMI ファイルはメタモデルの交換形式としてつかわれているファイル形式である。しかし, 本研究で作成するツールに対して必要な情報がかけていることから本研究の入力形式としては不適切であった。

次に XML ファイルを入力形式として考えた場合, 本研究で作成するツールに対し必要な情報全てが含まれており, 多くの UML ツールが XML ファイルを出力出来ることから入力形式として XML ファイルを採用した。

3.2 中間形生成ツールの設計と実現

次に入力された XML ファイルから中間形を生成するツールの作成を行った。

まず我々は, 中間形のデータ構造の設計を行った。

まず, アーキテクチャを記述する際にはダイアグラムが用意される。

次に, E-AoSAS++ におけるコンポーネントの役割に付いて, UMP, Policy, CSTM が挙げられ, Policy は UMP が持っているものであり, かつ CSTM の 1 種として考えられる。

最後に, CSTM は Policy に管理されるものである。

以上のことから我々はデータ構造を作成した。

作成しデータ構造を以下の図 4 に示す。

データ構造

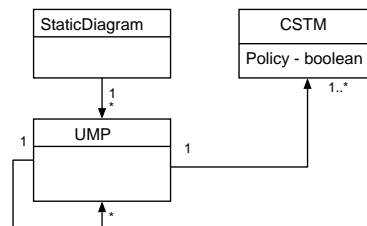


図 4 ツールのデータ構造

静的構造図から必要な情報

次に我々は静的構造図から必要な情報について分析を行った。静的構造図からデータ構造のオブジェクトに対して必要な情報を以下に示す。

1. ダイアグラム名,UMP の名前,CSTM の名前
2. ダイアグラム ID,UMP の ID,CSTM の ID
3. CSTM が Policy であるかどうか
4. どのようなステレオタイプが付与しているか

またデータ構造のオブジェクト間の関係について必要な情報を以下に示す。

5. コンポーネントがどのコンポーネントを入れ子にしているのか。
6. アセンブリコネクタで接続されたコンポーネントの入れ子関係。
7. コンポーネントが E-AoSAS++ でのどの役割なのか。
8. Policy と対応する Policy のクラス図ダイアグラムとの対応関係

1,2,4 で必要な情報に対して、我々は XML ファイルからダイアグラム、コンポーネントごとにオブジェクトを生成し、名前・ID・ステレオタイプ等の情報をオブジェクトに格納を行った。

3 の情報に対して CSTM が Policy かどうかを判断するには以下の条件から判断を行った。

- UMP に入れ子にされている
- 他のコンポーネントを入れ子にしていない
- ステレオタイプが付与している

5 の関係については XML ファイルから、コンポーネント同士の入れ子関係の情報を読み取ることが可能である。

6 の関係についてアセンブリコネクタで接続されているコンポーネントは XML ファイルから、提供側・要求側の ID から対応する ID を検出し提供側のコンポーネントが要求側のコンポーネントによって入れ子にされる。

7 の関係については XML ファイルからダイアグラムは識別が可能である。UMP,CSTM においては Policy を入れ子にしているか、していないかから判別を行った。

8 については Policy の名前とダイアグラムの名前とで対応付けを行った。

最終的なデータ構造を以下の図 5 に示す。

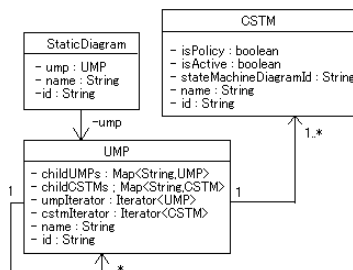


図 5 ツールのデータ構造 2

3.3 データ構造

本ツールを用いて実際にデータ構造を構築した例を以下の図 6 に示す。

まず,StateDiagram が UMP を持ち, その UMP が Policy,UMP,CSTM 等の入れ子にしているものを木構造のようにデータを構築した。

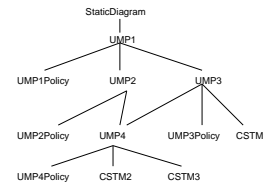


図 6 データ構造 (例)

3.4 シンタックスチェック

本研究で作成した中間形生成ツールをもとに、静的構造図において検査すべきシンタックスを以下に示す。

1. UMP,CSTM,Policy はコンポーネントである
2. UMP に Policy が 1 つある
3. Policy にステレオタイプがある
4. CSTM にアセンブリコネクタを接続する際は常に提供インタフェースが接続される

本研究で作成したシンタックス検査ツールで検査を行うシンタックスエラーについて以下に記述する。

検査方法

- 1) XML の情報から UMP,CSTM,Policy がコンポーネントであるかを検査することで検査を行う。
- 2) UMP のなかで Policy であるコンポーネントを検出し,UMP の条件をみたしているかどうか検査を行う。
- 3) XML の情報から Policy に対しステレオタイプが付属しているかどうかを検出することで検査を行う。
- 4) XML ファイルのアセンブリコネクタの要求側と提供側の ID を検出し、もし要求側のコンポーネントが CSTM である場合にエラーを排出する。

3.5 セマンティクスチェック

静的構造図においてセマンティクスチェックを行うべき項目として,Policy に付与するステレオタイプが正しいものが付与しているかどうかを判別する必要がある。

セマンティクスの整理

セマンティクス検査を行う際に、各ステレオタイプのセマンティクスの整理を行う必要があると考える、我々が各ステレオタイプに対してセマンティクスの整理を行った。

- <<Aggregation>>
複数の CSTM,UMP を管理し、協調動作を行うコンポーネント
- <<IAD>>
外部の UMP や CSTM に対して、メッセージ間通信を行っているコンポーネント

セマンティクス検査を行う方法として、次の方法が考えられる。

Policy に関するステレオタイプのセマンティクス検査として、アーキテクチャの静的構造図と各 Policy に対応するクラス図から Policy に付与すべきステレオタイプの判別を行う方法が考えられる。

Policy のクラス図において Policy が UMP と UMP を接続しているもの、または Policy を含む UMP が外部にある CSTM とメッセージ通信を行っている場合はステレオタイプ<<IAD>>を付与するものであると考える。

また、Policy がいくつかの UMP と CSTM を用いて機能を実現しているものはステレオタイプ<<Aggregation>>を付与するものであると考える。

4 考察

本研究で作成したツール、セマンティクス検査に対して考察を行う。

4.1 セマンティクス検査に関する考察

本研究ではセマンティクス検査まで行うツールは実現が出来なかったが、本研究で提案したセマンティクス検査の方法に付いて考察を行う。

本研究では Policy に付与するステレオタイプのセマンティクス検査において Policy のクラス図からステレオタイプを判別する方法を提案した。この手法において、Policy のステレオタイプの判別は行えると考えられる。しかし、Policy のクラス図においてもシンタックス検査、セマンティクス検査を行う必要があると考えられる。

Policy のクラス図のシンタックス検査

Policy のクラス図において検査すべきシンタックスを以下に示す。

1. Policy にステレオタイプ<<meta>>がついている
2. アセンブリコネクタで接続されているコンポーネントにはステレオタイプ<<interface>>がついている

1,2 についてはアーキテクチャ全体の図からクラスに対して対応しているコンポーネントの情報を検出し、検査が可能であると考えられる。

Policy のクラス図のセマンティクス検査

また、Policy のクラス図において検査すべきセマンティクスを以下に示す。

1. Policy が管理しているコンポーネントでダイアグラムが構成されている。

1 において、アーキテクチャ全体の図から Policy が管理しているコンポーネントの情報を読み込み、Policy が管理してくるコンポーネントでクラス図が構成されているかどうかの検査を行うことで可能であると考えられる。

今後の課題として Policy のクラス図のシンタックス、セマンティクス検査を行う機能を実現し、静的構造図のセマンティクス検査を行うツールの実現を行う必要がある。

4.2 ツールの汎用性に関する考察

我々が今回作成したダイアグラム情報提供ツールでは本研究で用いた UML ツールが出力する XML ファイルの

みにしか対応していないので汎用性はかなり低いという結論にいたった。

ツールの汎用性を高めるのに以下の手法が考えられる。

1. 提案した図式表現のダイアグラムを記述可能なツールの作成。

2. 様々な UML ツールが出力した XML ファイルを読み込み必要な情報を提供出来るツールの作成。

1 の手法を実現するには E-AoSAS++ のアーキテクチャの記述ができ、E-AoSAS++ のセマンティクスを満たしていない図式表現をした場合エラーを出すツールを作成することで実現可能であると考えられる。

2 の手法を実現するには様々な UML ツールが出力した XML ファイルを字句解析、構文解析を行い、統一的な記述に変換し出力を行い、変換したファイルを読み込み必要な情報を提供するツールを作成することで実現すること可能であると考えられる。

5 おわりに

本研究では E-AoSAS++ の静的構造図のアーキテクチャからコードの自動生成に対して情報を提供するツールの作成を行い、静的構造図のシンタックス検査ツールを実現した。次に、静的構造図で検査すべきセマンティクスチェックについて考え、セマンティクスの整理を行った。

最後に、セマンティクス検査に付いての考察、ツールの汎用性の考察も行った

以上のことから、今後の課題であるセマンティクス検査ツールを実現する前段階のツールが実現できた。

今後の課題は、セマンティクスを検査するためのツールの作成が考えられ、その実現に必要な Policy のクラス図におけるシンタックス検査、セマンティクス検査ツールの作成、また汎用性の高いツールの作成が考えられる。

6 謝辞

本研究をすすめるにあたり、熱心な御指導をいただいた野呂昌満教授、有益なアドバイスをくださった沢田篤司教授、蜂巢吉成講師、大学院生の皆さんに深く感謝致します。また 2 年間共にがんばってきた野呂研究室、沢田研究室、蜂巢研究室のみなさんにも感謝致します。

参考文献

- [1] Object Management Group 著, 西原 裕善 監訳 : UML2.0 仕様書, (2006)
- [2] 株式会社オーガス総研 オブジェクトの広場編集部 : その場で使えるしっかり学べる UML2.0, (2006)
- [3] Brett McLaughlin 著, 須賀 裕治 監訳 : Java&XML, (2002)
- [4] SparxSystemJapan : EnterpriseArchitect, <http://www.sparxsystems.jp/ea.htm> (2007.12 accessed)