

構造および語の類似性に基づくアナリシスパターンの自動抽出

野本悠太郎
早稲田大学理工学術院
基幹理工学研究科
情報理工学専攻

ytr-n

@akane.waseda.jp

久保淳人
青山メディア研究所
kubo@nii.ac.jp

鷲崎弘宜
早稲田大学理工学術院
基幹理工学研究科
情報理工学専攻

washizaki@waseda.jp

深澤良彰
早稲田大学理工学術院
基幹理工学研究科
情報理工学専攻

fukazawa@waseda.jp

あらまし

ソフトウェア開発における分析の工程においては、アナリシスパターンが利用される。しかし、アナリシスパターンを既存の資産から抽出するにはコストがかかるため、要求やモデルの資産が蓄積されているにも関わらずパターンが抽出されていないことがある。そこで我々は、アナリシスパターンを自動抽出する手法を提案する。本手法においては、対応した要求とモデルの資産の集合を構造および語の類似性に基づいて解析し、共通部分を特定することによって、アナリシスパターンを自動抽出する。本手法を用いることで、コストをかけることなくアナリシスパターンを抽出することができる。

Keywords

アナリシスパターン, パターン抽出, UML(Unified Modeling Language), 概念辞書 (シソーラス)

1. はじめに

オブジェクト指向のソフトウェア開発においては、製品に関する要求の理解性を向上させるためにモデリングを活用する。開発の初期段階である分析工程においては、モデルを作成して開発対象の要求を様々な方向から捉える。分析を経験していくにつれ、似たような要求からモデルを作成することが起こりうる。そのため、分析段階ではアナリシスパターン(分析パターン)を用いる[1]。アナリシスパターンとは、優れたモデルを導出する過程をパターン化したものである。分析時に頻出するパターンであり、実際のソフトウェアの実装ではなく、開発に対する要求と、開発するシステムの構造を反映した典型的なモデルを示したものである。

アナリシスパターンを用いることでモデルの作成が容易になり、ソフトウェアの再利用性向上が実現できる。現在、数多くのアナリシスパターンが定義されている[1][2][3]。アナリシスパターンには、ドメインに特化した特化型パターンと、ドメインを超えて再利用可能な汎用型パターンが存在する。しかし、アナリシスパターンの抽出はワークショップなど人手を介したものでありコストがかかる。また、アナリシスパターンを抽出する際には、抽出元の資産の構成に応じて、共通ドメインから集められていればそのドメインに特化したパターンを、異なる様々なドメインから集められていればドメインを超えて再利用可能な汎用的なパターンをそれぞれ抽出できることが望ましい。

そこで我々は、対応した複数の要求とモデルの集合から共通部分を特定してアナリシスパターンを自動抽出する手法を提案する(以下、本手法とする)。対応した複数の要求とモデルとは、同じソフトウェア開発の元となった要求記述などの要求と、クラス図などのモデルである。モデル資産のみからのアナリシスパターン抽出では、要求やドメインの記述との関連が分からないため、モデルと要求を対応付けた抽出が必要と考えられる。ドメインによっては要求とモデルが対応付けられ、資産として蓄積されている。

また、単に共通部分を特定するのみでは汎用的なパターンは抽出できないため、クラス名間の類似度を測定する。これにより、モデルを抽象化して扱うことができるようになるため、異なるドメインを入力として与えた場合は、汎用的なパターンが抽出できる。さらに、パターンの適用を支援するため、パターンの具象度を定義することで、抽出したパターンの特化・汎用の程度を定量的に区別する。

モデルのみを提示しても、要求やドメインの記述との関連が分からず、効率的な再利用が可能なアナリシスパターンとなりえないため、関連している要求を提示する。その際、異なるドメインの記述が混在してしまうため、関連している要求を定量化することで、パターン適用の参考とする。本手法を用いることで、コストをかけることなく特化型と汎用型のアナリシスパターンを抽出することができる。と考えられる。

以降、本稿では2章でアナリシスパターンの背景と抽出に関する問題を述べる。3章で要素技術や具体的な手法を説明する。4章で実験結果の評価と考察を検討する。5章で関連研究を挙げる。6章で総括と今後の課題を示す。

2. アナリシスパターンの抽出に関する問題

2.1 アナリシスパターン

オブジェクト指向開発においては、例えばUML(Unified Modeling Language)のクラス図を用いてモデルを作成する。モデルとは、要求の特徴を捉え抽象化したシステムのモデルである。分析段階で用いられるソフトウェアパターンをアナリシスパターンと呼ぶ。アナリシスパターンは、ソフトウェア開発を行うドメインで必要となるデータや概念を明確化し、要求や対象世界の記述からモデルを作成する過程をパターン化する。

多くのソフトウェアパターンは、特定のドメインや実装技術に特化したものと、汎用的なものに分けることができる。汎用的なアナリシスパターンの基本となる「もの-こと-もの」パターンの構造を図 1 に示す[1]。単純に主体となる「もの」と対象となる「もの」を関連付けるのではなく、記録すべき「こと」を概念として書き出すことで、変更しやすくわかりやすいモデルの作成を支援する。

ドメインに特化したアナリシスパターンとして、Martin Fowler の『アナリシスパターン』が挙げられる[2]。例として、「責任関係の知識レベルパターン」の構造を図 2 に示す[3]。「責任関係の知識レベルパターン」に対応する要求として、(1)組織構造に階層あるいはより複雑なリンクがある、(2)新たな種類のリンクが現れる、(3)構造の変更履歴を保持する、が挙げられる。「責任関係の知識レベルパターン」を適用する例題として、勤怠システムのモデル作成を考える。このシステムには組織と人、組織と組織の関係が含まれ契約や雇用が関係するので、「責任関係の知識レベルパターン」が適用できる。派遣先や派遣者が変動してもそれが追跡可能であることを考慮に入れた組織構造のモデルを作成することができ、パーティ間に柔軟性を持たせた構造をモデリングすることができる。

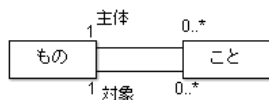


図 1. 「もの-こと-もの」パターンの構造[1]

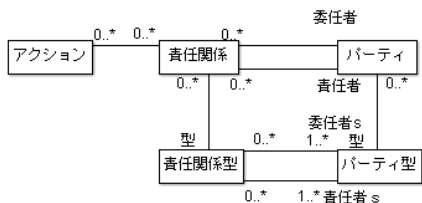


図 2. Fowler の「責任関係の知識レベル」パターンの構造[3]

2.2 現状のアナリシスパターン抽出の問題点

アナリシスパターン抽出に関する問題点を挙げる。

- アナリシスパターンは抽出し尽くされていない可能性がある
- パターンの抽象度が適切に示されていない
- モデルのみからでは、適切なパターンが抽出できない

これまでアナリシスパターンは数多く定義されているが、ソフトウェア開発におけるドメインの多様さや進展の速さにより、特化型のパターンは抽出されていない可能性がある。また、汎用型のパターンについても抽出し尽くされていない可能性が考えられる。しかし、現在アナリシスパターンを自動で抽出する有効な仕組みがないため、アナリシスパターンの抽出にはコストがかかり、要求とモデルの資

産が蓄積されているにも関わらずアナリシスパターンが抽出されていないことが多い[4][5]。

さらに、抽象度の観点から、抽出元の資産の構成に応じて、共通ドメインから集められていればそのドメインに特化したパターンを、異なる様々なドメインから集められていればドメインを超えて再利用可能な汎用的なパターンをそれぞれ抽出できる必要がある。図 1 と図 2 を区別するように、抽出したパターンの特化・汎用の度合を示すことも望まれる。

また、アナリシスパターンにおいては、要求段階で得られた情報をパターン化することでモデル作成を簡略化する。そのため、モデル資産のみからのアナリシスパターン抽出では、要求やドメインの記述との関連が分からず、効率的な再利用が可能なアナリシスパターンとなりえない。前述の例題を考える場合、図 2 のみではパターンを適用する背景との関連が理解できないためモデルの作成が困難となる。よって、モデルと要求を対応付けた抽出が必要と考えられる。

我々は、これらの問題を解決するため、アナリシスパターンを自動抽出する手法を提案する。また、対応した要求とモデルの資産からの抽出を行うことによって、適切なパターンを抽出し、その抽象度も示す。

3. 構造および語の類似性に基づくアナリシスパターンの自動抽出

アナリシスパターン抽出にかかるコストの問題を解消するため、我々はアナリシスパターンを自動で抽出する手法を提案する。入力となる資産 Set_x を式(1)で定義する。引数 x は、入力する資産を表わす。複数の対応付いたモデル M_x と要求 D_x の資産の組を入力として、パターンのモデル構造と対応した要求記述が出力される。

$$Set_x = (M_x, D_x) \quad (1)$$

本手法における対応付いた要求とモデルの資産とは、同じソフトウェア開発の元となった要求と UML クラス図の組である。対応付いた要求とモデルの例として、飛行船システムにおける要求とモデルを図 3, 4 に示す。図 3 における記述が図 4 において抽象化してモデリングされている。本システムは Java で実装され、要求をテキストファイル形式(.txt)に、UML クラス図を XMI 形式(.xmi)に変換して入力する。モデル資産のみからでもパターンを抽出することができるが、要求やドメインの記述との関連が分からず効率的な再利用が可能なアナリシスパターンが抽出できない。

本手法においては、対応付けられた要求とモデルの資産の集合から共通部分を特定して、アナリシスパターンを自動抽出する。入力として与えるモデル資産の構成に応じて、共通ドメインから集められていればそのドメインに特化したパターンを、異なる様々なドメインから集められていればドメインを超えて再利用可能な汎用的なパターンをそれぞれ抽出する。パターンを抽出する際はモデルを抽象化し

てそれらの類似性を求めてモデルを汎用的に扱うため、異なる様々なドメインから集められたモデルでも共通部分を特定できる。

例として、図5に示した2つのモデル「エレベーターフロア表示」（エレベータシステム）と「飛行制御-画面表示」（図4の飛行船システムから抜粋）について考える。この2つのモデルの場合、「エレベータ」と「飛行制御」が概念的にコントローラで類似し、「フロア表示」と「画面表示」が概念的にディスプレイで類似している。このため、アナリシスパターンのモデル構造として、抽象化されたモデル「コントローラ-ディスプレイ」が抽出できる。

本手法においては、モデルとしてUMLクラス図を扱うため、モデルがクラスの集合として構成されている。入力となるモデル M_x を式(2)で定義する。モデル M_x は、クラスの集合 C_x と各クラス間の関連の集合 R_x の組と考えることができる。

$$M_x = (C_x, R_x) \quad (2)$$

$$C_x = \{c_{x1}, c_{x2}, \dots, c_{xf}\}$$

$$R_x = \{r_{x1}, r_{x2}, \dots, r_{xg}\} \subseteq C_x \times C_x$$

※ f : C_x の要素数

※ g : R_x の要素数

モデルの類似性を求めるためにはクラス名間の類似度を測定する必要がある。クラス名 cn_{xf} と cn_{xh} に含まれる単語 cw_{xfh} を式(3)で定義する。クラス名 cn_{xf} はクラス C_x の要素と考えられる。また、クラス名 cn_{xf} には単語 cw_{xfh} が含まれている。

$$cn_{xf} \in C_x \quad (3)$$

$$K(cn_{xf}) = \{cw_{xf1}, cw_{xf2}, \dots, cw_{xfh}\}$$

※ $K(cn_{xf})$: cn_{xf} を形態素解析し得られる単語の集合

※ h : $K(cn_{xf})$ の要素数

クラス名間の類似度は単語間の意味の類似度で求められる。単語間の意味の類似度を定量的に表す方法として、本手法においては単語間の類似度を測定する。クラス名に使われている単語間の類似度が高いほど、クラス間の類似度が高いと判断できる。図5の例の場合「フロア表示」と「画面表示」は単語間の類似度が高いため、概念的に「ディスプレイ」で類似したと考えられる。

モデルの集合からは概念的に類似したモデル構造が抽出される（例えば図5の「コントローラ-ディスプレイ」）。

また、特化・汎用の度合を定量的に示すため、概念的に類似した構造の具象度を定義し測定する。具象度を利用することにより、扱うドメインにそのパターンが適用可能か判定できる。

さらに、概念的に類似した構造と対応している要求の記述をそれぞれの要求から選択する。異なるドメインの記述が混在してしまうため、選択された各要求の集合から、概念的に類似した構造との関連の強さを定量化して表すことで、パターン適用の参考とする。これにより、抽出された構造を用いる背景となる要求がわかりやすくなるため、より効率的なアナリシスパターンとなる。

本手法の全体像を図6に示し、概要を以下に記述する。以降の節においてそれぞれ詳細に説明する。

3.1 単語の抽出：要求記述の中で、利用されている単語を抜き出す。

3.2 クラス名の限定：3.1で抜き出された単語を名前に持つクラスに限定する。

3.3 クラス名間の類似度の測定：クラス名間の類似度を測定することで、入力したモデル（クラスの集合）間の類似度を測定する。

3.4 類似した構造の抽出：3.3の結果から、入力したモデル間において類似した構造を抽出する。

3.5 具象度の測定：類似した構造の特化・汎用の度合を定量的に示すため、具象度を定義し測定する。

3.6 要求の定量化：類似した構造と入力した要求との関連度合を定量化する。

- ・ 自律飛行を開始する。
- ・ タイマを起動させる。
- ・ 飛行制御を行う。
- ・ コースデータを読み取る。
- ・ コースデータを入力する。
- ・ センサで受けた情報から現在の飛行船の状況を画面表示する。
- ・ . . .

図3.要求記述の例

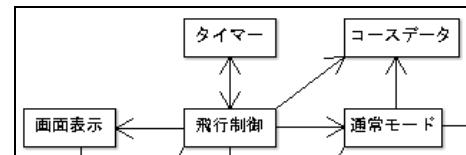


図4.モデルの例

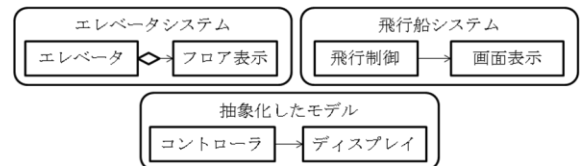


図5.モデルを抽象化した例

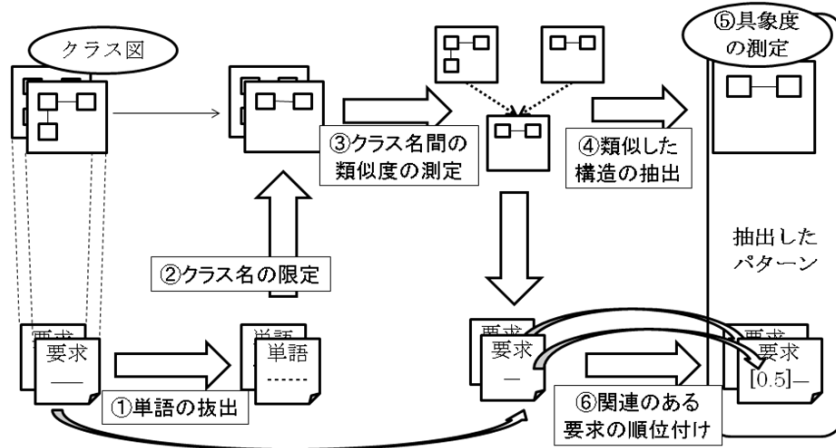


図 6.提案手法の全体像

3.1 単語の抽出

要求記述は、句点や改行で分割し、シナリオ S_{xi} の集合として扱うことができる。各シナリオ S_{xi} を形態素解析することにより、それぞれの要求において利用されている、すべての単語 w_{xjv} を抜き出すことができる。入力された要求 D_x を式(4)で定義する。

$$D_x = \{S_{x1}, S_{x2}, \dots, S_{xi}\} \quad (4)$$

$$K(S_{xj}) = \{w_{xj1}, w_{xj2}, \dots, w_{xjv}\}$$

※ $K(S_{xj})$: S_{xj} を形態素解析し得られる単語の集合

※ u : D_x の要素数

※ v : $K(S_{xj})$ の要素数

3.2 クラス名の限定

3.1 において抜き出した単語の集合 $K(S_{xj})$ の要素が、クラス名に含まれている単語の集合 $K(cn_{xp})$ の要素と、1つでも一致するクラスのみを扱うことによって、要求とモデルの対応が取れたアナリシスパターンを抽出することが可能となる。

モデルの集合のみからであっても抽象化されたモデルを抽出することはできるが、抽出したモデルを適用する際の要求が明確にならないため、効率的な再利用が可能なアナリシスパターンとならない。

3.3 クラス名間の類似度の測定

本手法においては 2 単語間の類似度を測定するために、WordNet[6]をもとに作られたツールの WordNet::Similarity を用いる。

WordNet は英語の同義語や類義語などを分類し体系付けた概念辞書(シソーラス)である。WordNet をもとに作られ

た WordNet::Similarity において pathLength 法を用いることにより、2 単語間の類似度を測ることができる。WordNet の各単語には上位語となる単語が木構造として体系づけられている。

2 単語間の類似度を測る場合は、それぞれの単語の上位単語をたどり共通する単語を求める。2 単語がたどった全単語数 (パス数 + 1) の逆数が 2 単語間の類似度と定義されている。単語 w_1, w_2 間の類似度 $wSim(w_1, w_2)$ の定義を式(5)に示す。 $wSim(w_1, w_2)$ が大きいほど単語 w_1, w_2 間の類似度は高くなる。また、例として図 5 におけるクラス「フロア表示」と「画面表示」で使われているフロア (floor) と画面 (screen) の木構造を図 7 に示す。式(5)より、フロア (floor) と画面 (screen) の類似度 $wSim(floor, screen) = \frac{1}{3+1} = 0.25$ となる。

$$wSim(w_1, w_2) = \frac{1}{n+1} \quad (5)$$

※ n : 単語 w_1, w_2 間のパス数

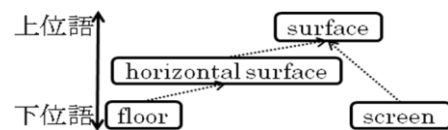


図 7.floor と screen の木構造

概念的に類似した構造を抽出するため、すべてのモデルのクラス名間の類似度を測定する。クラス名 cn_{xa}, cn_{yb} 間の類似度 $cSim(cn_{xa}, cn_{yb})$ の定義を式(6)に示す。

$$cSim(cn_{xa}, cn_{yb}) = \frac{\sum_{j=1}^{f_x} \sum_{k=1}^{f_y} Max(wSim(cw_{xaj}, cw_{ybk}))}{Max(f_x, f_y)} \quad (6)$$

※ $f_x, f_y: cn_{xa}, cn_{yb}$ の要素 cw_{xaj}, cw_{ybk} の数

※ 1つの単語は1度しか使わない

クラス名 cn_{xa}, cn_{yb} 間の類似度は、 cn_{xa}, cn_{yb} を形態素解析して $wSim(cw_{xaj}, cw_{ybk})$ を測定し、単語間の類似度が最も高い組み合わせを特定する。特定された単語を除いて同様の計算を単語数が少ないクラス名の単語数分繰り返し和を求め、単語数の多いクラス名の単語数で割る。 $cSim(cn_{xa}, cn_{yb})$ が大きいほどクラス間の類似度は高くなる。和 \sum は、1つの単語が複数の単語と対応付かないようにするために、単語数の少ないクラス名の単語数分繰り返す。単語数の多いクラス名の単語数分で割るのは、対応付かなかった単語の分だけ類似度が低くなると考えたためである。

クラス名間の類似度の測定として「フロア表示(floor-display)」と「画面表示(screen-display)」の測定例を、図8に示す。図8から、 $wSim(display, display) = 1.00$ が最大になっている。それぞれのクラス名の display を除くと、最大の類似度は $wSim(floor, screen) = 0.25$ である。 $wSim(display, display)$ と $wSim(floor, screen)$ の和を最大単語数2で割ることで、クラス名「フロア表示(floor-display)」と「画面表示(screen-display)」の類似度を定義する。図8から、

$cSim(floor - display, screen - display) = \frac{1.00 + 0.25}{2} \approx 0.63$ となる。

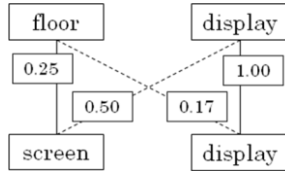


図8.クラス名間の類似度の測定例

3.4 類似した構造の抽出

入力した各モデルから1つずつクラスを選択したすべての組み合わせについて考える。入力となるクラスの集合 C_1, C_2, \dots, C_i における、すべての組み合わせの集合 O の定義を式(7)に示す。引数 i は入力する資産の数を表す。 $|A|$ は集合 A の要素数を示す。

$$O = \{o_1, o_2, \dots, o_i\} \subseteq C_1 \times C_2 \times \dots \times C_i \quad (7)$$

$$t = \prod_{k=1}^i |C_k|$$

すべての組み合わせである、 O の各要素 o_i に対して、クラス名間の類似度の平均を求める。 o_i におけるクラス名間の類似度の平均 $Av(o_i)$ の定義を式(8)に示す。 $Av(o_i)$ が最

も高い組み合わせを類似したクラスの組み合わせとして特定し、 o_i に含まれるクラスを除いて同様の測定を繰り返す。

$$Av(o_i) = \frac{\sum_{a=1}^{i-1} \sum_{b=2}^i (cSim(c_a, c_b))}{n(n-1) \times \frac{1}{2}} \quad (8)$$

$$c_a, c_b \in o_i$$

$$a \neq b$$

※クラス c_a, c_b は全体を通して1度しか使わない

$Av(o_i)$ で特定されたすべての各組み合わせ o_i において、2つずつの組み合わせに対して、各モデルで対応したクラス同士の関連を調べる。すべてのモデルにおいて o_i に対応したクラス同士が関連をもっている場合に、概念的に類似した構造として抽出する。概念的に類似した構造として抽出する場合を以下に示す。

$$o_j = \{c_{xj}, c_{yj}, \dots, c_{ij}\}$$

$$o_k = \{c_{xk}, c_{yk}, \dots, c_{ik}\}$$

$$r_{x(j-k)} = (c_{xj}, c_{xk}), r_{y(j-k)} = (c_{yj}, c_{yk}), \dots, r_{i(j-k)} = (c_{ij}, c_{ik})$$

※ j_x, j_y, \dots, j_i と k_x, k_y, \dots, k_i は任意の値

※ $r_{x(j-k)}, r_{y(j-k)}, \dots, r_{i(j-k)}$ がすべて存在する場合に、 o_j, o_k と $r_l = (r_{x(j-k)}, r_{y(j-k)}, \dots, r_{i(j-k)})$ を類似した構造として抽出する

抽出された o_j, o_k と r_l の集合を M_p と定義し、式(9)に示す。

$$M_p = (C_p, R_p) \quad (9)$$

$$C_p = \{o_1, o_2, \dots, o_f\}$$

$$R_p = \{r_1, r_2, \dots, r_g\} \subseteq C_f \times C_f$$

各 o_j に対して、抽象化したクラス名を付け、パターン構成クラス名と定義する。パターン構成クラス名は、WordNet::Similarity を用いて組み合わせ内で対応づいた単語同士の上位単語を求め、最初に共通した単語を取り出す。図7における、「floor」と「screen」の最初に共通した単語は「surface」である。単語数の少ないクラス名の単語数分取り出し、日本語に翻訳した語をパターン構成クラス名とする。パターン構成クラス名の例として、図5における「フロア表示」と「画面表示」のパターン構成クラス名は「ディスプレイ」となっている。

3.5 具象度の測定

3.4で抽出した M_p について具象度を定義する。具象度を求めることにより、抽出されたパターンの特化・汎用の度合を判断できる。

類似した構造の具象度合は、各クラス名間の類似度 $Av(O_i)$ と、関連 $r_{x(j-k)}, r_{y(j-k)}, \dots, r_{i(j-k)}$ の類似度 $rSim(r_{x(j-k)}, r_{y(j-k)}, \dots, r_{i(j-k)})$ によるものと考えられる。本手法においては、クラス図の関連を木構造として構築した。クラス図の関連の木構造を図9に示す。

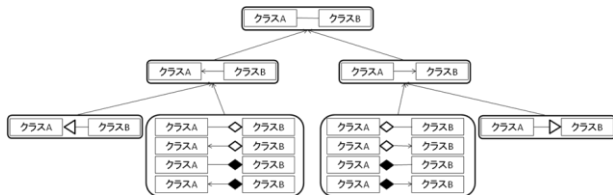


図9.クラス図の関連の木構造

図9の木構造は、種類の類似性よりも、方向の類似性により類似度が大きいとして構築した。これは、関連の方向がクラス間の関連において、強い意味合いを持つと考えられるからである。本手法では関連 $r_{x(j-k)}, r_{y(j-k)}, \dots, r_{i(j-k)}$ の類似度 $rSim(r_{x(j-k)}, r_{y(j-k)}, \dots, r_{i(j-k)})$ を、図9において pathLength 法を適用することにより定義した。類似度 $rSim(r_{x(j-k)}, r_{y(j-k)}, \dots, r_{i(j-k)})$ は、正規化した数値で表現され、似た関連を持つほど値が大きくなる。 M_p における具象度 $E(M_p)$ の定義を式(10)に示す。

$$E(M_p) = \frac{\sum_{f=1}^{|C_p|} Av(o_f) \sum_{g=1}^{|R_p|} rSim(r_{x(j-k)}, r_{y(j-k)}, \dots, r_{i(j-k)})}{|C_p| \cdot |R_p|} \quad (10)$$

具象度 $E(M_p)$ は、クラス名間の類似度 $Av(O_i)$ と、関連 $r_{x(j-k)}, r_{y(j-k)}, \dots, r_{i(j-k)}$ の類似度

$rSim(r_{x(j-k)}, r_{y(j-k)}, \dots, r_{i(j-k)})$ の、それぞれの平均を乗算したものとして定義した。

3.6 関連のある要求の順位付け

類似した構造と入力した各要求との対応を調べる。シナリオ S_{xu} と類似した構造との関連の強さ $sr(S_{xu})$ の定義を式(11)に示す。類似した構造に利用されるクラス C_x のクラス名 cn_{xf} に含まれる単語 cw_{xfh} が、シナリオ S_{xu} に含まれる割合を $sr(S_{xu})$ とする。異なるドメインの記述が混在してしまうため、類似した構造と要求の関連の強さを定量化して付与することで、パターンを適用する背景の理解を支援できる。

$$sr(S_{xu}) = \frac{|W_{(p-x)} \cap K(S_{xu})|}{|K(S_{xu})|} \quad (11)$$

※ $W_{(p-x)}: C_p$ の要素と対応する C_x の要素のクラス名 cn_{xf} に含まれる単語の集合

4. 実験・評価

本手法の実験として、同一ドメインにおける対応付いた要求とモデルの資産からの特化型パターンの抽出と、異なるドメインにおける対応付いた要求とモデルの資産からの汎用型パターンの抽出を行った。

また、抽出したパターンを適用したモデルを作成することで、パターンの抽象度や要求の必要性について評価した。

また、ET ロボコン 2009 に出場し、モデル資産のみから、本手法を用いて抽出したパターンをモデルに適用した。

本手法においては、クラス名や要求記述を単語に分割するため、形態素解析として Sen を用いる[7]。また、WordNet::Similarity は英語のみに対応しているため、クラス名を日本語から英語に翻訳する。翻訳は web 上の Google 翻訳を用いる[8]。

4.1 同一ドメインからの特化型パターンの抽出

web に公開されている、エレベータシステムの対応した要求とモデルの組 Set_X, Set_Y [9][10] を本手法の入力としてパターン抽出実験を行った。使用した要求とモデルの組 D_X, M_X, D_Y, M_Y を図10, 11, 12, 13に示す。また、抽出したパターンを図14, 15に示し「エレベータパターン」とする。図15における<要求 D_X >以下の記述が要求 D_X において抽出したパターンと関連のあるシナリオ S_{xu} であり、[s]がそのシナリオ S_{xu} と抽出したパターンとの関連の強さ $sr(S_{xu})$ である。

図14, 15からエレベータシステムに関するモデルとそれに対応した要求の記述が表わされているため、エレベータシステムに関するアナリシスパターンが抽出できたと考えられる。また、「エレベータパターン」の構造は具象度が0.45となり「コントローラパターン」(後述)の具象度0.13よりも高い。

web に公開されている、エレベータシステムに関する要求記述 D_α [11]について、「エレベータパターン」を定性的に適用した。要求記述 D_α を図16に、適用して得られたモデルを図17に示す。

図17における「エレベータ」「ドア」「ボタン」について「エレベータパターン」が適用されている。要求 D_α に沿って、図17のモデルが適切に作成されている。図17のモデル作成において「エレベータパターン」のパターン構成クラス名と同じクラス名を付けることができた。「エレベータパターン」を適用することによって、要求 D_α のみからのモデル作成に比べてクラス名とモデルの構造が再

利用できるため、モデル作成にかかるコストが削減されていることが分かる。

- ・開ボタンを押すと、エレベータのドアを開く。
- ・フロアにある、上ボタン、下ボタン、を押すと、最適なエレベータがそのフロアまで移動し、停止するようにスケジューリングされる。
- ・開ボタンを押すと、エレベータのドアを開く。
- ・閉ボタンを押すと、エレベータのドアを閉じる。
- ・...

図 10.要求 D_x [9]

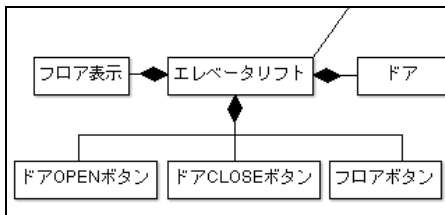


図 11.モデル M_x [9]

- ・M階建のビルのN基のO社製エレベータを、システムで制御する。
- ・それぞれのエレベータには一組のボタンがついていて、エレベータ内には、ドアを開けるボタンと閉じるボタンとがある。
- ・...

図 12.要求 D_y [10]

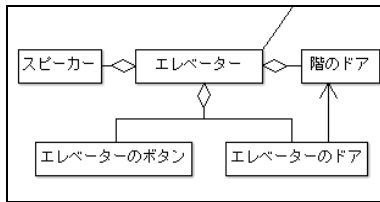


図 13.モデル M_y [10]

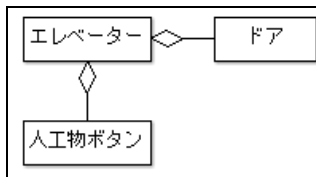


図 14.エレベータパターンのモデル

- <要求 D_x >
 [1.00]開ボタンを押すと、エレベータのドアを閉じる。
 [0.75]開ボタンを押すと、エレベータのドアを開く。
- <要求 D_y >
 [0.60]エレベータ内には、ドアを開けるボタンと閉じるボタンとがある。

図 15.エレベータパターンの要求

- ・エレベータの中にいる利用者は、移動すべき目的階を選ぶためにエレベータボタンを押す。
- ・利用者はエレベータボタンを押す。
- ・エレベータボタンセンサは、利用者が行きたい目的階を識別しつつ、システムにエレベータボタン要求を送る。
- ・新しい要求が行き先階リストに追加される。
- ・...

図 16.要求 D_a [11]

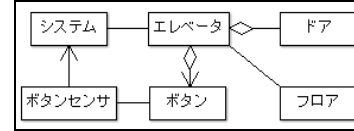


図 17.要求 D_a にエレベータパターンを適用した結果

4.2 異なるドメインからの汎用型パターンの抽出

飛行船システムの対応した要求とモデルの組 Set_z [12] と資産 Set_x [9] を本手法の入力としてパターン抽出実験を行った。飛行船システムの D_z, M_z を図 18, 19 に示す。また、抽出したパターンを図 20, 21 に示し「コントローラパターン」とする。

図 20, 21 から組み込みソフトウェアに汎用的なモデルとそれに対応した要求の記述が表わされているため、組み込みソフトウェアに関する汎用的なアナリシスパターンが抽出できたと考えられる。また、「コントローラパターン」の構造は具象度が 0.13 となり「エレベータパターン」の具象度 0.45 よりも低い。

web に公開されている、自動販売機システムに関する要求記述 D_β [13] について、「コントローラパターン」を定性的に適用した。要求記述 D_β を図 22 に、適用して得られたモデルを図 23 に示す。

図 23 において、「自動販売機コントローラ」「金銭返却部」「商品排出部」「金額表示ディスプレイ」「判定」「返却」「排出」について「コントローラパターン」が適用されている。要求 D_β に沿って図 23 のモデルが適切に作成されている。図 23 のモデル作成において、「コントローラパターン」が「エレベータパターン」に比べて抽象度が高いため、クラス名の変更が多く必要であった。「コントローラパターン」を適用することによって、要求 D_β のみからのモデル作成に比べてクラス名とモデルの構造が再利用できるため、モデル作成にかかるコストが削減されていることが分かる。

- ・自律飛行を開始する。
- ・タイマを起動させる。
- ・飛行制御を行う。
- ・コースに戻れば、情報を受け取り、保存する。
- ・コースデータを読み取る。
- ・コースデータを入力する。
- ・センサで受けた情報から現在の飛行船の状況を画面表示する。
- ・...

図 18.要求 D_z [12]

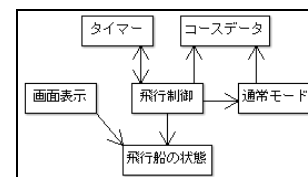


図 19.モデル M_z [12]

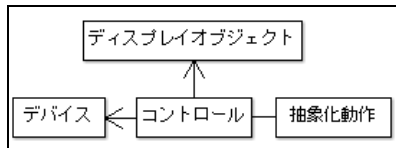


図 20.コントローラパターンのモデル

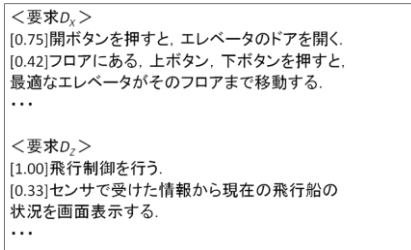


図 21.コントローラパターンの要求

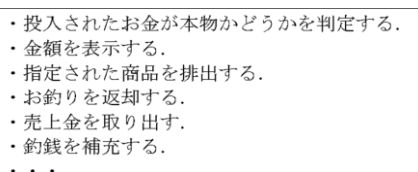


図 22.要求 D_{β} [13]

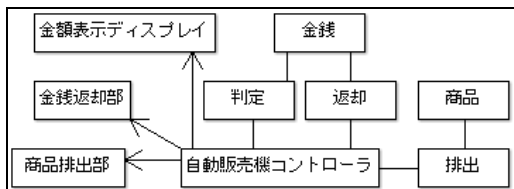


図 23.要求 D_{β} にコントローラパターンを適用した結果

4.3 モデル作成によるパターン評価

本手法において、抽出したパターンを適用したモデル作成を行い、パターンの評価を行った。この実験の評価基準を挙げる。

- 抽出したパターンを適用したモデルを作成することで、要求の必要性を評価する
- 各モデル作成から、再利用率と具象度の関係を考察し、具象度の有用性を評価する

「エレベータパターン」の適用において、図 15 の要求を理解した上で、要求 D_{α} から図 17 のモデル作成を行った。

「エレベータパターン」の要求と要求 D_{α} が同じドメインであることから、パターンの要求が容易に理解できた。特化型パターンの場合は、同一ドメインから要求を集めるため、要求の定量化を参考にする必要があまりないとわかる。

「コントローラパターン」の適用において、図 21 の要求を理解した上で、要求 D_{β} から図 23 のモデル作成を行った。「コントローラパターン」の要求と要求 D_{β} が異なるドメインであることから、パターンの要求を適用する際、

$sr(S_{xu})$ の値が低いシナリオ S_{xu} は、「コントローラパターン」のモデルとの関連度が弱いと感じられ、適用が困難であった。汎用型パターンの場合は、異なるドメインから要求を集めるため、要求の定量化を参考にすることで、パターン適用の参考になるとわかる。

モデル作成コストの指標として、パターン内のクラス名と関連において適用時に変更せずに利用した回数と、すべてのクラス名と関連の数との割合を再利用率として求める。再利用率、適用領域の広さ、具象度の関係を表 1 に示す。適用領域の広さは各実験において入力としたモデルと要求の資産のドメインから定性的に判定した。

表 1 から「エレベータパターン」は適用領域が狭いがモデルを作成する際の再利用率が高いため、特化型のパターンであると分かる。「コントローラパターン」は、適用領域が広いがモデルを作成する際の再利用率が低く、汎用型のパターンであると分かる。具象度については「エレベータパターン」が「コントローラパターン」に比べて高くなっている。以上から具象度を利用することにより、扱うドメインにそのパターンが適用可能か判定できると考えられ、具象度の正当性が得られる。システムの利用者は、具象度を利用することによって抽出されたパターンが特化型か汎用型か判断でき、扱うドメインにそのパターンが適用可能か判定できる。

本手法を用いることによって、特化型パターンだけでなく汎用型のパターンも抽出することができる。適切な具象度によって、効率的な再利用が可能なアナリシスパターンが自動抽出され、抽出にかかるコストの削減が実現できた。

表 1.再利用率・適用領域・具象度の関係

適用パターン	再利用率	ドメイン	具象度
エレベータパターン	0.80	狭い	0.45
コントローラパターン	0.43	広い	0.13

4.4 被験者のモデル作成によるパターン評価

本手法において、被験者によるパターン評価を行った。被験者は、UML の経験が約 3 年の学生 15 名である。被験者に要求記述を渡し、クラス図を作成させた。それぞれの学生に対して、パターンの構造と要求を提示した場合、パターンの構造のみを提示した場合、何も提示しなかった場合において、実験を行った。この実験の評価基準を挙げる。

- 各場合においての、クラス図作成にかかるコスト（時間）の違いを計測し評価する
- 各場合において、作成されたクラス図の品質を確認することで、パターンの有用性を評価する

この実験は、特化型パターンである、図 14,15 の「エレベータパターン」を対象とする。被験者に提示した要求記

述は図 16 の要求 D_{α} である。また、提示したパターン、クラス図作成時間、クラス図適合率の関係を図 24 に示す。

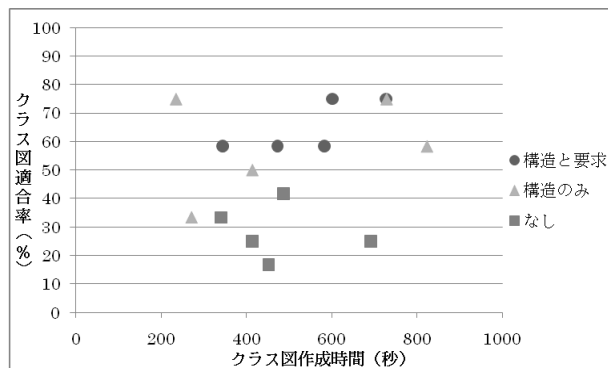


図 24.提示したパターン・クラス図作成時間・クラス図適合率の関係

クラス図作成にかかるコストの違いについて、パターンの構造と要求を提示した場合、パターンの構造のみを提示した場合、何も提示しなかった場合のクラス図作成時間(秒)の平均はそれぞれ、545.9, 494.1, 477.1 となった。また、最大値はそれぞれ、727.4, 822.9, 692.1, 最小値はそれぞれ、343.9, 234.7, 340.4 となった。クラス図作成時間は、各場合において、大きな違いがないことがわかる。これは、パターンの構造や要求を提示した場合、それらを理解するために時間を要したためだと考えられる。

パターンの構造と要求を提示した場合と構造のみを提示した場合を比較すると、パターンの構造のみを提示した被験者によっては、クラス図作成にかかるコストが大きくなっている。これは、パターンの要求がないため、構造の理解に時間を要したためだと考えられる。よって、モデルのみを提示しても、要求との関連が分からず、効率的な再利用が可能なパターンとなりえないため、関連している要求を定量化して提示する必要がある。

作成されたクラス図の品質について、各被験者が作成したクラス図と図 17 を比較し、クラス名と関連が一致している割合をクラス図適合率 (%) として求める。パターンの構造と要求を提示した場合、パターンの構造のみを提示した場合、何も提示しなかった場合のクラス図適合率の平均は、それぞれ 65.0, 58.3, 28.3 となった。また、最大値はそれぞれ、75.0, 75.0, 41.7, 最小値はそれぞれ、58.3, 33.3, 16.7 となった。

構造と要求を提示した場合は、「エレベータパターン」の要求を理解した上で、「エレベータパターン」のモデルを適用し、図 17 のクラス図と類似したモデルが作成されていた。パターンの構造のみを提示した場合は、「エレベータパターン」のモデルが提示された意図が理解できていなかったためか、不適切な適用をしている被験者がいた。よって、モデルのみを提示しても、要求との関連が分からず、効率的な再利用が可能なパターンとなりえないため、関連している要求を定量化して提示する必要がある。

何も提示しなかった場合は、クラスの適切な抽出が行われていないことが目立った。特に、図 16 の要求 D_{α} において重要な、「ボタン」や「ドア」のクラスが抽出されておらず、適切なモデルが作成されていなかった。また、クラス名の付け方が不適切であった。さらに、関連の方向や種類が適切な表現になっておらず、理解性の低いモデルが作成されていた。よって、本手法を用いることで、効率的な再利用が可能なアナリシスパターンを抽出できる。

4.5 モデル資産のみからのパターン抽出

ET ロボコン 2009 の東京大会に参戦し、本手法の適用を試みた。ET ロボコン (Embedded Technology Software Design Robot Contest) とは、組込みソフトウェア分野における技術教育をテーマに、レゴ、マインドストームの車で決められたコースを自律走行する競技である[14]。同一のハードウェアのもと、走行だけでなく UML 等で分析・設計したソフトウェアの技術を競う。コンテストは競技部門とモデル部門を併せたものが総合結果となる。

我々は、過去の ET ロボコンにおいて、優秀な成績を収めた 3 つのチームのモデル資産からパターンを抽出し、そのパターンを適用したモデルを提出した。入力として用いた各モデルはクラスが 20 以上ずつ存在した。それぞれの資産には、要求に関する記述が存在しなかったため、モデルのみからのパターン抽出を行った。抽出したパターンを「ロボコンパターン」とし、図 25 に示す。

図 25 から、ロボコンに関するモデルが表わされている。ET ロボコンのように、極めて限られたドメインに関する要求から作られたモデルには、背景に同様の要求が存在するため、モデルのみからでもパターンが抽出でき、効率的な再利用が可能であると考えられる。

ET ロボコン 2009 の東京大会の総合結果は 27 チーム中 9 位であったが、競技部門が 14 位であったことから、モデル部門においては、高い評価であったと考えられる。このことから、本手法の有用性がうかがえる。

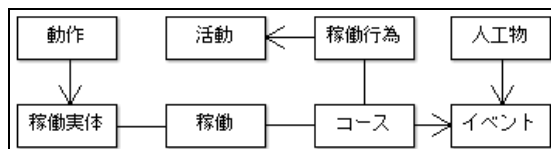


図 25.ロボコンパターンのモデル

5. 関連研究

Tonella らは、既存のオブジェクト指向プログラムソースコードからオブジェクト指向デザインパターンを発見する手法を提案している[15]。継承や関連、メソッドの呼び出し等の関係を共有するクラス群を識別するため、コンセプト分析(Concept Analysis)手法を用いている。この手法においては、モデルの構造のみを解析することによってデザインパターンを抽出している。対して本手法は、モデルと要求の対応関係を用いることで、抽出されたパターンに至る

背景を知ることができ、効率的な再利用が可能なアナリシスパターンを抽出することが可能である。

Lappe らは、ケーススタディの集合から共通部分を発見し一般化することでパターンを得るワークショップ型のパターン抽出手法を提案している[4]。また、久保らは、ソフトウェア要求獲得に関する熟練者に対してインタビューを行い、Lappe らの手法の変形を用いて互いに深く関連する一連のパターンを得ることによって、その手法の有効性を示している[5]。対して本手法は、アナリシスパターンを自動抽出することによって、抽出に人手を介すコストを削減することが可能である。

John らは、ドメインにおけるコア資産を発見する技術として CaVE(Commonality and Variability Extraction Approach)を提案している[16]。これは、ユーザドキュメントや開発ドキュメントに特化して、ドメインのバリエーションポイントとバリエーションを発見する方法である。この手法においてはドメインの可変性を抽出するのに対し、本手法はシステムを自動化してアナリシスパターンを抽出する。

6. おわりに

オブジェクト指向のソフトウェア開発における分析段階においては、アナリシスパターンを用いることで、モデルの作成が容易になる。しかし、アナリシスパターンの抽出は、ワークショップなど人手を介したものであり、コストがかかる。さらに、抽出元の資産の構成に応じて、特化・汎用型パターンをそれぞれ抽出できる必要があり、抽出したパターンの特化・汎用の割合を示すことも望まれる。

我々は、対応付いた要求とモデルの集合から、共通部分を特定してアナリシスパターンを自動抽出する手法を提案し、その有効性を検証した。対応した要求とモデルの資産を構造および語の類似性に基づいて解析することにより、具象度を利用したアナリシスパターンの自動抽出が確認できた。本手法を用いることで、コストをかけることなく、新しく効率的な再利用が可能なアナリシスパターンを抽出できる。

今後の課題として、本手法においては「属性」「操作」「ルール名」「多重度」は考慮せずクラス名のみを解析している。これらを扱うことで、より効率的な再利用が可能なアナリシスパターンの抽出が可能になると考えられる。

また、汎用型パターンの抽出において、抽出した構造と要求の抽象度に大きな違いがある。言語解析などを利用することで、より理解度の高い汎用型アナリシスパターンの抽出が可能になると考えられる。

さらに、実験において、抽出された2つのパターンの具象度を比較することで、特化型と汎用型のアナリシスパターンとした。今後さらに多くの実験を行うことで、具象度に閾値を設け、より適切に特化型と汎用型のアナリシスパターンを判別したい。

7. 参考文献

- [1] 鷲崎弘宜, 太田健一郎. 2005. よくわかるソフトウェア・パターン これから学ぶ人のデザインパターン入門. 日経ソフトウェア. 日経 BP 社. 2005年6月号
- [2] M.Fowler. 1996. *Analysis Patterns : REUSABLE OBJECT MODELS*. Pearson Education
- [3] パターンワーキンググループ. 2005. ソフトウェアパターン入門—基礎から応用へ—. ソフト・リサーチ・センター
- [4] L.Hagge, K.Lappe. 2005. *Sharing Requirements Engineering Experience Using Patterns..* IEEE Software. Vol.22. No.1
- [5] 久保淳人, 鷲崎弘宜, 深澤良彰. 2005. パターンマイニングによるソフトウェア要求の獲得知識の記述. 第12回ソフトウェア工学の基礎ワークショップ. FOSE2005in 仙台
- [6] Princeton University. *WordNet*. <http://wordnet.princeton.edu/>
- [7] 形態素解析システム Sen. <http://chasen.naist.jp/hiki/ChaSen/>
- [8] Google 翻訳. <http://translate.google.co.jp/>
- [9] 株式会社オーガス総研. *UML とオブジェクト指向分析・設計が開発リスクを軽減する*. <http://www.ogis-ri.co.jp/otc/hiroba/technical/DDJ/index.html>
- [10] 森武俊. *知能ソフトウェア(2005年度)過去のエレベータのページ*. 東京大学. <http://www.ics.t.u-tokyo.ac.jp/tmori/lectures/is/2005/elevator/elevator.html>
- [11] 牛尾遼平. 2007. *定理証明技法を用いたユースケースの追加支援システムの研究*. 北陸先端科学技術大学院大学修士論文
- [12] 和歌山大学, アーツテックラボ, 徳島大学. 2008. *MDD ロボットチャレンジ 2008 モデル審査用資料*. チーム WEST. 情報処理学会. MDD2008(Model Driven Development)
- [13] 吉田誠一. *良いユースケースを書くための発想法*. <http://www.aerith.net/index-j.html>
- [14] *ET ロボコン 2009*. 2009. <http://www.etrobo.jp/ETROBO2009/>
- [15] P.Tonella, G.Antoniol. 1999. *Object Oriented Design Pattern Inference*. Proc.ICSM'99(International Conference on Science and Technology of Synthetic Metals)
- [16] John,I, Doerr,J, Schmid,K. 2003. *User Documentation Based Product Line Modeling*. Fraunhofer IESE, IESE-Report No.004