

Title	UML2.4を対象とするUML図・UMLモデリング要素間の依存関係の自動生成法
Author(s)	王, 亜莉
Citation	
Issue Date	2012-03
Type	Thesis or Dissertation
Text version	author
URL	http://hdl.handle.net/10119/10403
Rights	
Description	Supervisor:落水 浩一郎 教授, 情報科学研究科, 修士

修 士 論 文

UML2.4を対象とするUML図・UMLモデリン
グ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科学専攻

王 亜莉

2012年3月

修 士 論 文

UML2.4を対象とするUML図・UMLモデリング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 青木 利晃 准教授

北陸先端科学技術大学院大学
情報科学研究科情報科学専攻

1010076 王 亜莉

提出年月: 2012年2月

概要

本論文では、UML1.5版の依存関係を自動生成するモデルを、UML最新版であるUML2.4版へ対応するように改良した。小谷による提案した依存関係生成モデルはUML1.5を対象とした。この自動生成モデルの基本となるのは、照合規則、付加規則、選択規則という3つの規則である。UML2.0以降は、モデリングの考え方に、大きな違いが出てきて、主に内部構造という概念が導入された。本論文では、UML2.4の追加点・変更点を分析しながら、内部構造に対応するために、主に照合規則と付加規則を拡張することにより、依存関係の自動生成モデルを改良した。

目次

第1章	はじめに	1
1.1	背景	1
1.2	目的	1
1.3	本論文の構成	3
第2章	UML2.4の追加点・変更点	4
第3章	既存の依存生成法	13
3.1	既存の自動生成法	13
3.2	既存の自動生成法の問題点・改善策	14
第4章	基本依存関係	16
4.1	UML2.4における依存関係のセマンティクス	16
4.2	UML2.4に対応の基本依存関係	18
第5章	付加規則	20
5.1	生成モデル要素の再分類	20
5.1.1	分類方法	20
5.1.2	UML2.4対応の生成モデル要素	23
5.2	付加規則	23
第6章	照合規則	27
6.1	分類方法	27
6.2	照合規則	30
第7章	選択規則	36
第8章	終わりに	38
8.1	まとめ	38
8.2	今後の課題	38

第1章 はじめに

1.1 背景

ソフトウェア開発においては、数多くの中間成果物（開発時文書）が作成され、その内容は複雑に関係している。ある特定の中間生成物は、すでに作成された文書内容を参照して作成される。この時、参照先の文書を変更した場合には参照元の文書を変更する必要がある。このような関係を依存関係と呼ぶ。この意味で、依存関係は変更にもなう、影響範囲の特定に直接役立つ情報である。しかし、何百枚もの文書があり、それぞれの文書が数十個の要素を含み、それぞれの要素間に依存関係がある場合には、これを人手によって記録し利用するのは困難を極める。また、文書の変更にもなう、依存関係自体も新たに設定しなおす必要がある、それらを効率よくかつ矛盾を生じないように管理することは困難である。これに対する一つの解は依存関係の自動生成である。

すでに、小谷が UML1.5 版に対して自動生成の理論と技術を提案しているが、現在、UMLは大改訂が行われ、UML2.4に移行している。本研究では、最新版となる UML2.4 に対応して依存関係の自動生成法を拡張する。

1.2 目的

小谷により提案された自動生成モデルは、実際のソフトウェア開発で作られた UML モデル（中間成果物）に対して、より抽象的なレベル（メタモデル）で、その依存関係を分析し、抽出するのである。本論文はこの自動生成モデルを、UML2.4 版に適用されるように、関係する用語やモデル要素などを明確に再定義・分類しながら拡張・改良していく。まず、関係する用語を説明してから、自動生成法の概要を紹介する。前文で述べた依存関係は、図 1.1 のように表現される。ソースはターゲットに依存していると読むソースをもとにターゲットが作成されているという前提があって、ターゲットが変更された場合には、ソースを見直す必要がある。

ターゲット <----- ソース

図 1.1: 依存関係の表記

- **プロセス情報**

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 図面群を定義した情報である。

- **UML 記述**

UML 図（例えば、クラス図やオブジェクト図など）や UML モデリング要素（例えば、クラスやオブジェクトなど）である。

- **基本依存関係**

UML1.5 版のメタモデルに定義された種々の依存関係を変更の影響内容の観点から、自動生成できるように再整理し、生存従属、情報共有、コピー、同一概念として定義したものである。実際に自動生成モデルにより生成されたのは、基本依存関係である。

- **生成モデル要素**

UML 記述をより抽象的な概念でグループとしてまとめたものを生成モデル要素と呼ぶ。

- **照合規則**

UML 記述の名前や型名には類似性がある。この類似性から UML 記述間の依存関係が推定される。照合規則はこの類似性を定義する規則である。具体的にいうと、UML 記述間の名前の対応付けルールである。

- **付加規則**

生成モデル要素間にあらかじめ定義された基本依存関係である。照合規則で候補となった UML 記述の組み合わせは、それぞれどの生成モデル要素に属するのかを決定し、生成モデル要素間に定義された基本依存関係を設定する。基本関係の両端にくる UML 記述の型は開発方法論に依存するので、採用する開発方法論ごとに異なる付加規則を定義する必要がある。本論文では Unified Process に基づく開発方法論に適用可能な付加規則を拡張する。

- **選択規則**

生成モデル要素間に、複数の基本依存関係が設定されている場合、プロセス情報を用いて、適切な基本依存関係を選択するための規則である。

次は、UP (Unified Process) 開発プロセスに基づいて、UML1.5 版が対象となった現存の自動生成法である。

依存関係の自動生成モデルは、図 1.2 で表しているように、照合規則、付加規則、選択規則で構成される。プロセス情報と UML 図面群を入力し、UML 図面群のための依存関係生成モデルに基づいて、UML 記述の依存関係の自動生成が期待される。

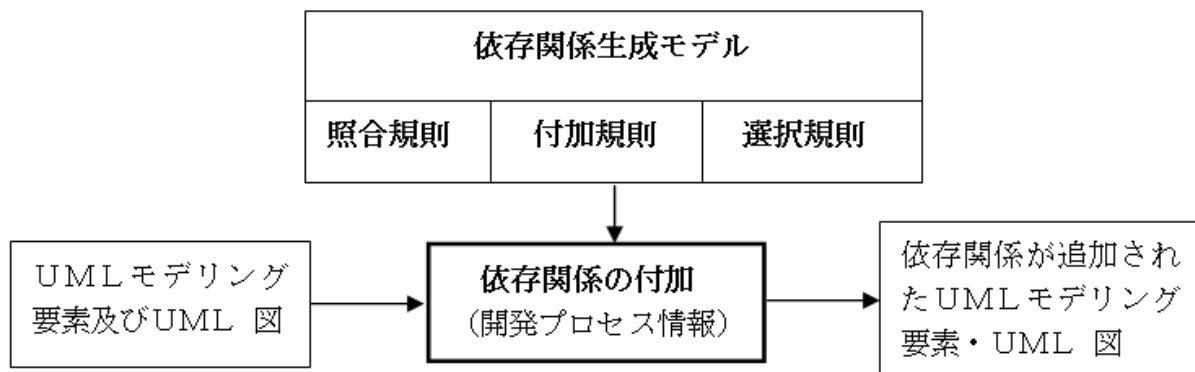


図 1.2: 依存関係の自動生成法の概要

1.3 本論文の構成

本論文は次のように構成される。第2章では、最新版となるUML2.4の変更点を説明しながらまとめる。第3章では、まず小谷が提案した依存関係の自動生成法の手順を紹介する。そして、提案の問題点・改善点を検討し、改善策・期待結果を説明する。第4章では、基本依存関係を検討する。第5章では、照合規則の分類方法を整理し、改善する。第6章では、生成モデルの再分類方法を説明し、最新版に拡張する。第7章では、選択規則を検討する。第8章では、本論文の成果をまとめ、さらに今後の課題を検討する。

第2章 UML2.4の追加点・変更点

UML2.0以降は、UML1.5までのUMLモデリング手法や用語の表現不足と曖昧さを解消したほか、UML1.xxに比べて、モデリングの考え方に大きな違いが出てきた。

例1：CBSDに対応するため、内部構造の概念が導入された。

例2：MDAに対応するため、シーケンス図に制御構成子が導入された。

例3：ビジネスモデリングの支援図式が導入された。

この結果、UML2.4では、たくさんのUML図と要素が追加・変更された。特に包含分類子のインスタンスの中に作られる相互接続された要素の構造を明示する機構が新規追加された（「内部構造」と呼ばれる）。本研究では、主に内部構造に対応するように、依存関係の自動生成法を改良する。本章では、UML2.4版の追加点・変更点を洗い出して分析する。まず、新規追加・変更されたモデリング要素を分析する。

● 新規追加されたモデリング要素

－ 協調

一種の分類子であり、協調している要素（役割）の構造を記述する。それぞれの要素は特化した機能を実行し、これらが集まってある目的とする機能性を達成する。その主要な目的はシステムがどのように稼動するかを説明することである。どのような種類の振舞いの記述も関連させられる。役割やコネクタに頼る内部構造として規定される。2のように、協調の名前を含んでいる破線の楕円アイコンで表示される。

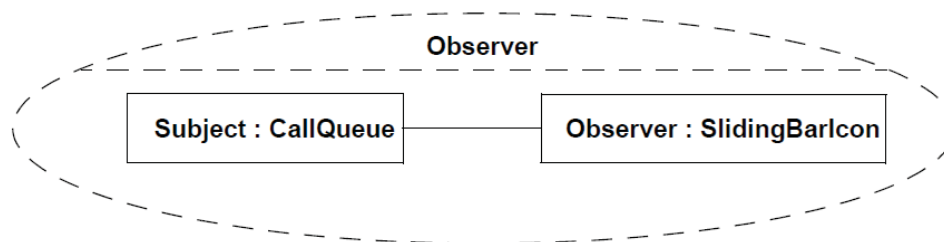


図 2.1: 協調アイコンの中に示される Observer 協調の内部構造（コネクションは Subject と Observer の役割の間に表されている）

代わりに、合成構造図で表示してもよい。Observer 協調の2つの役割 Subject と Observer は、目的とする振舞いをするための協調にする。Subject の役割を

果たすどのようなインスタンスも CallQueue によって規定されたプロパティを所有しなければならない。Observer の役割も同様である

– 協調使用

協調の役割を果たす特定のクラスまたはインスタンスを含んでいる特定の状況への協調によって記述されるパターンの適用を表しているとして定義されている。分類子のプロパティの関係を説明するために、協調の特定の使用を表すので、協調のインスタンスといえる。

– コネクタ

2つ以上のインスタンスの間でコミュニケーションを可能にするリンクを規定する。このリンクは、関連のインスタンスでもよいし、あるいはコミュニケーションできるインスタンスの可能性を表してもよい。関連する分類子の任意のインスタンス間のリンクとは違い、コネクタは接続された部品の実インスタンス間のリンクを規定している。また、UML1.4の関連の概念はコネクタによって包含された。

– 相互作用

内包する分類子の振舞いの単位である。協調だけではなく、分類子の中にも含まれるようになった。シーケンス図、相互作用概要図、コミュニケーション図といった複数の異なった図で表現することができる。2は、シーケンス図での相互作用の例である。User と ACSystem という型の、2つの生存線の間で通信する

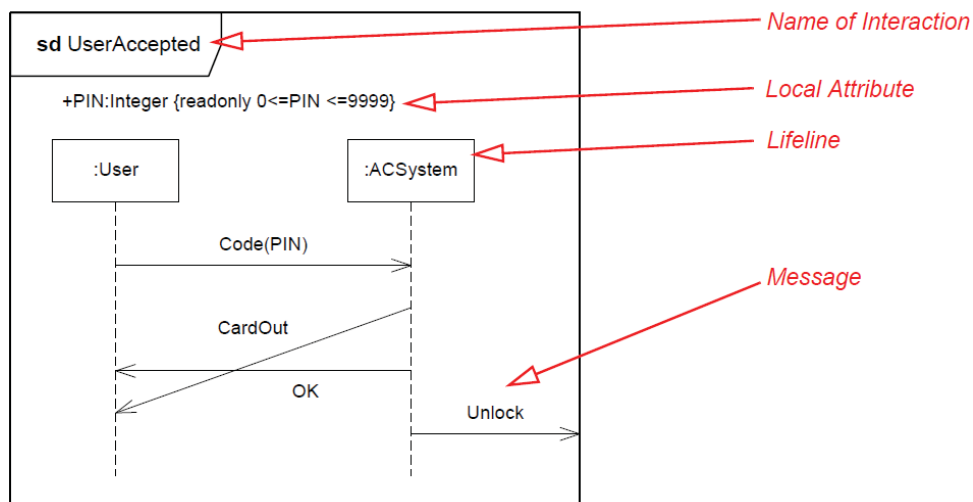


図 2.2: シーケンス図での相互作用の例

– 相互作用使用

相互作用使用は、相互作用を参照する。相互作用使用がある、参照された相互作用の内容をコピーすることを簡略化した表現方法である。この概念は UML1.5

にはなかった。相互作用使用は、複数の相互作用が、それらの仕様の共通の部分を表す1つの相互作用を参照することができるようになった。2の中では、入力変数「Illegal PIN」をもつ EstablishAccess という相互作用を参照している相互作用が見て取れる。オプションの CombinedFragment 内では、OpenDoor を参照し、変数をもたない他の相互作用使用もある。

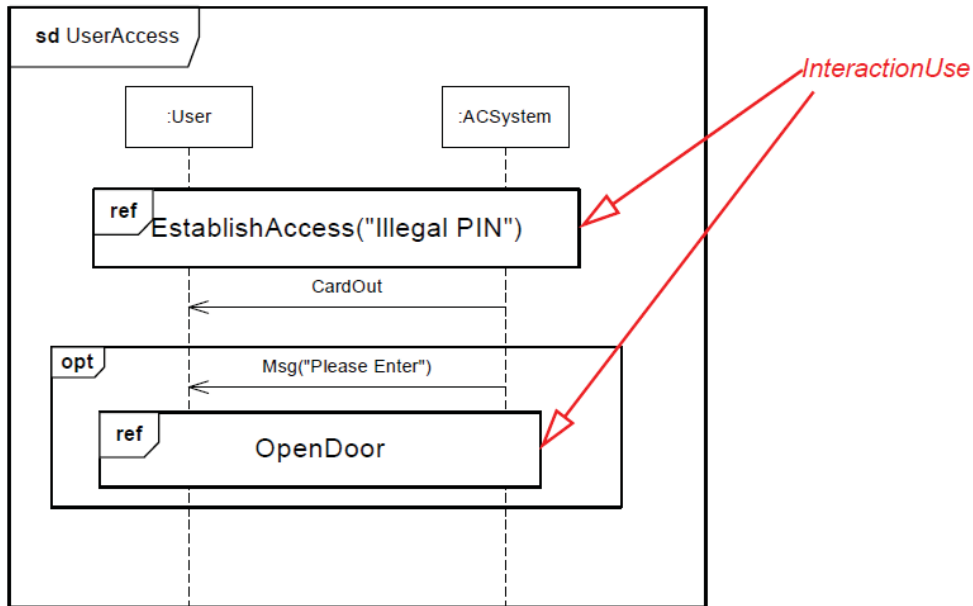


図 2.3: 相互作用使用

● 変更されたモデリング要素

– ユースケース

ユースケースとその主体間における関係が、明確にされている。また、ユースケースが、一般の分類子（ただのパッケージではない）により所有されることが可能である。

– メッセージ

相互作用の生存線間でのコミュニケーション（通信）を定義する。どちらかの端にゲートを待つことができるようになった。

– 状態機械

UML1.5から拡張された。1.5において、状態機械の汎化は、非標準的慣習として非形式的に説明されているだけである。そしてUML2.4では、次の状態種別が区別される：単純状態、合成状態、サブマシン状態。

– 単純状態

下位状態を持たない状態である。

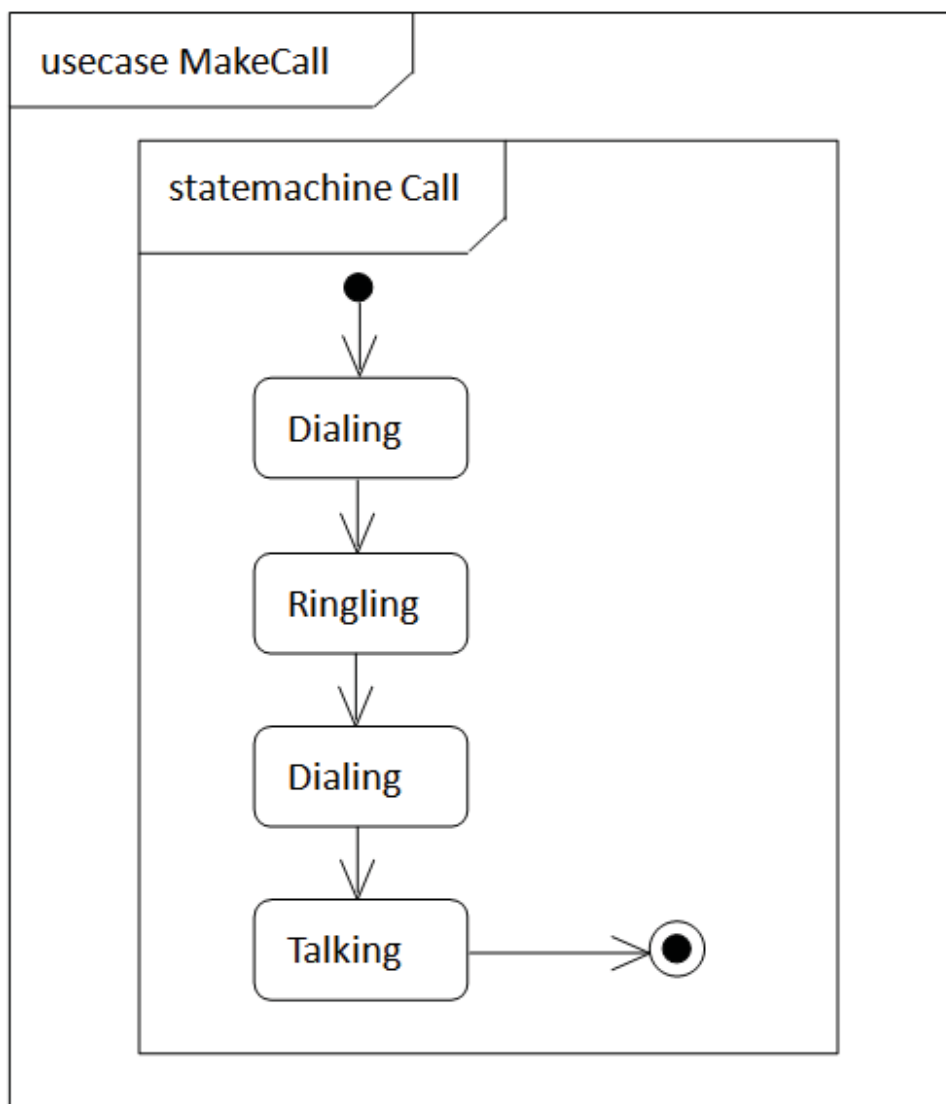


図 2.4: 状態機械図を内部構造として含むユースケース

- **合成状態**

下位に二つ以上の状態を包括する状態を表す。1つの領域を含むかまたは2つ以上の直交した領域に分解される。

- **サブマシン状態**

ある状態マシンを別の状態マシンの中の1つの状態として再利用するために定義されたものである。意味的に参照状態機械によって定義された合成状態と同等だが、通常の合成状態とは異なり、入場点および退場点を經由してこの合成状態に入場し退場する。

次は、UML2.4で定義された「構造図と振舞い図の分類」に基づいて、新規追加・変更された図を説明する。

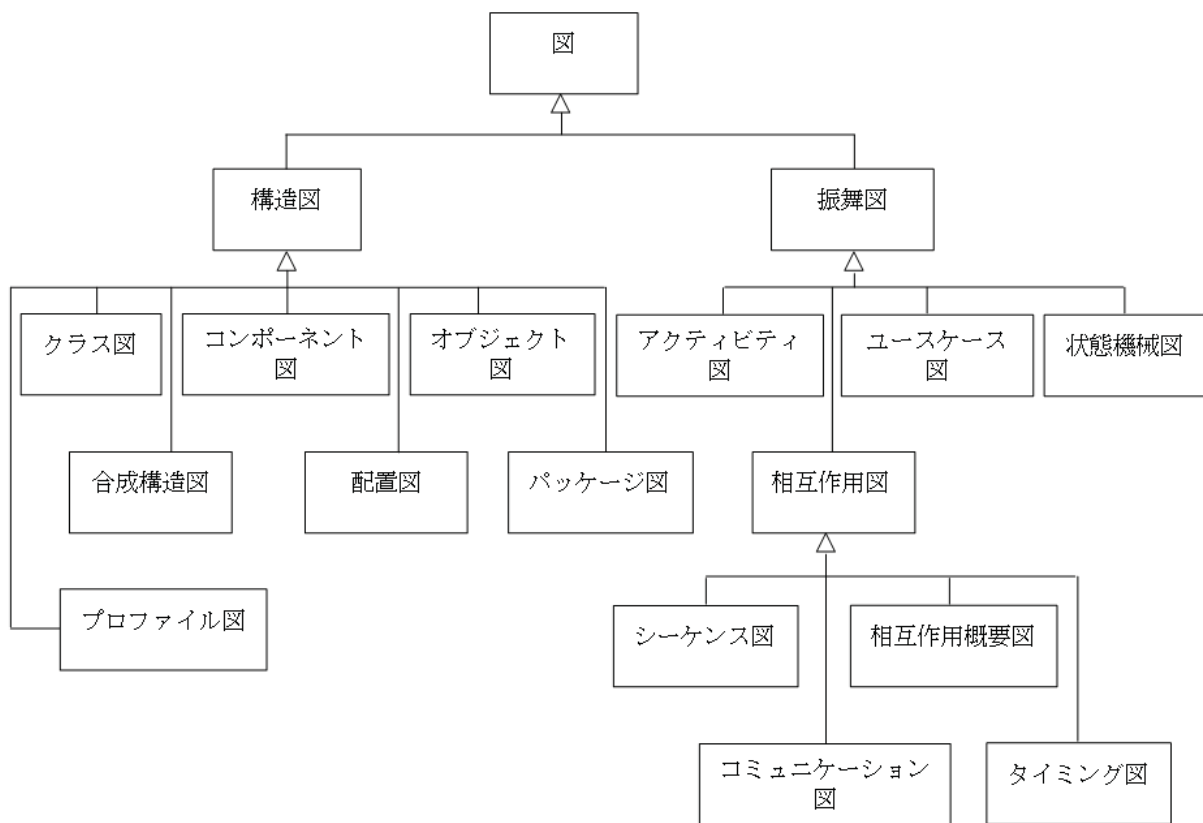


図 2.5: 構造図と振る舞い図の分類

- **新規追加された図**

- **プロフィール図**

プロフィール図では、カスタマイズされたステレオタイプ、タグ定義、UMLのダイアログ専用の制約などを定義することができる。カスタマイズはプロファ

イル内に定義され、パッケージに適用される。プロファイル図は `profile` のステレオタイプが定義されたパッケージ内の全ての図が該当する。UML プロファイルの機能を利用することで、UML の内容を独自に拡張することができる。拡張方法は、ステレオタイプとタグ付き値である。これらの情報定義し、要素・接続などに適用する。プロファイルを利用することにより、特定のドメインや目的に応じた、独自のモデルを作成するための要素や接続を定義して利用することができる。

– 相互作用概要図

機能ごとに記述された相互作用図（ユースケース図やシーケンス図など）が、より広域のシステム構成から見たとき、それぞれがどのように連携しているのかを表現する図である。具体的には、相互作用図をアクティビティ図の構成要素として使用する。つまり、相互作用を表すアクティビティ図の特化であるといえる。ただし、いくつかの点でアクティビティ図と異なる。図 2.6 では、相互作用 `EstablishAccess` 最初に発せし、その後に弱い順序付けで、インラインの相互作用に記述されている `CardOut` メッセージが続く。次に、分岐の 1 つに `InteractionConstraint` をもつ分岐ノードで表される選択肢がある。この制御フローに沿って、（弱い）順序付けのインラインの相互作用と相互作用使用がある。

– タイミング図

相互作用図の 1 つであり、クラスやオブジェクトの状態を時系列で表す。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場合に用いられる。タイミング図は個々の分類子と分類子の相互作用の両方の振舞いを記述し、生存線のモデルかされた条件に変化をもたらすイベントの発生の時間に焦点を当てる。タイミング図の表現方法は 3 通りあるが、図 2.7 はそのひとつでタイミング図を表す。

– コミュニケーション図

コラボレーション図が改名されたものである。

– 合成構造図

コミュニケーションリンクを通じて協調動作する互いに結び付いたインスタンスが、実行時にどうなっているかの例を調べるためのものである。協調使用における協調の使用と同様に、分類子の内部構造を描く。図 2.8 は `Observer` 強調を示す合成構造図である。

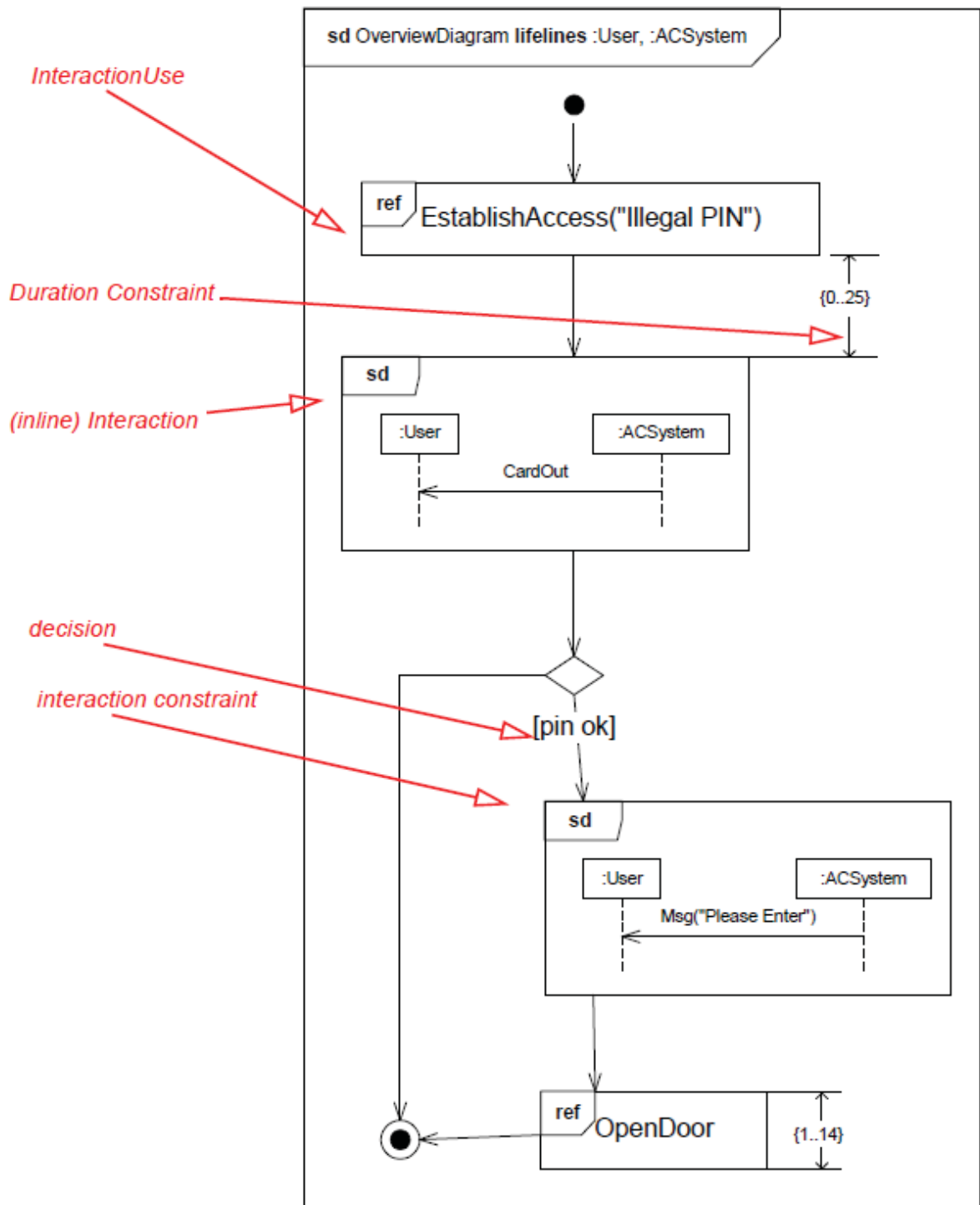


図 2.6: ハイレベルの相互作用図を表す相互作用概要図

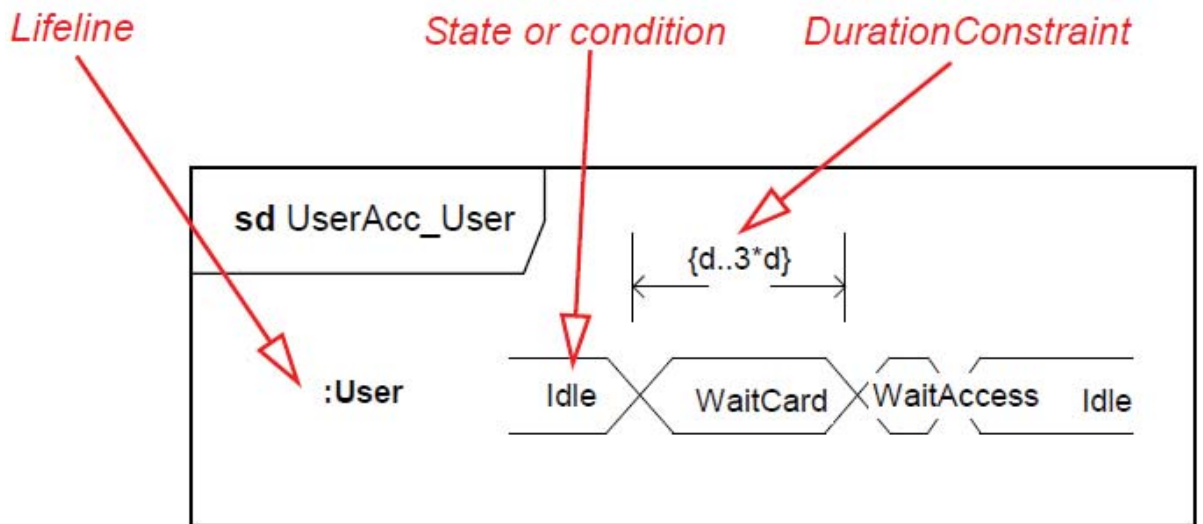


図 2.7: 状態を伴ったコンパクトな生存線のタイミング図

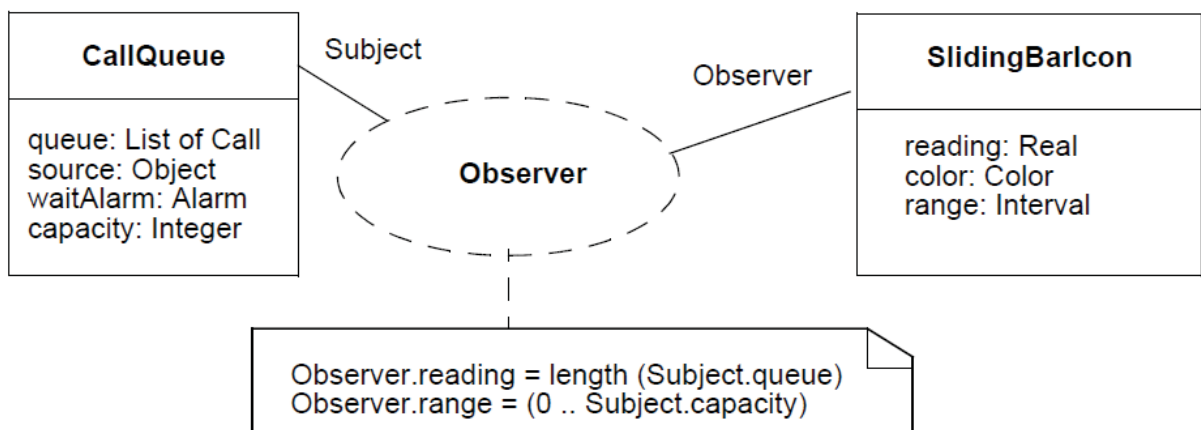


図 2.8: 合成構造図で表している Observer 協調

- 変更された図

- クラス図

内部構造とポートをもつように拡張されている。図 2.9 の (i) は、パッケージ Cars のクラス図である。この図中のクラス記号は、パッケージ Cars がクラス Car を包んでいることを表している。(ii) は、このクラス Car の合成構造図である。"rear:Wheel[2]" と "e:Engine" が内部構造のパートとして、"P" がポートとして含まれている。一般的にクラス記号にクラス名を書きただけである。一方、クラスの合成構造図には合成構造図の記号を書く。

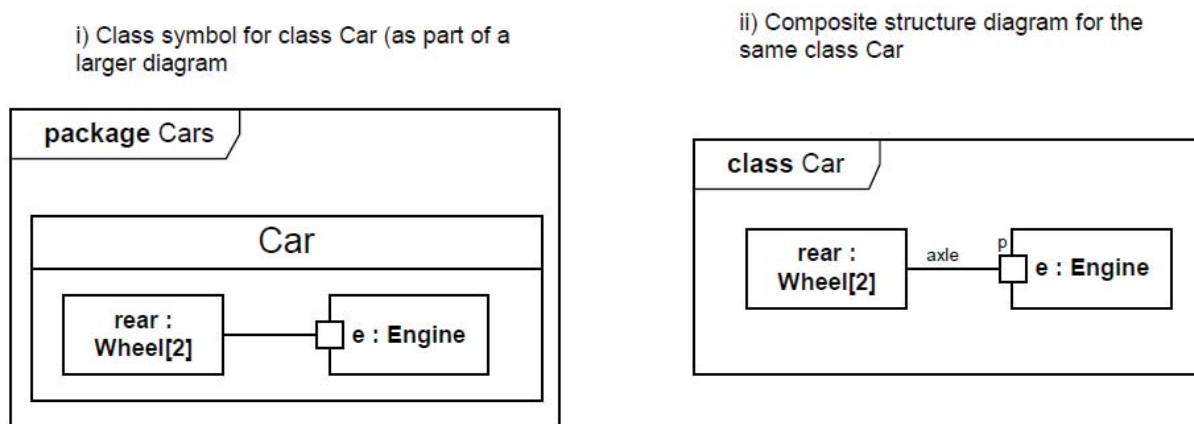


図 2.9: クラス図とそのクラスの合成構造図

- コンポーネント図

コンポーネントの振舞いを実現する内部分類子は、一般的な依存関係で表してもよい、代わりにコンポーネント記号内で入れ子状に表示してもよい。コンポーネントのつながりは、分類子とそれらの依存関係により、構造図で表せる。1つのコンポーネントは、2つの突き出た矩形をもたない分類子の記号によって表される。そして、コンポーネントの端のすべてのインタフェース記号の代わりに、「component」というキーワードが使われる。

- パッケージ図

マージ操作ができるようになった。

第3章 現存の依存生成法

3.1 現存の自動生成法

本節では、小谷が提案した自動生成法の手順を具体的に説明する。

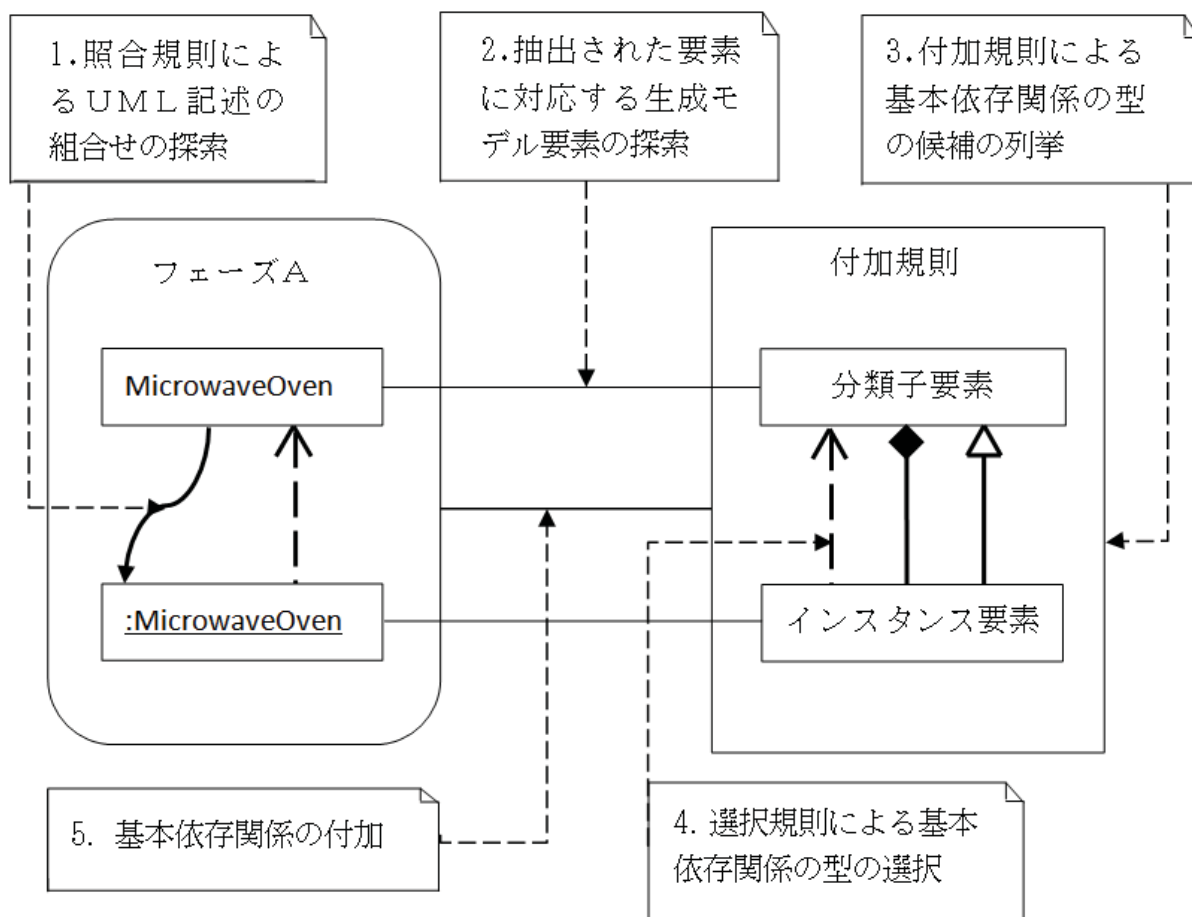


図 3.1: 自動生成法の手順

手順：

1. フェーズ情報と UML 図面群を自動生成モデルに入力し、UML 図面群における図やモデリング要素の名前と型名の類似性を分析し、照合規則により、この類似性をもつ UML 記述のすべての組合せを探索しておく。
2. 1 で探索された組合せのそれぞれはどの抽象的なグループ（生成モデル要素）に属するかを調べる。
3. 付加規則によって、2 で探索された 2 つの生成モデル要素間に、あらかじめどんな基本依存関係が設定されているかを調べる。
4. 3 で調べた、2 つの生成モデル要素間に存在可能な全ての基本依存関係のなかに、実際のフェーズ情報などを用いて、適切な基本依存関係を選択規則で決定する。

これで、依存関係の自動生成は成功した。

3.2 現存の自動生成法の問題点・改善策

「内部構造」という概念を対応するために、現存の自動生成法の改善点を分析しながら、解決方法を検討する。

- 用語定義の明確化
 - － 用語定義をより明確化に再定義・説明する必要がある。例えば、生成モデル要素は何なのか、UML 図とモデリング要素をどのように生成モデル要素とに分類するか。
- 照合規則
 - － UML 記述の一部が変更されたため、先行研究の照合規則の中に該当する UML 記述も修正しなければならない。例えば、UML2.4 では「コラボレーション図」が「コミュニケーション図」に変更され、「コラボレーション図」に関する照合規則を訂正する。また、「状態」という UML 要素は「単純状態」、「合成状態」と「サブマシン状態」に区別されようになったので、「状態」に関する照合規則を削除し、「単純状態」、「合成状態」と「サブマシン状態」のそれぞれに対応する照合規則を追加する。詳細は、第 6 章の照合規則で述べる。
 - － 内部構造を対応するように、依存関係が存在しえる UML モデリング要素または UML 図の組合せを分類してから、照合規則を拡張する。この分類は照合規則の分類法と呼ぶ。

- 付加規則
 - － 内部構造をもつ UML 図及び UML モデリング要素に対応する生成モデル要素を再定義してから、それらの間の基本依存関係を追加すると、付加規則の改良になる。
- 選択規則
 - － すでに定義されている選択規則は UML2.4 における内部構造に対応できるかどうかを検討する。

第4章 基本依存関係

4.1 UML2.4における依存関係のセマンティクス

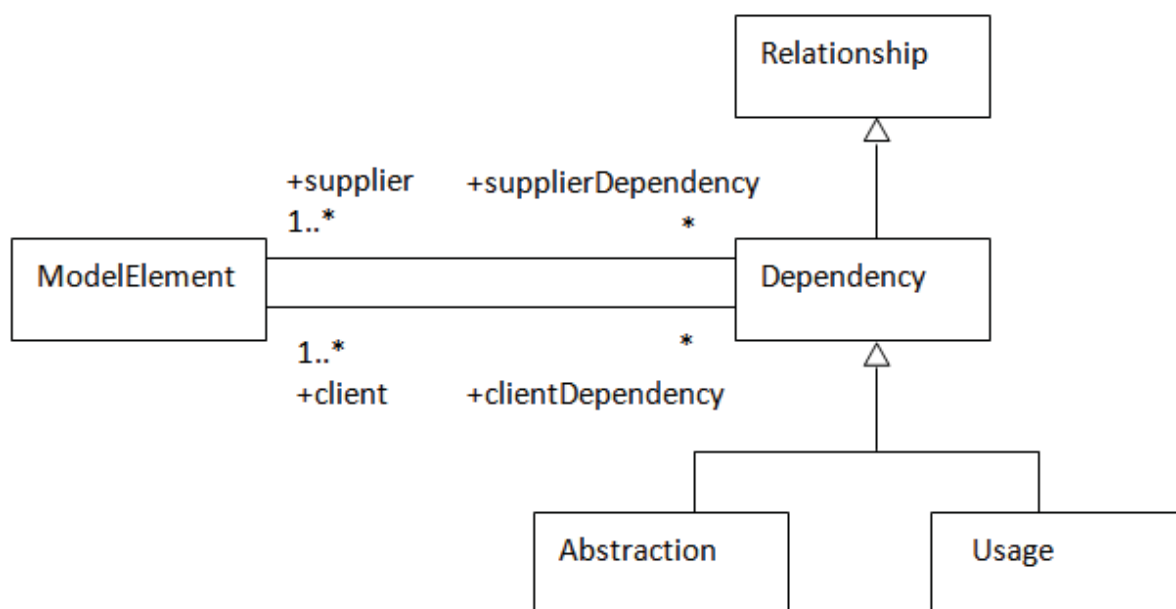


図 4.1: UML メタモデル：Core パッケージ-Dependencies

UML2.4では、Dependencyの下位に、「Permission」と「Binding」が削除され、「Abstraction」と「Usage」という2つの依存関係しか残っていない。

- **Abstraction**

ターゲットとソースの間のマッピングであり、異なる抽象レベルや異なる視点から、同じ概念を表す2つの要素や要素集合間の依存関係である。具体的には、「derive」「refine」と「trace」の3種類ある。

- **derive**

通常は同じ型だが、必ずしもそうとは限らないモデル要素間の派生関係を指定する。派生依存関係はターゲットがソースから計算できることを示す。

- **refine**

分析と設計のような異なるレベルにおける、モデル要素間の洗練関係を指定する。洗練は、分析から設計への変換あるいはその他の類似の変更をモデリングするために用いることができる。

- **trace**

異なるモデルにおいて同じ概念を表現するモデル要素あるいはモデル要素の集合間に存在するトレース関係を指定する。トレースは、主にモデル間で要求と変更を追跡するために用いられる。多くの場合は、計算可能でなく、非形式的である。

- **Usage**

ソースが完全な実装や操作のためにターゲットを要求する関係を示す。ソースがターゲットの定義や実行することにより使用されるということを強調する。具体的には「call」、「create」、「instantiate」、「responsibility」と「send」の5種類ある。

- **call**

起点を操作、終点を操作とする使用依存関係。この関係は、依存関係が適用される操作がクラス内存在することを示すために、その操作を含むクラスに対して組み込むこともできる。

- **create**

指定された特性が、この特性を付加された分類子のインスタンスを生成することを示す。さらにこの特性を含む分類子に適用されてもよい。

- **instantiate**

ターゲット側の操作がソースのインスタンスを生成することを示す分類子間の使用依存関係である。

- **responsibility**

ほかの要素との関係における要素の契約あるいは義務である。

- **send**

起点が操作であり、終点がシグナルであるような使用依存関係である。起点が終点に対しシグナルを送ることを示す。

- **包含関係**

内部構造の導入に対応する依存関係である。例えば、構造化クラスと内部構造間のように、構造化クラスが包含する側（ターゲット）で、内部構造が包含される側（ソース）であるため、包含関係が付加される。

- **コピー**

ターゲットはソースからコピーされた場合の依存関係である。

ほかに、設計者により、「コピー」という非形式的な依存関係が生成される。また、UML2.4では、分類子要素は内部構造をもつようになったので、「包含関係」という依存関係も生成される。

4.2 UML2.4 に対応の基本依存関係

依存関係といっても、様々な種類があるので、これを自動生成できるように明確に定義・分類しなければならない。その結果は基本依存関係と呼ぶ。つまり、実際に自動生成されるのは、基本依存関係である。小谷による UML1.5 を対象となって定義された基本依存関係を再掲する。

- **生存従属**

ソースはターゲットがないと存在しないことを示す。生存従属は、Usage と包含関係による依存関係から定義された。Usage と包含関係を示す依存関係は UML 記述の名前の比較や、包含の解析によって自動生成可能である。なので、生存従属は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**

ターゲットの情報がソースの情報の一部となることを示す。ターゲットとソースは何らかの“情報”を共有することである。“情報”は UML 記述の名前や属性、操作名などに対応する。情報共有は Usage から抽出された。Usage は UML 記述の名前が比較可能である場合に自動生成可能である。よって、情報共有は、共有される情報群の少なくとも 1 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**

ターゲットの全情報がソースの全情報となる場合に発生する。コピーされたものによる依存関係から抽出された。コピーは、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**

ターゲットの情報を元にソースの情報が作られる場合に発生する。同一概念は、Abstraction から抽出された。これはプロセス情報を利用し、UML 記述の名前を比較す

ることによって自動生成可能である。よって、同一概念は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。

また、情報共有と生存従属が共に成り立つ場合、情報共有のみ設定する。以上の分析を表 4.1 に示す。例えば、異なるレベルで同じ概念を表す要素間の関係を示す「Abstraction」は、基本依存関係の「同一概念」に対応し、開発プロセス情報を利用し、UML 記述の名前を比較することで自動生成可能である。

この 4 つの基本依存関係は、内部構造の導入による変更はないので、4.1 で分析した UML2.4 における依存関係に対応できるという結論ができた。

表 4.1: UML2.4 版対応の基本依存関係

依存関係		説明	基本依存関係	自動生成の方法
UML2.4 版で明確に定義された依存関係	Abstraction	異なるレベル/視点で、同じ概念を表す要素間の関係を示す。	同一概念	開発プロセス情報を利用して UML 記述の名前を比較する。
	Usage	ソースが完全な実装や操作のためにターゲットを要求する関係を示す。	情報共有	生存従属 UML 記述の名前を比較する。
その他	コピー	同じフェーズにある名前と型が一致する要素間の関係を示す。	コピー	UML 記述の名前を比較する。
	包含関係	構造化クラスなどクラスと内部クラス間の関係を示す。	生存従属	包含関係を解析し、UML 記述の名前を解析する。

第5章 付加規則

5.1 生成モデル要素の再分類

自動生成モデルはメタモデルのレベルで UML 記述間の依存関係を分析し抽出するので、UML 記述をより抽象的なレベル（メタモデルのレベル）でまとめなければならない。これを生成モデル要素である。

5.1.1 分類方法

UML 図とモデリング要素を別々に分析し、生成モデル要素にまとめる。

1. UML 図に関する生成モデル要素

UML2.4 版のスーパーストラクチャ記述で定義された「構造図と振る舞い図の分類」に参照して、生成モデル要素を作成する。

- **構造図**

クラス図、コンポーネント図、オブジェクト図、合成構造図、配置図、パッケージ図とプロファイル図は、静的な構造を表す UML 図なので、同じ UML 記述の型とみなし、ひとつのグループに分類する。「構造図」はこのグループの名前とする。

- **相互作用図**

振舞図の中に、シーケンス図、相互作用概要図、コミュニケーションズとタイミング図は、動的な振舞いを表すほか、オブジェクト間の相互作用も表現するので、同じ UML 記述の型とみなし、ひとつのグループに分類する。このグループの名前は「相互作用図」とする。

- **振舞図**

相互作用図以外の振舞図は、単に動的な振舞いを表すので、「振舞図」というグループにまとめる。

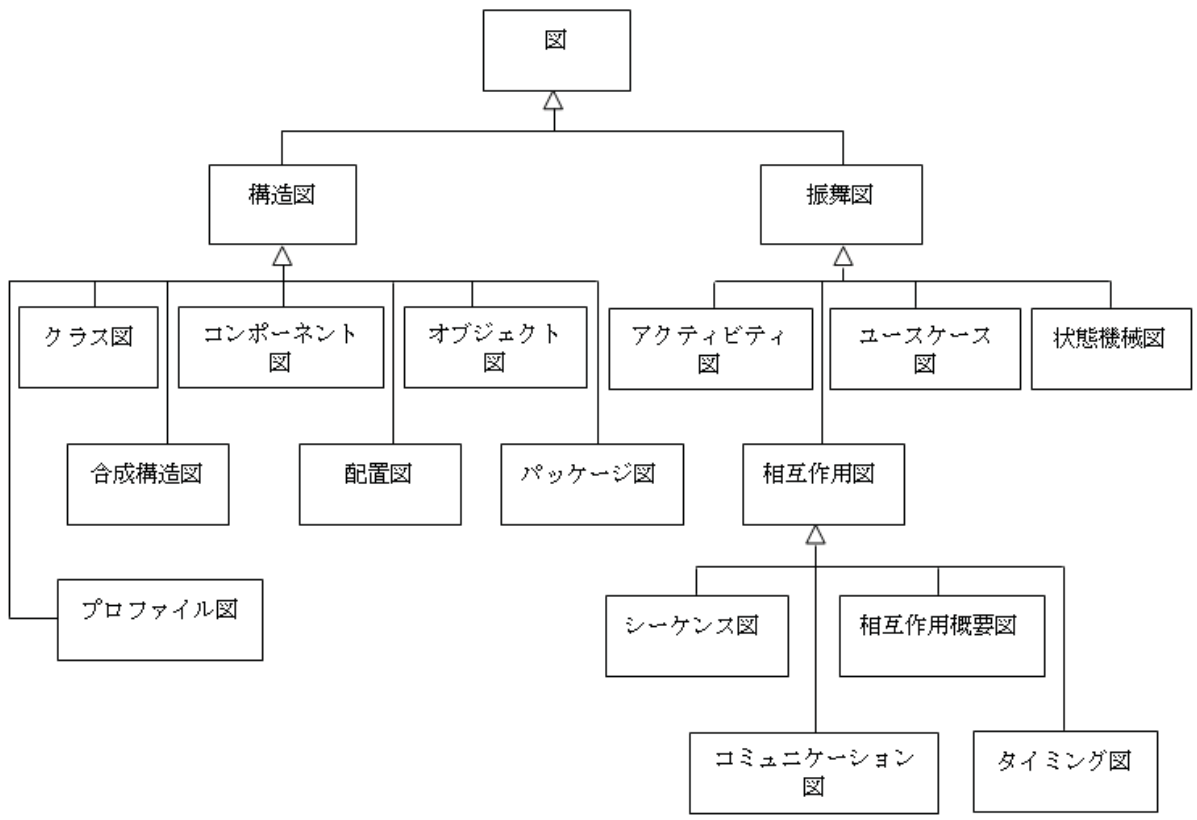


図 5.1: 構造図と振る舞い図の分類

上記の3つのグループを3種類の生成モデル要素として、表1にまとめる。

表 5.1: UML 図に関する3種類の生成モデル要素

分類対象	生成モデル要素	UML 図またはモデリング要素
UML 図	構造図	クラス図, コンポーネント図, オブジェクト図, 合成構造図, 配置図, パッケージ図, プロファイル図
	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
	振舞図	状態機械図, アクティビティ図, ユースケース図

2. UML モデリング要素に関する生成モデル要素

UML モデリング要素に関する6種類の生成モデル要素は以下のようになる。

表 5.2: UML モデリング要素に関する6種類の生成モデル要素

分類対象	生成モデル要素	UML 図またはモデリング要素
UML モデリング要素	分類子要素	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, 協調, 相互作用
	インスタンス要素	オブジェクト (コミュニケーション図, シーケンス図, 相互作用概要図, タイミング図), 協調使用, 相互作用使用, 配置指定
	状態要素	動作状態, 単純状態, 合成状態, サブマシン状態 (下位機械状態)
	関係要素	関連, 依存, 集約, 汎化, リンク
	遷移要素	遷移, イベント, アクション
	メッセージ要素	メッセージ

5.1.2 UML2.4 対応の生成モデル要素

以上9つの生成モデル要素を表5.3にまとめる。

表 5.3: UML2.4 対応の生成モデル要素

分類対象	生成モデル要素	UML 図またはモデリング要素
UML 図	構造図	クラス図, コンポーネント図, オブジェクト図, 合成構造図, 配置図, パッケージ図, プロファイル図
	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
	振舞図	状態機械図, アクティビティ図, ユースケース図
UML モデリング要素	分類子要素	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, 協調, 相互作用
	インスタンス要素	オブジェクト (コミュニケーション図, シーケンス図, 相互作用概要図, タイミング図), 協調使用, 相互作用使用, 配置指定
	状態要素	動作状態単純状態, 合成状態, サブマシン状態 (下位機械状態)
	関係要素	関連, 依存, 集約, 汎化, リンク
	遷移要素	遷移, イベント, アクション
	メッセージ要素	メッセージ

5.2 付加規則

本節では、「情報共有」、「コピー」、「同一概念」と「生存従属」の4種類の基本依存関係に対して、付加規則を拡張する。

● 情報共有

ターゲットの情報がソースの情報の一部となることを示す。「分類子要素とインスタンス要素」という組を例として分析する。例えば、協調と協調使用のように、ターゲットが「型」でソースがその「実体」の組発生する。「型」に対応する生成モデル要素を「分類子要素」として定義し、「実体」に対応する生成モデル要素を「インスタンス要素」として定義する。

- **コピー**

ターゲットの全ての情報がソースの全ての情報となる場合に発せする。どの生成モデル要素間でも存在しえる。なので、「生成モデル要素」と「生成モデル要素」間の基本依存関係である。

- **同一概念**

ターゲットの情報をもとにソースの情報で作成される場合に発生する。例えば、クラスとクラス図のように、ターゲットが「役割」でソースがその「構造」の組に発生する。「役割」に対応する生成モデル要素を「分類子要素」として定義し、「構造」に対応する生成モデル要素を「構造図」として定義する。

- **生存従属**

ある中間成果物が存在するために必要な中間成果物を示す場合に発生する。ソースはターゲットがないと存在しないことである。内部構造の導入による基本依存関係は主に生存従属となる。例えば、状態機械図とそのなかの単純状態のように、ターゲットが「包含する側」で、ソースが「包含される側」の組に発生する。なので、状態機械図を「構造図」という生成モデル要素に追加し、単純状態を「状態要素」という生成モデル要素に追加する。UML2.4に対応する付加規則の拡張は、主に生存従属に対する拡張である。

上記分析のように、すべての付加規則を拡張し、ターゲットとソースの順にまとめる。

- **情報共有**

- 分類子要素と振舞図の組
- 分類子要素とインスタンス要素の組
- インスタンス要素と振舞図の組
- インスタンス要素とインスタンス要素の組

- **コピー**

- 生成モデル要素と生成モデル要素の組

- **同一概念（詳細化）**

- 分類子要素と構造図の組
- 分類子要素と相互作用図の組
- 分類子要素と振舞図の組
- 生成モデル要素と生成モデル要素の組

- 分類子要素とインスタンス要素の組
- インスタンス要素と分類子要素の組
- 分類子要素と分類子要素の組

- **生存従属**

- 分類子要素と分類子要素の組
- 分類子要素とインスタンス要素
- 構造図と分類子要素の組
- 構造図と関係要素の組
- 振舞図と分類子要素の組
- 振舞図と振舞図の組
- 振舞図と状態要素の組
- 振舞図と遷移要素の組
- 相互作用図と分類子要素の組
- 相互作用図とインスタンス要素の組
- 相互作用図とメッセージ要素の組
- 状態要素と状態要素の組

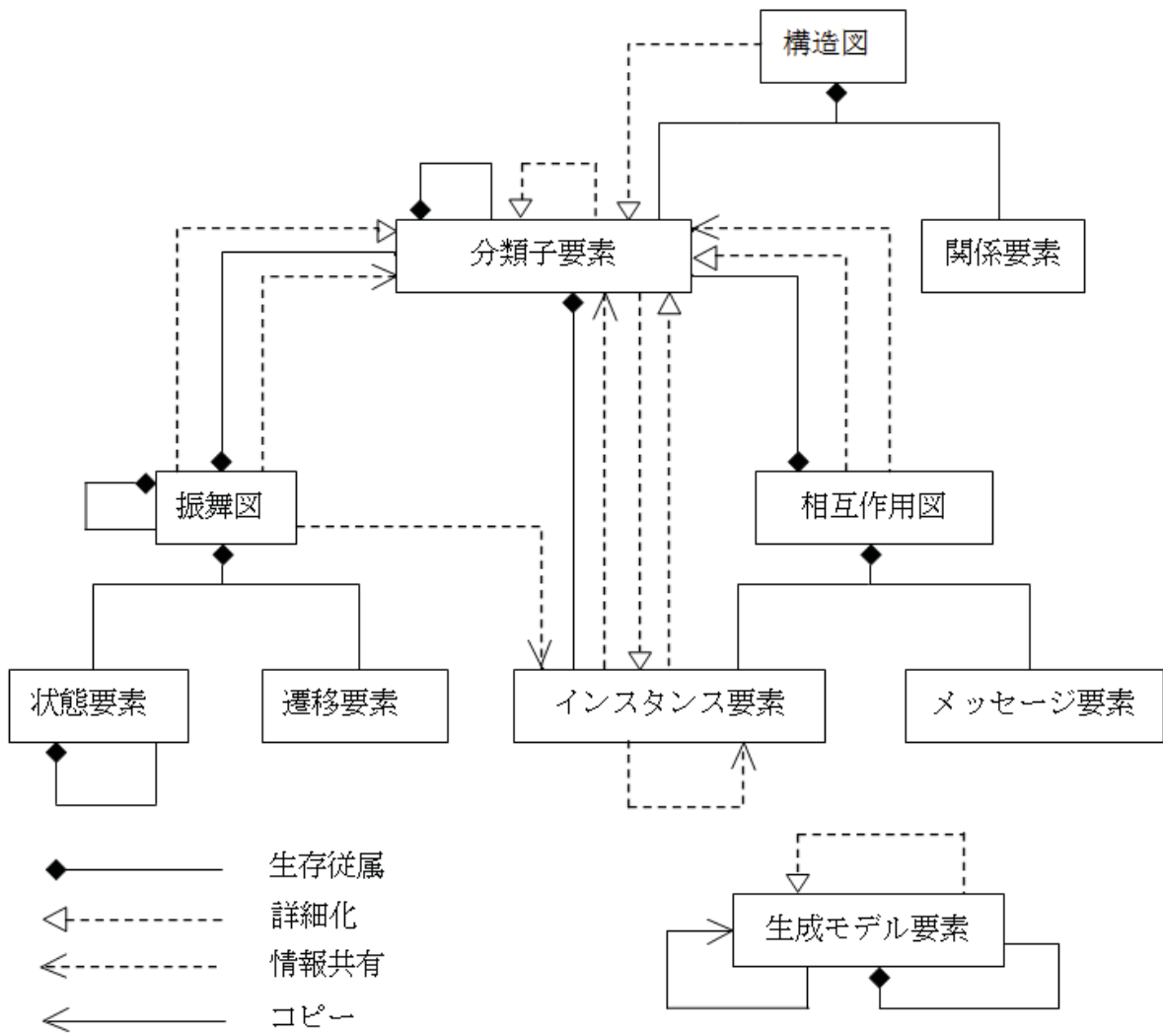


図 5.2: UML2.4 版に対応の付加規則

第6章 照合規則

照合規則はターゲット (T) とソース (S) の名前や型名の類似性を用いて、依存関係をもてる可能性のある UML 記述の組合せを定義する。この類似性を規定するのは照合条件である。すでに定義されている照合条件は以下の 5 種類ある。

- **Contained**

T の名前が S の名前に含まれる。例えば、ターゲットの名前が「電子レンジ」、ソースの名前が「電子レンジ制御」のとき、ソースの名前がターゲットの名前を含んでいるため、Contained と見なす。

- **Include**

T は、S に含まれる。例えば、ターゲットが構造化クラス、ソースがその内部構造の場合に発生する。

- **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。

6.1 分類方法

UML1.5 から UML2.4 への変更はたくさんあって、これらをなるべく自動生成モデルに反映するには、照合規則を明確的に分類・整理することが必要である。UML 図面群に対して、依存関係がもつ可能性がある UML 記述の組合せを探す規則である照合規則を拡張するために、特に分類・整理する方法が重要である。図 6.1 はこの分類方を示している。依存する側 (ターゲット) から依存される側 (ソース) へ破線矢印で依存関係を表現する。例えば、「図 1 ----> 図 2」は、図 2 が図 1 を依存していることを意味する。

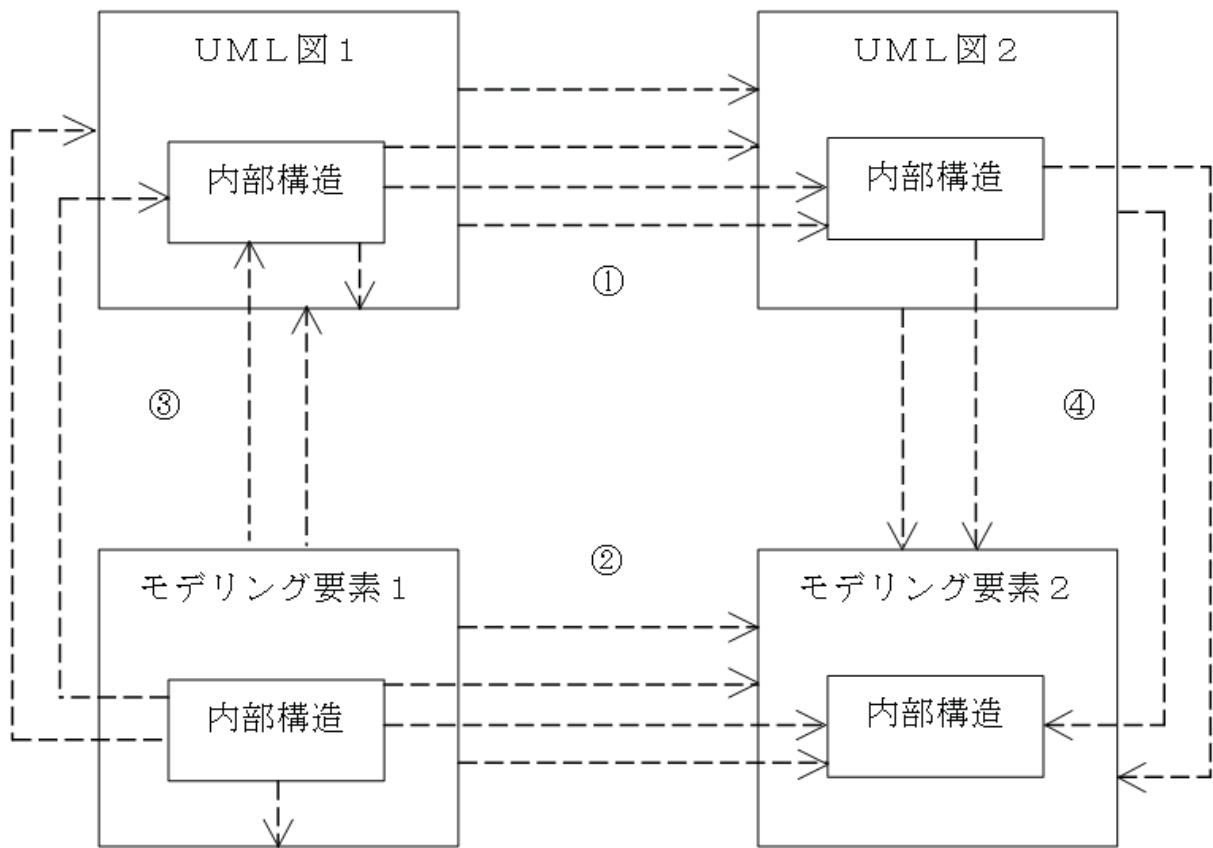


図 6.1: 照合規則の分類方法

1. 図同士の間

- 図 1 -----> 図 2
- 図 1 -----> 図 2 の内部構造
- 図 1 の内部構造 -----> 図 2
- 図 1 の内部構造 -----> 図 2 の内部構造
- 図 1 の内部構造 -----> 図 1

2. モデリング要素同士の間

- モデリング要素 1 -----> モデリング要素 2
- モデリング要素 1 -----> モデリング要素 2 の内部構造
- モデリング要素 1 の内部構造 -----> モデリング要素 2
- モデリング要素 1 の内部構造 -----> モデリング要素 2 の内部構造
- モデリング要素 1 の内部構造 -----> モデリング要素 1

3. モデリング要素と図の間

- モデリング要素 -----> 図
- モデリング要素 -----> 図の内部構造
- モデリング要素の内部構造 -----> 図
- モデリング要素の内部構造 -----> 図の内部構造

4. 図とモデリング要素の間

- 図 -----> モデリング要素
- 図 -----> モデリング要素の内部構造
- 図の内部構造 -----> モデリング要素
- 図の内部構造 -----> モデリング要素の内部構造

上記の 4 組は理論的に依存関係が存在する UML 記述（図とモデリング要素）の間である。これは照合規則を拡張する手がかり、あるいは照合規則の分類方法となる。次節では、重複を解消したり実際のモデリングに存在しえない間を削除したりしながら、UML2.4 版に対応する照合規則をまとめる。

6.2 照合規則

1. 図同士の組

- 図1 ----> 図2

従来の内部構造の持たない図同士の依存関係はこの組にあたる。それに新規追加された図、または変更された図を分析し、この組にあたる照合規則を拡張する。

表 6.1: 照合規則：図1 ----> 図2

ターゲット	ソース	照合条件
クラス図	クラス図	Similar
クラス図	オブジェクト図	SimType
クラス図	コンポーネント図	Similar, Contained

- 図1 ----> 図2の内部構造

隣接フェーズにおける図面間の組合せが多い。例えば、分析フェーズで作られたオブジェクト図は、設計フェーズでクラス図内に含まれるようになる。またはあるクラス図にもとづいて、ほかのクラス図の内部構造が作られるなど。

表 6.2: 照合規則：図1 ----> 図2の内部構造

ターゲット	ソース	照合条件
クラス図	オブジェクト図	Include
コンポーネント図	クラス図	Include
パッケージ図	クラス図	Include
状態機械図	状態機械図	Include
ユースケース図	状態機械図	Include
ユースケース図	クラス図	Include

- 図1の内部構造 ----> 図2

ある図の内部構造に基づいて、ほかの図が作られる場合にあたる。ただし、この場合は実際の作業には非常に少ない。

- 図1の内部構造 ----> 図2の内部構造

この組は実際に少ない。

- 図1の内部構造 ----> 図1

図とその中の関係なので、特に分析しない。

2. モデリング要素同士の組

- モデリング要素1 -----> モデリング要素2
内部構造を持たない場合のモデリング要素について分析すればよい。

表 6.3: 照合規則：モデリング要素1 -----> モデリング要素2 の内部構造

ターゲット	ソース	照合条件
協調使用	協調	TypeSim
協調	協調使用	SimType
相互作用	相互作用使用	Contained, Similar

- モデリング要素1 -----> モデリング要素2 の内部構造

表 6.4: 照合規則：モデリング要素1 -----> モデリング要素2 の内部構造

ターゲット	ソース	照合条件
クラス	クラス	Similar, Include
クラス	相互作用	Include
パッケージ	クラス	Include
パッケージ	ユースケース	Include
コンポーネント	コンポーネント	Include
ノード	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
サブマシヨン状態	シンプル状態	Include
合成状態	シンプル状態	Include
ノード	相互作用使用	Include

- モデリング要素1 の内部構造 -----> モデリング要素2
実際の作業には少ない。
- モデリング要素1 の内部構造 -----> モデリング要素2 の内部構造
図1の内部構造 -----> 図2の内部構造の組と同じように、実際に存在しえないとみなす。
- モデリング要素1 の内部構造 -----> モデリング要素1
内部構造をもてるモデリング要素の場合であるが、モデリング要素とその中身の関係なので、特に依存関係を生成する必要はない。

3. モデリング要素と図の組

- モデリング要素 ----> 図

表 6.5: 照合規則：モデリング要素 ----> 図

ターゲット	ソース	照合条件
配置図	配置指定	SimType
状態機械図	クラス	Simlar
状態機械図	アクション	Simlar
合成構造図	クラス	Similar, Contained

- モデリング要素 ----> 図の内部構造
内部構造をもてる図に対して、参照先のモデリング要素が変更されたら、参照元の図も変更しなければならない。表 6.6 にまとめている。
- デリリング要素の内部構造 ----> 図
こちらの組は「デリリング要素 ----> 図」の一部と「図 ----> 図」の一部と重なっている。
- モデリング要素の内部構造 ----> 図の内部構造
実際の作業には少ない。

4. 図とモデリング要素の組

- 図 ----> モデリング要素表 6.7 にまとめている。
- 図 ----> モデリング要素の内部構造
「図 ----> モデリング要素」の一部と重なっている。
- 図の内部構造 ----> モデリング要素
実際の作業には少ない。
- 図の内部構造 ----> モデリング要素の内部構造
実際の作業には少ない。

以上の分析をまとめると、実際に存在しえる照合規則の組合せは以下の6種類である。

- 図 1 ----> 図 2
- 図 1 ----> 図 2 の内部構造
- モデリング要素 1 ----> モデリング要素 2

表 6.6: 照合規則：モデリング要素 ----> 図の内部構造

ターゲット	ソース	照合条件
オブジェクト図	オブジェクト	Include
クラス図	クラス	Include
クラス図	パッケージ	Include
クラス図	相互作用	Include
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
パッケージ図	パッケージ	Include
パッケージ図	クラス	Include
コンポーネント図	コンポーネント	Include
コンポーネント図	クラス	Include
コミュニケーション図	オブジェクト	Include
シーケンス図	オブジェクト	Include
シーケンス図	メッセージ	Include
配置図	ノード	Include
状態機械図	シンプル状態	Include
状態機械図	合成状態	Include
状態機械図	サブマシオン状態	Include
アクティビティ図	動作状態	Include
相互作用概要図	相互作用使用	Include
相互作用概要図	相互作用	Include
合成構造図	協調	Include
合成構造図	協調使用	Include
タイミング図	シンプル状態	Include
タイミング図	メッセージ	Include

表 6.7: 照合規則：図 ----> モデリング要素

ターゲット	ソース	照合条件
ユースケース	クラス図	Contained
ユースケース	アクティビティ図	Contained
ユースケース	状態機械図	Include
クラス	アクティビティ図	Contained
クラス	アクティビティ図	Contained
クラス	合成構造図	Similar, Contained
クラス	状態機械図	Contained
クラス	状態機械図	Contained
クラス	合成構造図	Similar, Contained
オブジェクト	タイミング図	TypeSim
オブジェクト	状態機械図	TypeSim
ユースケース	コミュニケーションズ	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	状態機械図	Contained
配置指定	配置図	TypeSim

- モデリング要素1 -----> モデリング要素2の内部構造
- モデリング要素----->図
- 図-----> モデリング要素

この6種類に関する照合規則をまとめて、UML2.4に対応する照合規則となる。詳細は、付録にある。

第7章 選択規則

生成モデルの組合せに複数の基本依存関係が付加されている場合は、プロセス情報などの基準を用いて、適切な基本依存関係意を選択する。これは選択規則と呼ぶ。用いられる基準は、フェーズ、図面、UML モデリング要素の型、名前という 4 つである。

- **生存従属**

- フェーズ：異なる
- 図面：異なる
- 名前：同じ図面ならば同じでも異なっても可能である。異なる図面ならば同じである。
- UML 記述：名前と同じ

- **情報共有**

- フェーズ：同じ
- 図面：異なる
- UML 記述：異なる

- **コピー**

- フェーズ：同じ
- 図面：異なる
- UML 記述の型：同じ
- 名前：同じ

- **同一概念**

- フェーズ：隣接する

上記の結果をまとめて, 表 7.1. になる。

図 7.1: UML2.4 に対応できる選択規則

		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	生存従属		同じ	名前
			—	同一概念	—	異なる	
	情報共有		同じ				
	異なる	同じ	異なる			異なる	
図面							

第8章 終わりに

8.1 まとめ

本論文では,UML1.5 版から最新版である UML2.4 版へ変更を分析しながら,UML1.5 版が対象となった依存関係の自動生成モデルを最新版に対応するように拡張・改善した。

改善点&問題点

- **UML2.4 版の追加点・変更点の洗い出し**

新規追加や変更された図と要素を洗い出して、分析した。

- **現存自動生成モデルの必要な拡張とその手段の検討**

内部構造を対応するために、自動生成モデルの改善点と対策を検討した。

- **用語定義の明確化**

現存の自動生成モデルに関する用語定義（生成モデル要素, 付加規則, 照合規則, 選択規則など）にはある程度の曖昧さがある。これらを理解しやすく明確的に再定義・再整理した。

- **付加規則の改良**

基本依存関係を検討した上で、必要な生成モデルを再定義し, それらの間の基本依存関係を追加した。

- **照合規則の改良**

照合規則の拡張だけではなく, 照合規則の分類・拡張方法も提案した。この提案により, 今後 UML の拡張などによる照合規則の拡張にも非常に役立つ。

しかし, 以上拡張した自動生成モデルはどの程度で改善されたか, また依存関係を自動生成する精度と正確度など評価していないという問題点がある。

8.2 今後の課題

今後の課題は以下の2つについて検討してもよい。

- 本論文で拡張した自動生成モデルの有効性を検討し, 依存関係を自動生成する精度と正確度を検証する。
- UP 以外の開発方法論に適用する自動生成モデルを修正・拡張する。

謝辞

本研究を遂行するにあたり、終始熱心なご指導を賜りました落水 浩一郎教授に深謝いたします。また、修士中間審査において貴重なごアドバイスを賜りました鈴木 正人准教授、青木 利晃准教授に深く感謝いたします。

また、本論文を日本語で作成できるのは、丁寧に日本語を教えていただいた山口 実千代先生、鹿野 緑先生、寺 朱美先生のおかげです。心より感謝いたします。

そして、日頃から有益なご助言をいただき、多面にわたって支援していただきました、吉田 博幸様、新海 卓夫様およびアジア人財事業部の方々に心より感謝いたします。

本論文をまとめるにあたり、ご協力いただきました落水・鈴木研究室の先輩・同期・後輩の皆様に、深く御礼申し上げます。

最後に、いつも暖かく応援していただきました家族と友達に、深く御礼申し上げます。

参考文献

- [1] 小谷正行, 落水浩一郎, 依存関係生成モデルを用いたソフトウェア成果物の変更波及解析支援, 情報処理学会論文誌, Vol. 49, No. 7, pp2265–2291, 2008.
- [2] Object Management Group, UML Infrastructure Specification, Version 2.4.1, <http://www.omg.org/spec/UML/2.4.1/Infrastructure/PDF/>
- [3] Object Management Group, UML Superstructure Specification, Version 2.4.1, <http://www.omg.org/spec/UML/2.4.1/Superstructure/PDF/>
- [4] Object Management Group 著, 西原裕善監訳, UML2.0仕様書 UML2.1対応, オーム社, 2006.
- [5] 金井 健太郎, UML 図面要素間の依存関係の自動生成法, 2009.

付録 UML2.4対応の照合規則

太字で表示されている照合規則は新規追加である。

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	アクティビティ図	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	SimTyp, 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	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
シーケンス図	オブジェクト	Include

ターゲット	ソース	照合条件
クラス図	クラス図	Similar
状態機械図	クラス	Similar
状態機械図	アクション	Similar
相互作用	相互作用	Similar
合成構造図	クラス	Similar
相互作用	相互作用使用	Similar
クラス図	コンポーネント図	Similar
クラス	合成構造図	Similar
クラス図	オブジェクト図	SimType
配置図	配置指定	SimType
協調	協調使用	SimType
協調使用	協調	TypeSim
オブジェク	タイミング図	TypeSim
オブジェクト	状態機械図	TypeSim
配置指定	配置図	TypeSim
クラス図	コンポーネント図	Contained
クラス	合成構造図	Contained
クラス	状態機械図	Contained
ユースケース	相互作用概要図	Contained
ユースケース	状態機械図	Contained
ユースケース	コミュニケーション図	Contained
合成構造図	クラス	Contained
相互作用	相互作用使用	Contained
コンポーネント図	クラス図	Include
パッケージ図	クラス図	Include
パッケージ図	パッケージ	Include
パッケージ図	クラス	Include
状態機械図	状態機械図	Include
ユースケース図	クラス図	Include
相互作用	オブジェクト	Include
サブマシン状態	シンプル状態	Include
合成状態	シンプル状態	Include
ノード	相互作用使用	Include
クラス図	相互作用	Include

ターゲット	ソース	照合条件
クラス図	オブジェクト図	Include
クラス	相互作用	Include
コンポーネント図	クラス	Include
コミュニケーション図	オブジェクト	Include
シーケンス図	メッセージ	Include
状態機械図	シンプル状態	Include
状態機械図	合成状態	Include
状態機械図	サブマシヨン状態	Include
アクティビティ図	動作状態	Include
相互作用概要図	相互作用使用	Include
相互作用概要図	相互作用	Include
合成構造図	協調	Include
合成構造図	協調使用	Include
タイミング図	シンプル状態	Include
タイミング図	メッセージ	Include
ノード	相互作用	Include