

Title	UML図面要素間の依存関係の自動生成法
Author(s)	金井, 健太郎
Citation	
Issue Date	2009-03
Type	Thesis or Dissertation
Text version	author
URL	http://hdl.handle.net/10119/8146
Rights	
Description	Supervisor:落水 浩一郎, 情報科学研究科, 修士

修 士 論 文

UML 図面要素間の依存関係の自動生成法

北陸先端科学技術大学院大学
情報科学研究科情報システム学専攻

金井 健太郎

2009 年 3 月

修 士 論 文

UML 図面要素間の依存関係の自動生成法

指導教官 落水 浩一郎 教授

審査委員主査 落水 浩一郎 教授

審査委員 鈴木 正人 准教授

審査委員 岸 知二 特任教授

北陸先端科学技術大学院大学
情報科学研究科情報システム学専攻

710018 金井 健太郎

提出年月: 2009 年 2 月

概要

本研究では、UML 図面間の変更波及解析の対象を UML1.5 版から UML2.0 版に拡張する。小谷が提案した基本依存関係は UML2.0 版でも利用可能であり、基本依存関係を自動生成する依存関係生成モデルに改良を加えることで、UML2.0 版における基本依存関係を自動生成法を提案する。また、小谷の研究において変更波及解析不能であった概念の分解に関して、変更波及解析をできるように基本依存関係の自動生成法を一部改良する。基本依存関係を自動生成できない部分に関して、設計者に基本依存関係を付加させることで対処する。エレベータ制御システムの事例研究を対象にして実験し、提案手法の有効性を示す。

目次

第1章	はじめに	1
1.1	背景と目的	1
1.2	本論文の構成	3
第2章	依存関係生成モデルによる依存関係の自動生成	5
2.1	基本依存関係	5
2.2	依存関係生成モデル	5
2.2.1	照合規則	6
2.2.2	付加規則	6
2.2.3	選択規則	9
2.3	依存関係の自動生成	9
2.4	ツール	10
2.4.1	中間成果物間の基本依存関係生成機構	11
2.4.2	中間成果物の読み込み	12
2.4.3	フレームワークの定義	12
2.4.4	プラグイン構造	13
2.5	課題	13
2.5.1	UML2.0 への対応	14
2.5.2	概念の分解への対応	14
第3章	依存関係生成モデルの改良	17
3.1	UML2.0 への対応	17
3.1.1	UML2.0 の分析	17
3.1.2	照合規則	18
3.1.3	付加規則	20
3.2	概念の分解への対応	23

第4章	実装	25
4.1	依存関係生成層	25
4.2	成果物リーダ層	25
第5章	評価	27
5.1	実験対象の特徴	27
5.2	評価	27
5.3	考察	29
第6章	結論	30
6.1	本研究のまとめ	30
6.2	今後の課題	30
付録A	UML2.0 版対応の照合規則	32

目 次

1.1	依存関係の表記	1
1.2	基本依存関係の自動生成法	2
1.3	変更支援ツール	3
2.1	付加規則	8
2.2	依存関係の自動生成例	10
2.3	変更波及解析法の概要	11
2.4	フレームワーク	12
2.5	プラグインの依存関係	14
2.6	ユースケース図	15
2.7	クラス図	16
3.1	概念の分解できる例	23
3.2	概念の分解の解析ミス	24
4.1	フレームワーク (再掲)	25
4.2	プラグイン構造	26

表 目 次

2.1	照合規則	7
2.2	生成モデル要素	8
2.3	選択規則	9
3.1	図の分類	17
3.2	UML2.0 版に追加した照合規則	21
3.3	UML2.0 版対応の生成モデル要素	22
5.1	評価結果	29
A.1	UML2.0 版対応のすべての照合規則	32
A.1	UML2.0 版対応のすべての照合規則	33

第1章 はじめに

1.1 背景と目的

ソフトウェアの開発工程では、要求定義書、設計図、ソースコードなど様々な成果物が作成され、開発工程が進むにつれその数は膨大になる。そのため、開発者自身が成果物間の関連を十分に把握することは困難である。このような状況下で、これらの成果物は、開発、テストや保守時に、仕様変更やバグなどの発生にともない頻繁に変更される。この変更作業において、ある成果物 A が変更されるとこれに依存する成果物 B にも変更が波及する可能性がある。しかし、それらを膨大な数の成果物の中から探し出す際には次のような問題が発生する。

- 成果物間の関連の認識不足によって、変更すべき成果物を把握しづらい
- 開発者とは別の成果物間の関連を十分に把握していない保守担当者によって変更作業が行われる

このような現状を踏まえ、小谷 [1] は、UML 図とその構成要素 (以下、UML 図式要素

The diagram shows a rectangular box containing the text "ターゲット ←----- ソース". The text is centered and the box has a thin black border.

図 1.1: 依存関係の表記

小谷は、UML1.5 版を対象に 2 章で説明する変更波及解析に利用でき、かつ、自動生成可能な基本依存関係を、ターゲットとソースの間の存在従属と共有情報に注目した分析を

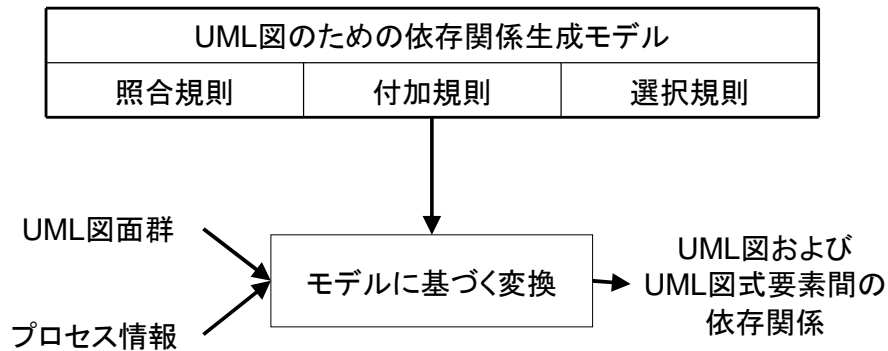


図 1.2: 基本依存関係の自動生成法

図 1.2 は、UML 図面群とプロセス情報を入力し、依存関係生成モデルを利用して、UML 記述間の基本依存関係を出力する手法である。ここでプロセス情報とは、開発方法論に依存する情報であり、開発方法論を構成するフェーズ、フェーズの実行順序、各フェーズに含まれる UML 図面群を定義したものである。

自動生成の中核となる依存関係生成モデルは、照合規則、付加規則、選択規則で構成される。照合規則は、依存関係が付加可能である UML 記述の組み合わせを取り出す規則である。付加規則は、UML 記述間に接続可能なすべての基本依存関係の型を定義した規則である。選択規則は、プロセス情報を用いて、適切な基本依存関係の型を選択する規則である。

小谷 [1] は、以上の依存関係生成モデルを用いて、UML 記述から基本依存関係を自動生成する変更波及解析ツールを実装した。図 1.3 は変更波及解析ツールの実行例である。これは対象となる文書として UML1.5 版によって描かれた図面を対象として、依存関係を自動抽出している。

現在 UML2.0 版 [3][4] が公開され、ソフトウェア開発現場でも利用され始めている。本研究では、変更波及解析の対象を UML1.5 版から UML2.0 版へ拡張することを目的とする。UML1.5 版での依存関係生成モデルの照合規則と付加規則を UML2.0 版へ拡張することで、基本依存関係を自動生成でき、UML2.0 版で変更波及解析をできる。

また、小谷の研究では設計において概念の分解¹が生じるとき、基本依存関係を自動生成できていない例 (2.5.2 節参照) がある。これに対し、本研究では、UML 記述の一部に設

¹概念の分解に関しては 2.5.2 節で説明する。

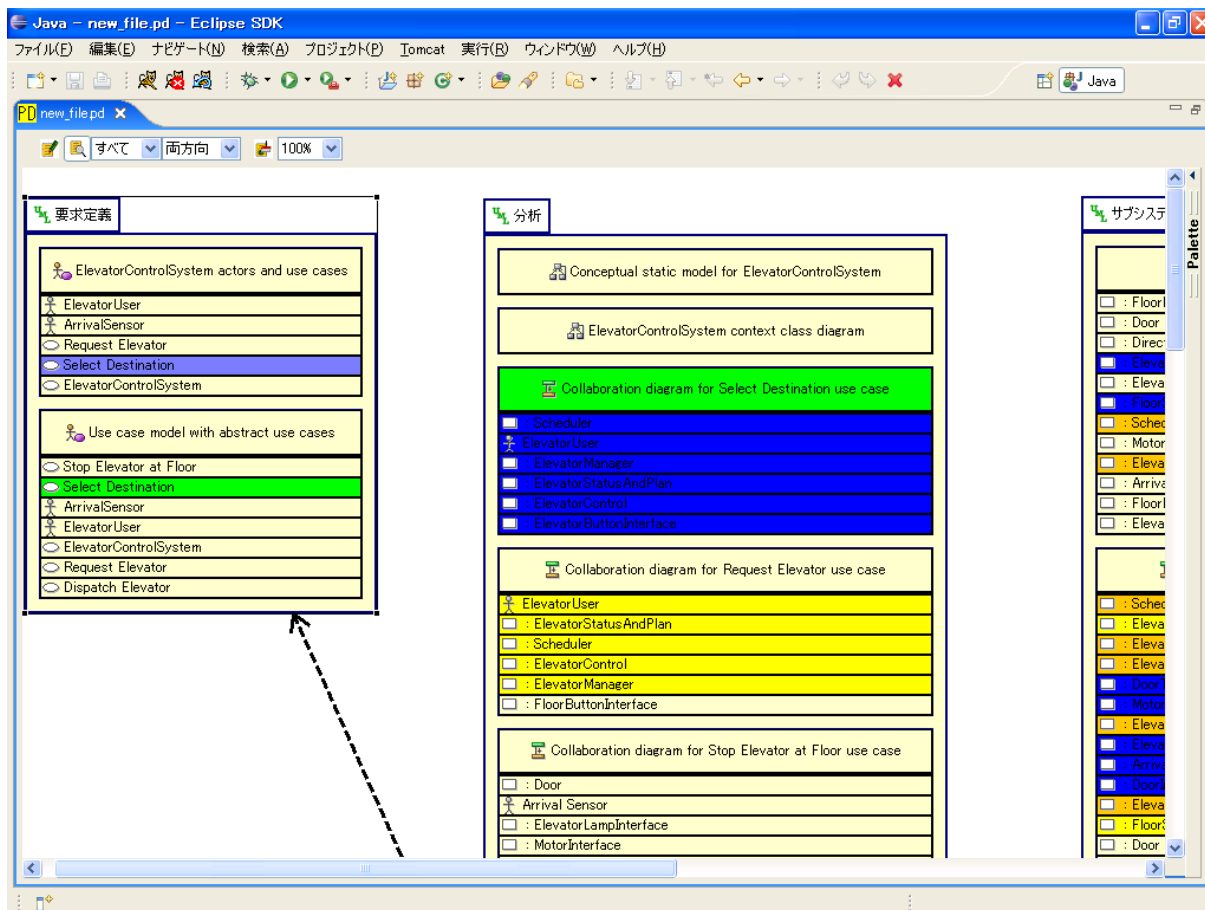


図 1.3: 変更支援ツール

計者の手で依存関係を付加させることで対処する。

1.2 本論文の構成

本論文の構成を以下に示す。

- 2章では、小谷 [1] による成果と問題点を提起する。
- 3章では、2章で挙げた問題点に対し、本研究ではどのようなアプローチをとるか説明する。
- 4章では、変更波及解析支援ツールについて説明する。
- 5章では本研究に対して実験評価を行う。

- 6章で本研究をまとめる。

第2章 依存関係生成モデルによる依存関係の自動生成

この章では、小谷 [1] により提案された、変更波及解析に有用な基本依存関係の自動生成法に関して説明する。次に、この基本依存関係を利用して、実際に変更波及解析を行うツールについて説明する。それらをふまえて、小谷の研究における問題点を明確にする。

2.1 基本依存関係

小谷 [1] は UML メタモデルで定義された依存関係から 4 種類の基本依存関係を与えた。以下で 4 種類の基本依存関係の自動生成法を示す。

- 生存従属：名前が比較可能であるか、または包含関係が解析可能である場合に自動生成可能である
- 情報共有：共有される情報群の少なくとも一つの名前が比較可能である場合に自動生成可能である
- コピー：UML 図式要素が同一フェーズ内に存在し、かつ、名前と型が一致するときに自動生成可能である
- 同一概念：同じ概念には類似する名前が与えられるという前提とプロセス情報の存在を前提として自動生成可能である

2.2 依存関係生成モデル

UML 記述を与え、変更波及解析に有用な基本依存関係に変換するのが依存関係生成モデルである。依存関係生成モデルは照合規則、付加規則、選択規則で構成される。

2.2.1 照合規則

UML メタモデルの依存関係は、変更に関する本質的な情報を定義しているが、UML 記述に直接表現されない場合が多い。そこで依存関係を付加する UML 記述の組み合わせを探す規則を定義する必要がある、これを照合規則として形式化する。その手がかりとして、2.1 節において、基本依存関係の自動生成に利用する情報として挙げた、UML 記述の名前と UML 記述の包含関係を利用する。これらを用いて、ターゲット (T) とソース (S) の比較する情報を定義した、5 種類の照合条件を定義する。

- **Contained** : T の名前が、S の名前に含まれる
- **Similar** : T の名前は S の名前に似ている
- **TypeSim** : T の型が、S の名前に似ている
- **SimType** : T の名前が、S の型に似ている
- **Include** : T は S を包含する。すなわち、UML 図とそれを構成する UML 図式要素や、UML 図式要素とその内部要素の関係

表 2.1 に照合規則を示す。左から順に、依存関係のターゲットの UML 記述の型、ソースの UML 記述の型、その組み合わせの照合条件を示す。

2.2.2 付加規則

2.1 節で定義した 4 種類の基本依存関係は、その両端に設定可能な UML 記述の型に関して制限がある。この制限は開発方法論に依存する。ここでは開発方法論 Unified Process を採用した。基本依存関係の両端にくる UML 記述の種類を分類した結果を表 2.2 に示す。表 2.2 において、基本依存関係の両端に設定可能な UML 記述の型を生成モデル要素と呼ぶことにする。このとき、生成モデル要素間に 4 種類の基本依存関係の一部を設定することが出来る。生成モデル要素間にどのような基本依存関係が付加できるのかを定義したものを付加規則と呼ぶ。図 2.1 に付加規則を示す。図 2.1 において、長方形は生成モデル要素を示し、4 種類の矢印は各種基本依存関係を示す。

表 2.1: 照合規則

ターゲット	ソース	照合条件
ユースケース図	アクター	Include
ユースケース図	ユースケース	Include
ユースケース	クラス図	Contained
ユースケース	状態チャート図	Contained
ユースケース	アクティビティ図	Contained
ユースケース	コラボレーション図	Contained
ユースケース	シーケンス図	Contained
ユースケース	ユースケース	Similar
アクター	アクター	Similar
アクター	クラス	Similar
アクター	オブジェクト	TypeSim
クラス図	クラス	Include
クラス図	パッケージ	Include
パッケージ	クラス	Include
パッケージ	パッケージ	Similar
クラス	パッケージ	Similar
クラス	オブジェクト	SimType, Contained
クラス	状態チャート図	Contained
クラス	アクティビティ図	Contained
クラス	クラス	Similar, Include
オブジェクト図	オブジェクト	Include
オブジェクト	クラス	TypeSim
オブジェクト	状態チャート図	TypeSim
オブジェクト	アクティビティ図	TypeSim
オブジェクト	オブジェクト	Similar, Include
コンポーネント図	コンポーネント	Include
コンポーネント	コンポーネント	Similar
配置図	ノード	Include
ノード	ノード	Similar
状態チャート図	状態	Include
状態	状態	Similar, Include
アクティビティ図	アクション状態	Include
アクション状態	アクション状態	Similar, Include
コラボレーション図	オブジェクト	Include
シーケンス図	オブジェクト	Include

表 2.2: 生成モデル要素

生成モデル要素	UML 図または図式要素
Classifier 要素	アクター、ユースケース、クラス、パッケージ、ノード、コンポーネント、オブジェクト (オブジェクト図)
関係要素	関連、依存、集約、汎化、リンク
状態要素	状態、アクション状態
遷移要素	遷移、イベント、アクション
インスタンス要素	オブジェクト (コラボレーション図、シーケンス図)
メッセージ要素	メッセージ
関係図	ユースケース図 クラス図 オブジェクト図

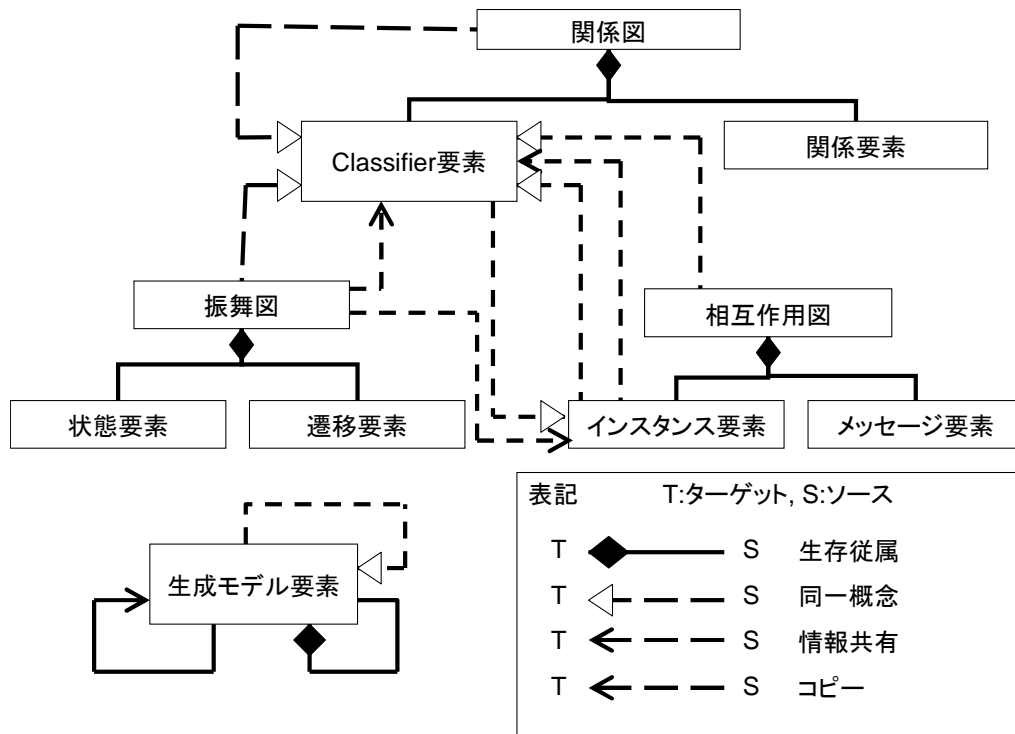


図 2.1: 付加規則

2.2.3 選択規則

付加規則において、いくつかの生成モデル要素の組に数種類の基本依存関係が設定された。一方で、4種類の基本依存関係にはフェーズ、図面、UML 記述の型、名前を用いて排他的な基準が存在する。そのため、基本依存関係を付加する場合、設定された数種類の中から適切なものだけを付加することになる。これらの基準を用いて、各 UML 記述の組み合わせに適切な種類を選択する規則として選択規則を定義し、表 2.3 に示す。

表 2.3: 選択規則

		フェーズ					
		同じ	隣接	離れている			
UML 記述の型	同じ	生存従属	コピー	同一概念	-	同じ	名前
	異なる		-			異なる	
			情報共有			同じ	
		同じ	異なる				
		図面					

2.3 依存関係の自動生成

2.2節で定義した依存関係生成モデルを利用して、依存関係を自動生成する過程を簡単な例を用いて説明する。例として、クラス”ElevatorControl”とオブジェクト”:ElevatorControl”の間に基本依存関係を自動生成する過程を以下に述べ、図 2.2 に示す。ただし、二つの UML 記述は同じフェーズに存在するとする。

1. UML 記述の組み合わせの抽出 照合規則により、ターゲットをクラス、ソースをオブジェクトとする規則「Contained」を適用する。
2. 生成モデル要素の検索 表 2.2 より、クラスとオブジェクトの生成モデル要素はそれぞれ Classifier 要素とインスタンス要素に分類されている。

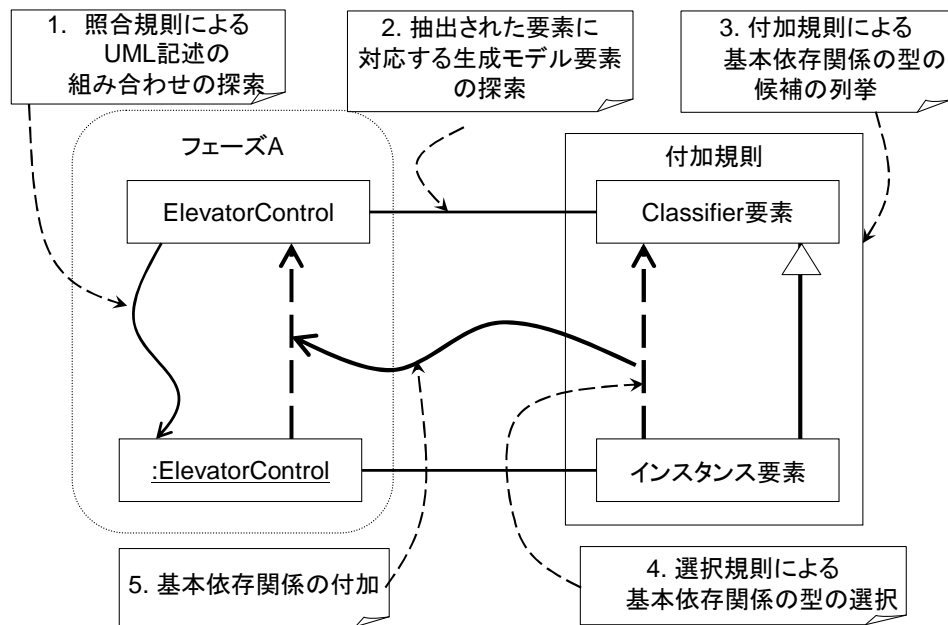


図 2.2: 依存関係の自動生成例

3. 基本依存関係の型の列挙 図 2.1 の付加規則より、Classifier 要素をターゲット、インスタンス要素をソースとする基本依存関係の情報共有と同一概念を、両要素をつなぐ基本依存関係の候補として列挙する。
4. 型の選択 2つの図面が同一フェーズで作成されたことを示すプロセス情報から、選択規則より「情報共有」を選択する。
5. 基本依存関係の付加 基本依存関係「情報共有」を両UML記述間に付加する。

2.4 ツール

本節では、小谷 [1] が提案した手法を実装した変更波及解析ツールと、そのフレームワークについて説明する。依存関係の追跡による変更波及解析は、UML 図に限らず、様々な種類の間接成果物に適用可能であると考え、様々な種類の基本依存関係を自動生成し変更波及解析を行うツールのフレームワークを定義し、Eclipse[9] プラグイン機構を用いて実装した¹。読み込み可能なUMLモデリングツールのファイルは、PatternWeaver[7]で作られたファイルである。

¹Eclipse3.2.2 版上で稼動し、GEF3.2.2 版プラグイン [10] を必要とする。

基本依存関係を用いた変更波及解析は、図 2.3 に示すように、以下の 2 ステップで行わ

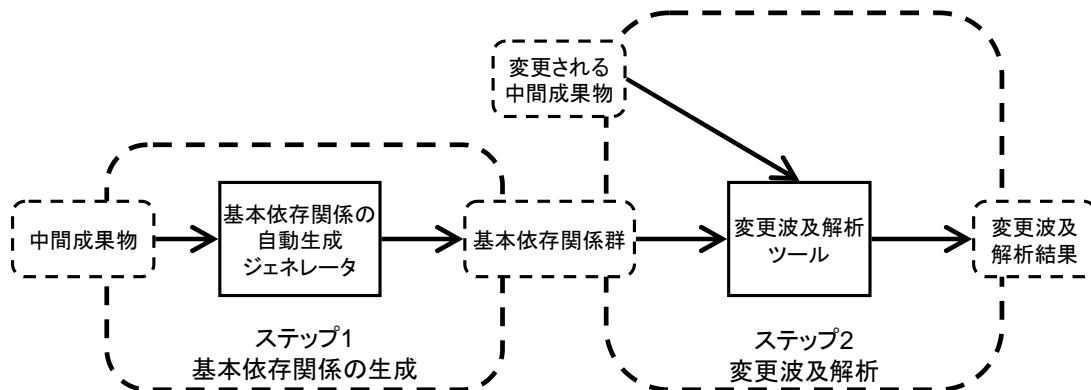


図 2.3: 変更波及解析法の概要

ステップ 1 では、中間成果物の各種類に適した手法を用いて、中間成果物間の基本依存関係を生成する。ステップ 2 では、基本依存関係の追跡による変更波及解析を行う。このため、基本依存関係の両端にくる中間成果物の種類には依存しない。

図 2.3 で示したステップ 1 における基本依存関係の自動生成を考察し、本ツールの拡張ポイントを検討した。これを用いて、中間成果物の種類に依存しない変更波及解析ツールのフレームワークを定義した。

2.4.1 中間成果物間の基本依存関係生成機構

中間成果物の種類は、ここまでの議論の対象であった UML 図のほかにそれを実装したソースコードなどがある。中間成果物の内容は、各仕様に準拠したものであるため、仕様ごとに基本依存関係の自動生成機構が必要となる。図 2.3 では、「基本依存関係ジェネレータ」が自動生成機構に対応する。任意の中間成果物の仕様に対応するため、このジェネレータが本ツールの拡張ポイントとなる。

2.4.2 中間成果物の読み込み

中間成果物の内容はその仕様に準拠しているが、それをファイルに出力する仕様(以下、記録仕様と呼ぶ)は、‘中間成果物の仕様’や‘開発ツール’の組み合わせによって異なる。中間成果物は、各‘開発ツール’を利用して作成される。例えば、UML1.5版の図に対応したモデリングツールにはPatternWeaver[7]やBorland®Together®[8]などがあり、Java言語プログラムを記述する支援ツールはテキストエディタやEclipseなどがある。これら開発ツールからの記録仕様に従って出力されたものが、中間成果物となる。各開発ツールが用いる記録仕様の定義は、中間成果物の仕様そのもの、または、開発ツールによる定義のどちらかである。

以上より、中間成果物の記録仕様は‘中間成果物の仕様’と‘開発ツール’の組み合わせによって異なる。さまざまな出力仕様に対応できるように、各組み合わせによって作られた中間成果物を、ツールが読み込み可能なフォーマットへ変換するコンバータを作成する必要がある。そのため、変更波及解析ツールは中間成果物のコンバータを持つ必要があり、この中間成果物のコンバータをツールの拡張ポイントとする。

2.4.3 フレームワークの定義

2.4.2節より、以下の2種類の拡張ポイントを得た。

- 中間成果物の仕様に適した基本依存関係ジェネレータ

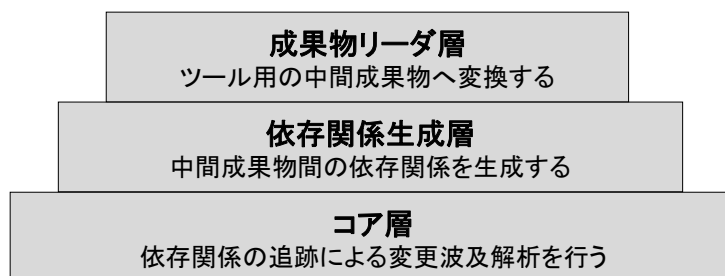


図 2.4: フレームワーク

- 最下層:コア層 基本依存関係の追跡による変更波及解析
- 中間層:依存関係生成層 本ツール用の中間成果物間の基本依存関係の生成
- 最上層:成果物リーダ層 中間成果物から本ツール用の中間成果物の生成

このツールで拡張可能な機能は、“依存関係を生成可能な中間成果物の種類の追加”と、“読み込み可能なファイルの種類追加”である。“依存関係生成層”は、依存関係を生成可能な中間成果物の種類を決定し、“成果物リーダ層”は、読み込み可能なファイルの種類を決定する。

2.4.4 プラグイン構造

Eclipse プラグインは拡張機構を持っており、それを利用することで Eclipse の機能を追加できる。本ツールは Eclipse の拡張機構を利用し、2.4.3 節のフレームワークに基づいて、図 2.5 のように 3 種類の Eclipse プラグインを実装した。

- コア層 変更波及解析プラグイン
- 依存関係生成層 UML 図の基本依存関係生成ツールプラグイン
- 成果物リーダ層 PatternWeaver ファイルリーダプラグイン

コア層に対応するプラグインとして変更波及解析ツールプラグインを作成した。このプラグインは、変更波及解析・表示機能を持つ。

依存関係生成層に対応するプラグインとして、UML 図の基本依存関係生成ツールを実装した。このプラグインは、フェーズに

2.5 課題

本研究では以下の 2 点の課題に取り組んだ。

- UML2.0 への対応
- 概念の分解への対応

それぞれの課題について、2.5.1 節、2.5.2 節で説明する。

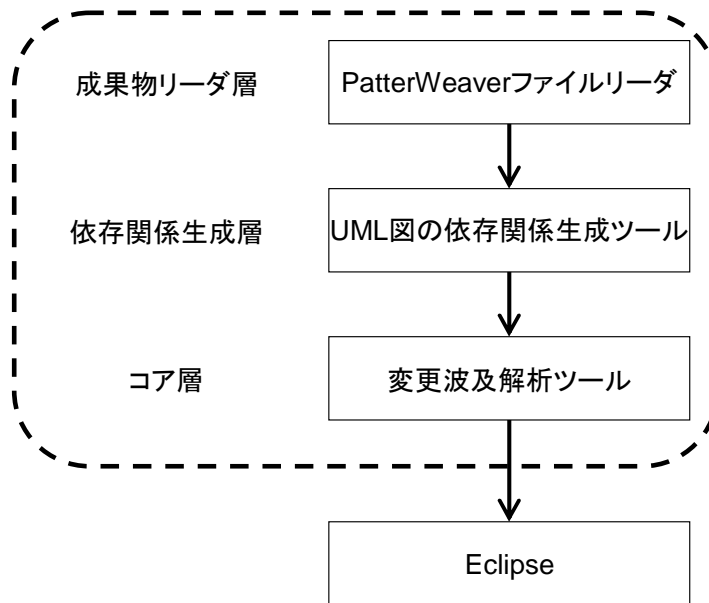


図 2.5: プラグインの依存関係

2.5.1 UML2.0 への対応

小谷の研究 [1] では、変更波及解析の対象としている成果物として、UML1.5 版を対象としている。現在 UML2.0 版 [3][4] が公開され、ソフトウェア開発現場でも導入され始めている。

UML2.0 版では、組込みシステム開発に対応するために、細かい時間間隔での状態遷移や時間制約、メッセージの送受信などを表すモデルを描くことができるタイミング図が新たに定義された。

また、コンポーネントベース開発に対応するため、UML2.0 版ではクラスやコンポーネントの内部構造と外部の境界を明確に表現することができるようになった。

これら UML2.0 版で新しく追加された UML 記述や定義され直した UML 記述に対応することが課題である。

2.5.2 概念の分解への対応

図 2.6、図 2.7 は文献 [6] でケーススタディとして、エレベータ制御システムを UML を用いて設計した図面の一部である。図 2.6 のユースケース図の図式要素に” Select Desti-

nation”ユースケースという概念²がある。これに対し、図 2.7 はエレベータ制御システムの問題領域における現実世界をクラスとして表現したクラス図である。“Select Destination”ユースケースのユースケース記述より、図 2.7 の中の“Floor”、“Elevator”、“ArrivalSensor”、“ElevatorButton”、“Motor”、“Door”は“Select Destination”ユースケースに依存している。このようなとき、“Floor”、“Elevator”、“ArrivalSensor”、“ElevatorButton”、“Motor”、“Door”は“Select Destination”の「概念を分解」したものであると表現する。

ところが、小谷の研究においては“Select Destination”と図 2.7 の 6 個のクラスの間に関係を自動生成することができない。

上記の例のように、概念の分解はユースケースと静的モデリングなど上位のフェーズに生じやすく、基本依存関係を抽出できないことが多い。[1]

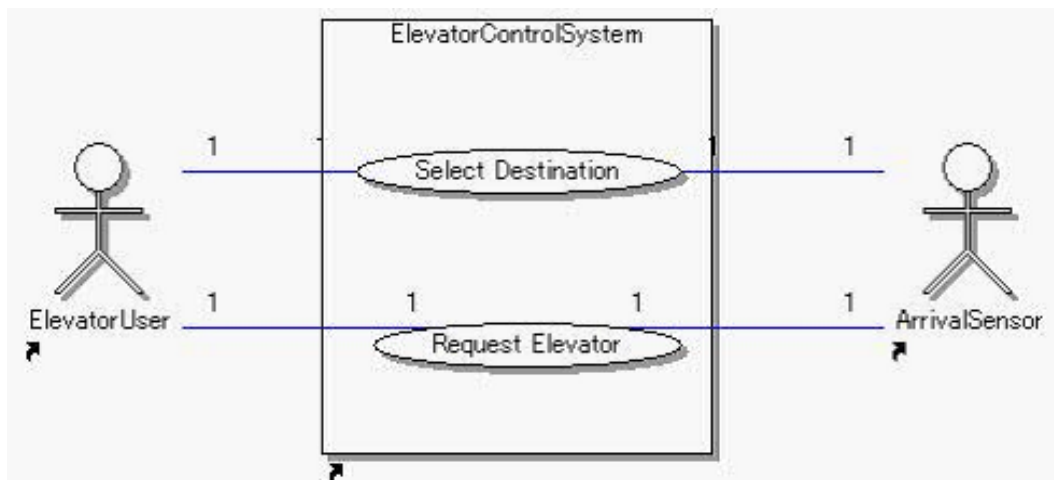


図 2.6: ユースケース図

²概念とは設計者が UML 図式要素に与えた振舞や機能のことを示す。

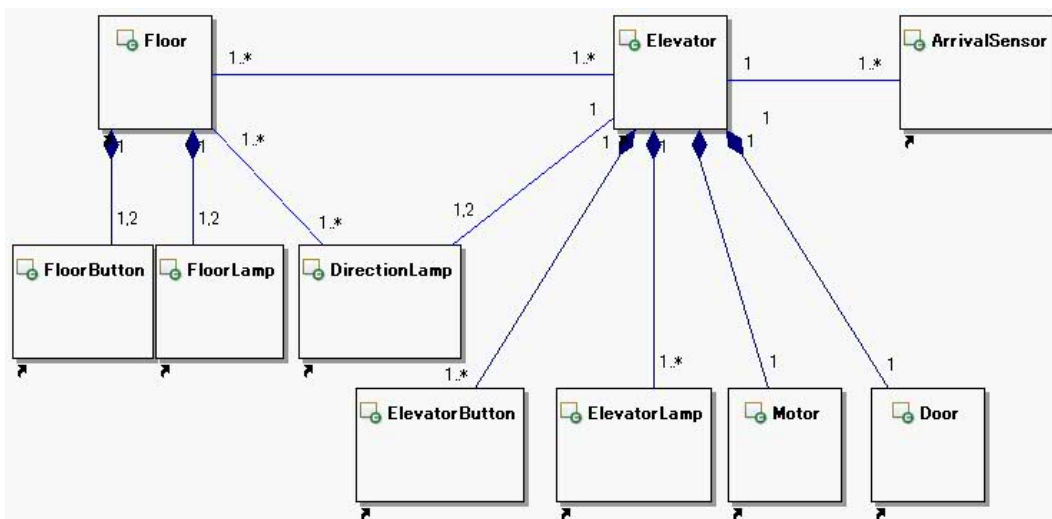


図 2.7: クラス図

第3章 依存関係生成モデルの改良

3.1 UML2.0 への対応

UML2.0 版に対応するにあたり、依存関係生成モデルの照合規則及び付加規則を再定義する必要がある。照合規則では、ターゲットとソース間の図の包含関係や名前の類似性を比較するので、UML2.0 版で新規に追加された図面や更新された図面に関して定義する必要がある。また付加規則に関しては、基本依存関係の両端にくる生成モデル要素を UML2.0 版で変更されたり、追加された UML 記述に対して再定義する必要がある。

3.1.1 UML2.0 の分析

UML2.0 版では 13 種類の図面が定義され、これらは表 3.1 に示すように 3 種類に分類できる。

表 3.1: 図の分類

図の分類	図面
関係図	ユースケース図、クラス図、オブジェクト図、コンポーネント図 配置図、パッケージ図、コンポジット構造図
振舞い図	ステートマシン図、アクティビティ図
相互作用図	シーケンス図、コミュニケーション図、相互作用概要図、タイミング図

次に、UML2.0 版で新規に加えられた図面や、UML1.5 版から大幅に変更の加えられた図に関して特徴を以下に示す。

- コンポジット構造図：UML2.0 版で新規に追加された図である。クラスの内部をパートを用いて表現できる。

- コンポーネント図：UML1.5 版以前では実装レベルの粒度の細かいソフトウェアコンポーネントだけではなく、設計や分析の段階でも利用されるよう改良された。さらには、コンポーネントの振る舞いを実現する内部要素を記述できるようになった。
- ステートマシン図：UML1.5 版ではステートチャート図と呼ばれていたものが、名前を変えた。
- シーケンス図：オブジェクトやクラス、コンポーネント間がメッセージをやり取りしながら実現する単位として相互作用が定義された。ソースコードレベルのきめ細かな内容 (例えば、分岐や繰り返しなど) をわかりやすく表現できるようになった。階層化された相互作用を表現できるようになった。
- コミュニケーション図：UML1.5 版ではコラボレーション図と呼ばれていたものが、名前を変えた。オブジェクトやクラス、コンポーネント間がメッセージをやり取りしながら実現する単位として相互作用が定義された。
- 相互作用概要図：UML2.0 版で新規に追加された図である。相互作用図のひとつであり、アクティビティ図の機能を利用して相互作用の順番を定義することができる。
- タイミング図：UML2.0 版で新規に追加された図である。相互作用図のひとつであり、時間軸に沿って、オブジェクト、クラス、コンポーネント等の状態の変化およびその間の関係を表現できる。細かい時間間隔での状態遷移や時間制約などを表す必要がある組込みシステム開発向けの図である。

3.1.2 照合規則

3.1.1 節の分析から、UML2.0 版で新たに定義した照合規則を表 3.2 に示す。UML1.5 版と変わっていないものに関しては、表 3.2 では省略した。UML2.0 版の照合規則の完全な表は付録 A 参照。以下で、表 3.2 の各照合条件を導出した理由を説明する。導出の際には文献 [5] を参照した。

1. ソースのステートマシン図はUML1.5 版では、ステートチャート図と呼ばれていた図が名前を改められたので、ステートチャート図のとき同様に照合条件を”Contained”とする。

2. ソースのコミュニケーション図はUML1.5版では、コラボレーション図と呼ばれていた図が名前を改められたので、コラボレーション図のとき同様に照合条件を”Contained”とする。
3. ソースの相互作用概要図は相互作用図に分類され、アクティビティ図の機能を利用を持っている。そこでターゲットをユースケース、ソースをアクティビティ図と考え、照合条件を”Contained”とする。
4. ソースのステートマシン図はUML1.5版では、ステートチャート図と呼ばれていた図が名前を改められたので、ステートチャート図のとき同様に照合条件を”Contained”とする。
5. ターゲットをクラス、ソースをコンポジット構造図としたときに二つの名前が似ているとき、コンポジット構造図がクラスの内部構造を表しているとして、照合条件を”Similar”とする。
6. ターゲットをオブジェクト、ソースをステートマシン図としたときステートマシン図がオブジェクトの状態遷移図を表しているとして、照合条件を”TypeSim”とする。
7. ターゲットをオブジェクト、ソースをタイミング図としたときタイミング図がオブジェクトの状態の時系列変化を表しているとして、照合条件を”TypeSim”とする。
8. コンポーネントは内部に内部要素を持つことができる。これは図の包含関係に該当するので照合条件を”Include”とする。
9. ノードは内部に成果物を持つことができる。これは図の包含関係に該当するので照合条件を”Include”とする。
10. ターゲットを成果物、ソースを成果物としたときに二つの名前が似ているとき、照合条件を”Similar”とする。
11. ステートマシン図は内部に状態を持つことができる。これは図の包含関係に該当するので照合条件を”Include”とする。
12. ステートマシン図は内部にコンポジット状態を持つことができる。これは図の包含関係に該当するので照合条件を”Include”とする。

13. コンポジット状態は内部に状態を持つことができる。これは図の包含関係に該当するので照合条件を”Include”とする。
14. コミュニケーション図は内部に相互作用を持つことができる。これは図の包含関係に該当するので照合条件を”Include”とする。
15. 相互作用は内部に相互作用を持つことができる。これは図の包含関係に該当するので照合条件を”Include”とする。また、ターゲットとソースの間に包含関係がなくとも相互作用同士の名前が似ているとき、照合条件を”Similar”とする。
16. 相互作用は内部にオブジェクトを持つことができる。これは図の包含関係に該当するので照合条件を”Include”とする。
17. シーケンス図は内部に相互作用を持つことができる。これは図の包含関係に該当するので照合条件を”Include”とする。
18. 相互作用概要図は内部に相互作用を持つことができる。これは図の包含関係に該当するので照合条件を”Include”とする。
19. パッケージ図は内部にパッケージを持つことができる。これは図の包含関係に該当するので照合条件を”Include”とする。
20. コンポジット構造図は内部にパートを持つことができる。これは図の包含関係に該当するので照合条件を”Include”とする。
21. ターゲットをパート、ソースをパートとしたときに二つの名前が似ているとき、照合条件を”Similar”とする。
22. タイミング図は内部に状態を持つことができる。これは図の包含関係に該当するので照合条件を”Include”とする。

3.1.3 付加規則

UML2.0 版で変更されたり、追加された UML 記述に対して生成モデル要素に分類する必要がある。分類した結果を表 3.3 に示す。表 3.3 中で太字で表されている生成モデル要素が、UML2.0 版において変更されたり、追加された UML 記述である。以下で、表 3.3 に分類した理由を説明する。

表 3.2: UML2.0 版に追加した照合規則

番号	ターゲット	ソース	照合条件
1	ユースケース	ステートマシン図	Contained
2	ユースケース	コミュニケーション図	Contained
3	ユースケース	相互作用概要図	Contained
4	クラス	ステートマシン図	Contained
5	クラス	コンポジット構造図	Similar
6	オブジェクト	ステートマシン図	TypeSim
7	オブジェクト	タイミング図	TypeSim
8	コンポーネント	内部要素	Include
9	ノード	成果物	Include
10	成果物	成果物	Similar
11	ステートマシン図	状態	Include
12	ステートマシン図	コンポジット状態	Include
13	コンポジット状態	状態	Include
14	コミュニケーション図	相互作用	Include
15	相互作用	相互作用	Similar,Include
16	相互作用	オブジェクト	Include
17	シーケンス図	相互作用	Include
18	相互作用概要図	相互作用	Include
19	パッケージ図	パッケージ	Include
20	コンポジット構造図	パート	Include
21	パート	パート	Similar
22	タイミング図	状態	Include

- 内部要素はコンポーネントの内部のクラスを表現したものであるから、Classifier 要素に分類した。
- 成果物はノードの内部構造を表現したものであるから、Classifier 要素に分類した。
- パートはクラスやコンポーネントの内部を表現する要素でこれ自身がクラスやコンポーネントであるから、Classifier 要素に分類した。
- コネクタはパート間の関係を表すものであるから、関係要素に分類した。
- コンポジット状態は、内部に別の状態遷移領域を持ちそれ自身も状態として機能するため状態要素に分類した。

- 相互作用はオブジェクトがメッセージをやり取りしながら処理をする単位であると定義されていて、この単位自体が一つのインスタンスであるとみなしインスタンス要素とした。
- 関係図、振舞図、相互作用図に関しては表 3.1 より

表 3.3: UML2.0 版対応の生成モデル要素

生成モデル要素	UML 図または図式要素
Classifier 要素	アクター、ユースケース、クラス、パッケージ、 ノード、コンポーネント、オブジェクト (オブジェクト図) 内部要素、成果物、パート
関係要素	関連、依存、集約、汎化、リンク、コネクタ
状態要素	状態、アクション状態、コンポジット状態
遷移要素	遷移、イベント、アクション
インスタンス要素	オブジェクト (コラボレーション図、シーケンス図) 相互作用
メッセージ要素	メッセージ
関係図	ユースケース図、クラス図、オブジェクト図、 コンポーネント図、配置図、コンポジット構造図、パッケージ図
振舞図	ステートマシン図、アクティビティ図
相互作用図	シーケンス図、コミュニケーション図、相互作用概要図、 タイミング図

3.2 概念の分解への対応

下位のフェーズで生じる概念の分解に関しては、小谷 [1] による評価実験でも基本依存

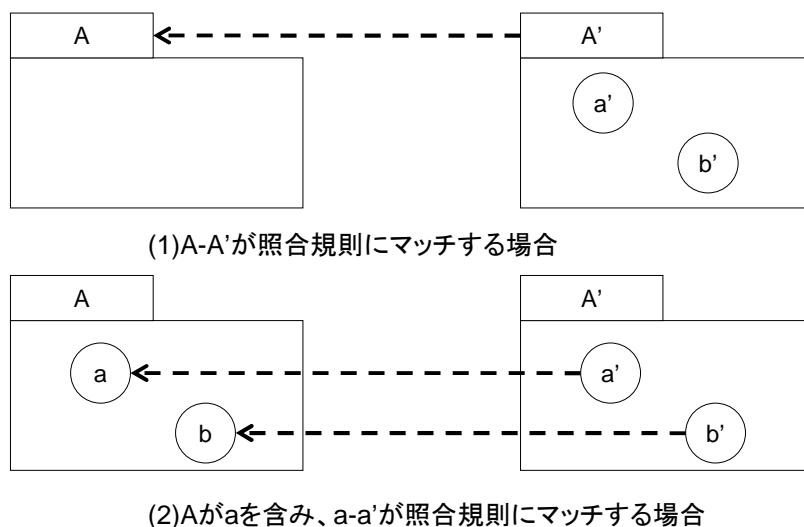


図 3.1: 概念の分解できる例

これに対し、上位のフェーズで生じる概念の分解に関しては、2.5.2節の例のようにUML記述間に名前の類似性が見られないことが多い。

そこで本研究では、ユースケースの概念の分解においては、UML記述間に設計者が意図する依存関係を付加するという方法を採用する。以下に、この立場の妥当性を説明する。

上位フェーズでの依存関係の見落としは、そこから波及し下位フェーズの依存関係の見落としにつながり結果的に非常に多くの変更波及対象を見落とす可能性を含んでいる。図3.2を例にとって説明する。フェーズはフェーズ1から順に設計されたものとする。設計者は各フェーズにおいて図のようなUML記述を設計した。また、設計者の意図としてはaとbはAに依存し、tとyはaに依存し、wはbに依存している。ここで2.5.2節の例を当てはめて考える。Aを”Select Destination”、aを”Motor”、bを”Door”とする。このとき、”Motor”と”Door”は”Select Destination”の依存関係を自動生成できないので、その下のフェーズのUML記述t、v、yとの依存関係まで抽出できなくなってしまい、結果

的に二つの変更対象を見逃すことで、そこから波及する多くの変更対象を見落としてしまうことにつながる。

そこで、“Select Destination”と”Motor”、“Select Destination”と”Door”の間に設計者に依存関係を付加させる。すると、下位のフェーズで生じる概念の分解に関してはUML記述間の名前が似ていることが多く、従来の方法で変更波及解析できる。

それでは、設計している際に生じる全ての依存関係に関して、設計者の手で依存関係を

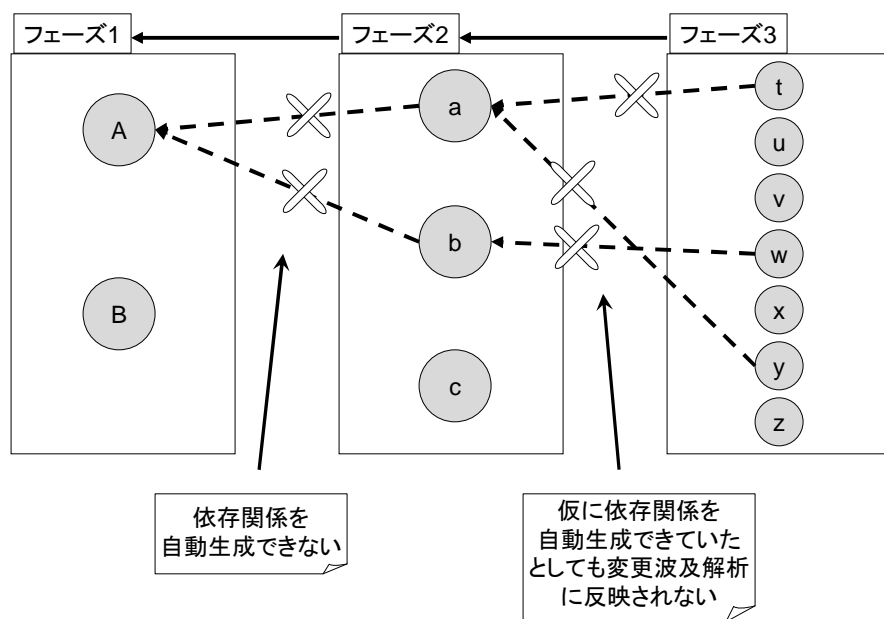


図 3.2: 概念の分解の解析ミス

第4章 実装

と

じ

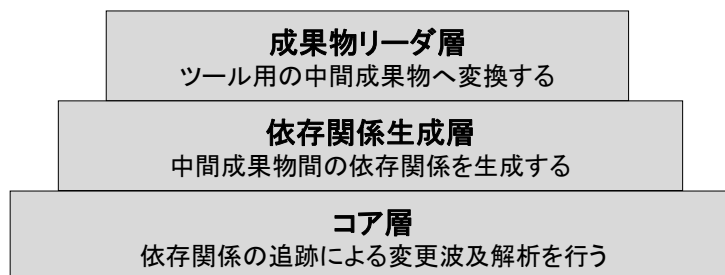


図 4.1: フレームワーク (再掲)

4.1 依存関係生成層

依存関係生成層は本ツール用の中間成果物間の基本依存関係を生成する層であった。UML2.0 版に拡張するにあたり、3.1 節より基本依存関係を自動生成する依存関係生成モデルが拡張された。これにともない依存関係生成モデルに対応するプラグインである UML 図の基本依存関係生成ツールプラグインを拡張した。

4.2 成果物リーダ層

小谷 [1] においては、UML モデリングツール Pattern Weaver[7] によって描かれた、UML1.5 版の UML 図面要素群を中間成果物として変更波及解析ツールに読み込ませていた。本研究では、UML2.0 版で基本依存関係の自動生成を行うため、UML2.0 版対応の UML モデリングツール Borland®Together®[8] によって描かれた、UML 図面要素群を中間成果物として変更波及解析ツールが読み込めるように拡張した。これには、成果物リーダ層に対応する Together ファイルリーダプラグインを実装した。

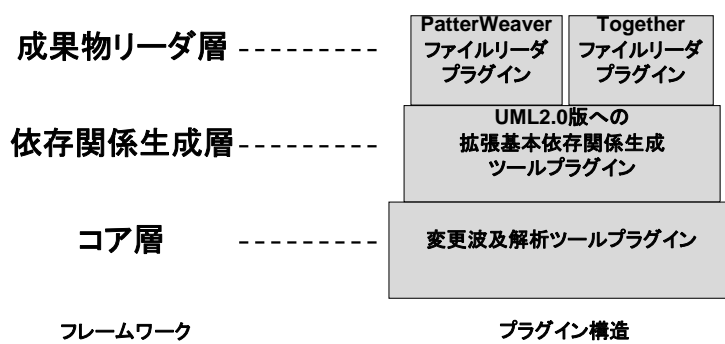


図 4.2: プラグイン構造

第5章 評価

本章では、概念の分解に対する本研究のアプローチが変更波及解析に対してどの程度有効であったか、また、小谷の成果 [1] に比べてどの程度改善されたかを評価する。今回、評価実験の対象として選んだのは小谷が題材としたエレベータ制御システム [6] である。

5.1 実験対象の特徴

エレベータ制御システムの特徴を以下に述べる。

1. ドメイン センサー・アクチュエータベースのリアルタイム制御システム
2. アーキテクチャ 集中制御型
3. フェーズ構成の特徴 ユースケースによる機能要求定義、問題領域オブジェクトの発見と動的モデリング、サブシステム設計、タスク設計
4. 各フェーズで利用している図面の種類と数 ユースケースによる機能要求定義(ユースケース図 2 枚)、問題領域オブジェクトの発見と動的モデリング(クラス図 2 枚、状態マシン図 5 枚、コミュニケーション図 5 枚)、サブシステム設計(クラス図 1 枚、コミュニケーション図 3 枚)、タスク設計(クラス図 1 枚、コミュニケーション図 8 枚)
5. UML 図面の数と UML 図式要素の数 UML 図面が 27 枚、UML 図式要素が 260 個

5.2 評価

ユースケースを変更したときに、そこから変更が波及する UML 記述 (UML 図及び UML 図式要素) をどれだけ抽出できるかを評価する。評価にあたり、エレベータ制御システムの分析・設計例 (UML 図 27 枚) をもとに三つのデータを作成した。

1. 文献 [6] の著者 (設計者) が認識したであろう UML 記述間の依存関係をもとにユースケースが変更されたとき、どれだけの UML 記述が変更されたかを観測する。
2. 小谷の研究による依存関係生成モデルを利用し、UML 記述間の依存関係を自動生成し、ユースケースが変更されたとき、どれだけの UML 記述が変更されたかを観測する。
3. 本研究で提案したユースケースとクラスの間設計者が依存関係を付加し、残りを小谷の研究による依存関係生成モデルを利用し、UML 記述間の依存関係を自動生成し、ユースケースが変更されたとき、どれだけの UML 記述が変更されたかを観測する。

1. のデータが設計者が意図した依存関係を反映した変更波及解析結果であるから、このデータが正解であるとして、2. と 3. の結果を 1. の結果と比較する。比較結果を表 5.1 に示す。各行の上段が 3. の本研究による解析結果で、下段の括弧内の数値が小谷の研究による解析結果である。

3. のデータ作成にあたり、設計者に依存関係を付加させたのは以下の 8 点である。

- ユースケース”Select Destination”とクラス”Motor”
- ユースケース”Select Destination”とクラス”Door”
- ユースケース”Select Destination”とクラス”ArrivalSensor”
- ユースケース”Select Destination”とクラス”ElevatorButton”
- ユースケース”Request Elevator”とクラス”Motor”
- ユースケース”Request Elevator”とクラス”Door”
- ユースケース”Request Elevator”とクラス”ArrivalSensor”
- ユースケース”Request Elevator”とクラス”FloorButton”

表 5.1 において、再現率 (Recall) と適合率 (Precision) は、 $\#X$ を事象 X の個数として以下の式で示す。

$$Recall(\%) = \frac{\#(A \cap B)}{\#A}(\%)$$

$$Precision(\%) = \frac{\#(A \cap B)}{\#B}(\%)$$

表 5.1: 評価結果

変更されたユースケース	正解	自動生成	共通	検索漏れ	ノイズ	再現率	適合率
Select Destination	163	159 (143)	153 (110)	10 (53)	6 (33)	94% (67%)	96% (77%)
Request Elevator	166	167 (149)	157 (117)	9 (39)	10 (32)	95% (70%)	94% (79%)

5.3 考察

再現率、適合率の値から本研究での概念の分割へのアプローチが有効であるといえる。上位フェーズに少数の依存関係を設計者の手で加えるという操作で、加えた依存関係以上に、下位フェーズへの依存関係の波及も逃さず抽出できる結果を得られた。

第6章 結論

この章では、本研究をまとめ、さらに今後の課題について述べる。

6.1 本研究のまとめ

本研究では、UML 図面間の変更波及解析の対象を UML1.5 版から UML2.0 版に拡張した。小谷が提案した基本依存関係は UML2.0 版でも利用可能であり、基本依存関係を自動生成する依存関係生成モデルに改良を加えることで、UML2.0 版における基本依存関係を自動生成法を提案した。

また、小谷の研究において変更波及解析不能であった概念の分解に関して、変更波及解析をできるように基本依存関係の自動生成法を一部改良した。基本依存関係を自動生成できない部分に関して、設計者に基本依存関係を付加させることにより、基本依存関係を自動生成できていながらも変更波及解析できていなかった部分の変更波及解析を可能にした。これをエレベータ制御システムの事例研究を題材として、変更波及解析を行った結果、小谷による変更波及の対象の抽出結果が 70% 程度であったのに対し、本研究では、94% 以上に改善された。

6.2 今後の課題

本研究での概念の分解へのアプローチは、開発方法論依存である。これを他の開発方法論でも適用できるようにするのが、今後の課題である。

参考文献

- [1] 小谷正行, 落水浩一郎: “UML 記述の変更波及解析に利用可能な依存関係の自動生成法”, 情報処理学会論文誌 Vol.49 No.7 pp.2265-2291, 2008.7
- [2] 菅井拓海, “クラス間の参照関係の強さと名前を利用した Java ソースコード中の強調クラス群の抽出”, 北陸先端科学技術大学院大学, 修士論文, 2007.3
- [3] Object Management Group, UML Superstructure Specification, Version 2.1.2, <http://www.omg.org/spec/UML/2.1.2/Superstructure/PDF/>
- [4] Object Management Group, UML Infrastructure Specification, Version 2.1.2, <http://www.omg.org/spec/UML/2.1.2/Infrastructure/PDF/>
- [5] Object Management Group 著, 西原裕善監訳, UML2.0 仕様書 UML2.1 対応, オーム社, 2006.
- [6] Hassan Gomma, Designing concurrent, distributed, and real-time applications with UML, Addison Wesley, Inc., 2000.
- [7] Pattern Weaver, <http://pw.tech-arts.co.jp>
- [8] Borland®Together®, <http://www.borland.com/jp/products/together/index.html>
- [9] Eclipse Project, <http://www.eclipse.org/eclipse/>
- [10] Eclipse Graphical Editing Framework (GEF), <http://www.eclipse.org/gef/>

付録A UML2.0版対応の照合規則

表は3.1.2節で定義したUML2.0版対応の照合規則の全部である。表中で赤太字になっているものが、UML2.0版になって変更もしくは新たに定義されたUML記述である。

表 A.1: UML2.0 版対応のすべての照合規則

ターゲット	ソース	照合条件
ユースケース図	アクター	Include
ユースケース図	ユースケース	Include
ユースケース	クラス図	Contained
ユースケース	ステートマシン図	Contained
ユースケース	アクティビティ図	Contained
ユースケース	コミュニケーション図	Contained
ユースケース	シーケンス図	Contained
ユースケース	相互作用概要図	Contained
ユースケース	ユースケース	Similar
アクター	アクター	Similar
アクター	クラス	Similar
アクター	オブジェクト	TypeSim
クラス図	クラス	Include
クラス図	パッケージ	Include
パッケージ	クラス	Include
パッケージ	パッケージ	Similar
クラス	パッケージ	Similar
クラス	オブジェクト	SimType, Contained
クラス	ステートマシン図	Contained
クラス	アクティビティ図	Contained
クラス	コンポジット構造図	Similar
クラス	クラス	Similar, Include
オブジェクト図	オブジェクト	Include
オブジェクト	クラス	TypeSim
オブジェクト	ステートマシン図	TypeSim

表 A.1: UML2.0 版対応のすべての照合規則

ターゲット	ソース	照合条件
オブジェクト	アクティビティ図	TypeSim
オブジェクト	タイミング図	TypeSim
オブジェクト	オブジェクト	Similar,Include
コンポーネント図	コンポーネント	Include
コンポーネント	コンポーネント	Similar
コンポーネント	内部要素	Include
配置図	ノード	Include
ノード	ノード	Similar
ノード	成果物	Include
成果物	成果物	Similar
ステートマシン図	状態	Include
ステートマシン図	コンポジット状態	Include
コンポジット状態	状態	Include
状態	状態	Similar,Include
アクティビティ図	アクション状態	Include
アクション状態	アクション状態	Similar,Include
コラボレーション図	オブジェクト	Include
シーケンス図	オブジェクト	Include
コミュニケーション図	相互作用	Include
相互作用	相互作用	Similar,Include
相互作用	オブジェクト	Include
シーケンス図	相互作用	Include
相互作用概要図	相互作用	Include
パッケージ図	パッケージ	Include
コンポジット構造図	パート	Include
パート	パート	Similar
タイミング図	状態	Include