

---

# 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

Automatic extracting of analysis pattern based on similarity of structure and word

野本 悠太郎\* 久保 淳人† 鷲崎 弘宜‡ 深澤 良彰§

あらまし ソフトウェア開発における分析の工程ではアナリシスパターンを利用する方法が提案されている。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。提案手法においては、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。我々の実験においては提案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た。

## 1 はじめに

オブジェクト指向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。分析工程においては、概念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概念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、優れた概念モデルを導出する構造をパターン化したものである。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]。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るためコストがかかる。

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。クラス名間の類似度を測定することで、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、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ため、パターンの具象度を定義する。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おける問題

### 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概念モデルを作成する。概念モデルとは、要求の特徴を捉え抽象化したモデルである。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開発を行う問題領域で必要となるデータや

---

\*Yutaro Nomoto, 早稲田大学 理工学術院 情報理工学研究科 情報理工専攻

†Atsuto Kubo, 国立情報学研究所

‡Hironori Washizaki, 早稲田大学 理工学術院 情報理工学研究科

§Yoshiaki Fukazawa, 早稲田大学 理工学術院 情報理工学研究科

概念を明確化し、要求の記述から概念モデルを作成する構造をパターン化する。

多くのパターンは、特定の問題領域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。汎用型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、変更に強くわかりやすい概念モデルの作成を支援する。業務領域に特化したアナリシスパターンとして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]。このパターンを再利用可能な状況や要求として、(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、が該当する。

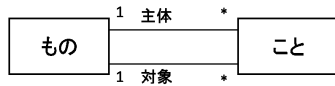


図1 「もの-こと-もの」パターンの構造 [1]

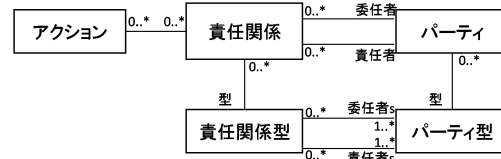


図2 Fowlerの「責任関係の知識レベルパターン」の構造 [3]

## 2.2 現状のアナリシスパターン抽出の問題点

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、モデルに自然言語が用いられることなどから、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。このため、アナリシスパターンの抽出にはコストがかかり、要求・モデル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]。

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概念モデル作成を簡略化する。そのため、概念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対象領域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。例えば、図2のみから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概念モデルの作成が困難となる。以上から概念モデル資産と要求資産を対応付けた抽出が必要と考えられる。

## 3 構造および語の類似性による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。本手法における対応付いた要求・モデル資産とは、同じ開発の元となった要求記述とUMLクラス図の組である。飛行船システムにおける対応付いた要求とモデルを図3,4に示す[6]。図3の記述が図4に抽象化されてモデリングされている。クラス図の構造およびクラス名の類似度を測定し、類似度の高い順に複数のモデル間のクラスを1対1に対応付けてパターンとして抽出する。本システムはJavaで実装され、要求記述をテキストファイル形式(.txt)に、クラス図をXML形式(.xmi)に変換して入力する。複数の対応付いた要求とモデルの組を入力として、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。モデルのみからでも抽出は行えるが、要求との関連が分か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を抽出できない。

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。本手法はそ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、共通ドメインからモデルが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抽出することを目的とする。パターンを抽出する際にはモデルを抽象化して類似性を求めてから汎用的に扱うため、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 ⇄ 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という抽象化した語で類似し、「フロア表示」と「画面表示」が概念的にディスプレイという抽象化した語で類似している。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がクラスの集合として構成されており、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。クラス名間の類似度は単語間の意味の類似度によって求められると考えられるため、単語間の類似度を計測して定量的に扱う。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。

本手法においては2単語間の類似度を測定するために、英語のシソーラスであるWordNet [7]をもとに作られたツール WordNet::Similarity を用いる。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。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、扱う問題領域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関連の強さを定量化することで、パターンを利用すべき要求記述が分かりやすくなる。本手法の全体像を図6に示し、概要を以下に記述する。3.1節においてそれぞれ詳細に説明する。

1. 頻出単語の抽出：要求記述の中で、頻出している単語を抽出する。
2. クラス名の特定：1.で抽出された単語を名前に持つクラスを特定する。
3. 概念的に類似した構造の抽出：モデルの集合から類似した構造を抽出する。
4. 具象度の測定：具象度を測定することにより3.の構造の特化・汎用の度合を示す。
5. 対応した要求の選択：3.の構造に対応した要求を、入力した各要求から選択する。
6. 要求の定量化：3.の構造と5.の各要求との関連の強さを定量化する。

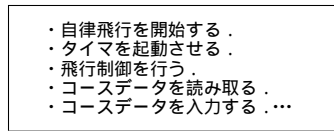


図3 要求記述の例

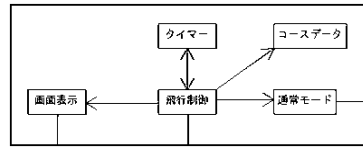


図4 モデルの例

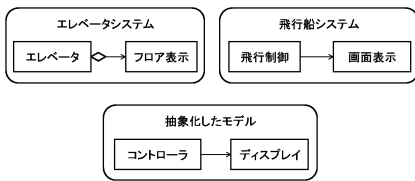


図5 モデルを抽象化した例

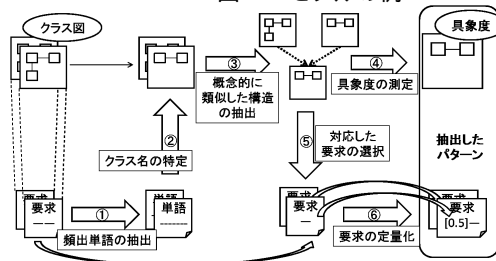


図6 提案手法の全体像

### 3.1 単語間の類似度の測定方法

WordNet は類義語などを分類し体系付けた英語の概念辞書(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が木構造として体系付けられている。WordNet::Similarity において定義されてい

る、単語  $a, b$  間の類似度  $wSim(a, b)$  を式 (1) に示す。  $wSim(a, b)$  が大きいほど単語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 (1) より  $wSim(floor, screen) = \frac{1}{3+1} = 0.25$  となる。

$$wSim(a, b) = \frac{1}{p+1} \quad (p: \text{単語 } a, b \text{ 間のパス数}) \quad (1)$$

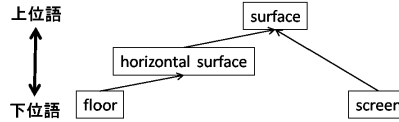


図 7 floor と screen の木構造

### 3.2 アナリシスパターンの自動抽出方法

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 [8]。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 Excite 翻訳を用いる [9]。

入力となるモデル  $M$  を式 (2) で定義する。モデル  $M$  は、クラス  $c$  の集合と各クラス間の関連  $r$  の集合の組と考えることができる。

$$M = (C, R) \quad C = \{c_1, c_2, \dots, c_m\} \quad R = \{r_1, r_2, \dots, r_n\} \subseteq C \times C \quad (2)$$

まず、入力された要求記述を形態素解析し出現頻度の高い単語群を抽出する (図 6 の 1. 頻出単語の抽出)。抽出した単語がクラス名に含まれているクラスのみを扱う (図 6 の 2. クラス名の特定) ことによって、要求と概念的に類似した構造の対応が取れる。特定されたクラスについて、すべてのモデル資産から概念的に類似した構造を抽出する (図 6 の 3. 概念的に類似した構造の抽出)。

#### (1) クラス名間の類似度の測定

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  $cn_a, cn_b$  間の類似度  $cSim(cn_a, cn_b)$  の定義を式 (3) に示す。

$$cSim(cn_a, cn_b) = \frac{\sum_{a_i \in cn_a, b_j \in cn_b}^n Max(wSim(a_i, b_j))}{m} \quad (m \geq n) \quad (3)$$

( $a_i, b_j$ : クラス名  $cn_a, cn_b$  を構成する単語 (単語  $a_i, b_j$  は 1 度しか使わない))

クラス名  $cn_a, cn_b$  間の類似度は、 $cn_a$  と  $cn_b$  を形態素解析して  $wSim(a_i, b_j)$  を測定し、単語間の類似度が最も高い組み合わせ  $Max(wSim(a_i, b_j))$  を算出する。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  $n$  回繰り返して和を求め、単語数の多いクラス名の単語数  $m$  で割る。 $cSim(cn_a, cn_b)$  が大きいほどクラス間の類似度は高くなる。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、表 1、図 8 に示す。

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

表 1 クラス名間の類似度の測定例 1

$wSim$	floor	display
screen	0.25	0.50
display	0.17	1.00

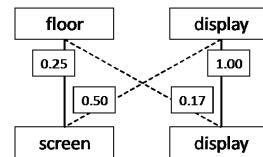


図 8 クラス名間の類似度の測定例 2

#### (2) 各モデル間のすべての組み合わせ

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。入力となるモデル  $M_1, M_2, \dots, M_n$  の組み合わせ  $x_t$  の定義を式 (4) に示す。図

5 の場合  $x_1 = \{ \text{エレベータ, 飛行制御} \}$ ,  $x_2 = \{ \text{エレベータ, 画面表示} \}$ ,  $x_3 = \{ \text{フロア表示, 飛行制御} \}$ ,  $x_4 = \{ \text{フロア表示, 画面表示} \}$  となる .

$$x_t = \{c_{1p}, c_{2q}, \dots, c_{nr}\} \quad (c_{is} \in C_i, \quad |x_t| = \prod_{k=1}^n |C_k|) \quad (4)$$

各  $x_t$  についてクラス名間の類似度の平均を求める .  $x_t$  におけるクラス名間の類似度の平均  $Av(x_t)$  の定義を式 (5) に示す .  $Av(x_t)$  が最も高い組み合わせを類似したクラスとして特定し , その  $\{c_1, c_2, \dots, c_n\} \in x_t$  を除いて同様の測定を繰り返す . 図 5 の場合  $Av(x_4)$  が最大とすると  $x_4 = \{ \text{フロア表示, 画面表示} \}$  であるため「フロア表示」と「画面表示」が含まれる  $x_2, x_3$  は除かれる .

$$Av(x_t) = \frac{\sum_{i=1}^{n-1} \sum_{j=2}^n (cSim(c_i, c_j))}{n(n-1) \times \frac{1}{2}} \quad (c_i, c_j \in x_t, \quad i \neq j) \quad (5)$$

(クラス  $c_i, c_j$  は 1 度しか使わない)

### (3) パターン構成クラス名の決定

(2) で特定された各  $x_t$  に対して , それらを抽象化したクラス名を付けパターン構成クラス名  $X_t$  と定義する . WordNet::Similarity を用いて組み合わせ内で対応付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し , 日本語に翻訳した語を  $X_t$  とする . 図 7 における「floor」と「screen」の最初に共通した単語は「surface」, 図 5 における「フロア表示」と「画面表示」の  $X_t$  は「ディスプレイ」である .

### (4) 概念的に類似した構造の抽出

(3) で決定された  $X_t$  の 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 . 各モデルにおいて  $X_t$  に含まれるクラス同士が関連を持っている場合に類似した構造として抽出する . 類似した構造として抽出する場合を以下に示す . 図 5 の場合  $X_1 = \{ \text{エレベータ, 飛行制御} \}$  と  $X_4 = \{ \text{フロア表示, 画面表示} \}$  は ,  $r_1 = \{ \text{エレベータ} \leftrightarrow \text{フロア表示} \}$  と  $r_2 = \{ \text{飛行制御} \rightarrow \text{画面表示} \}$  が存在し  $X_1$  と  $X_4$  はクラス間の関連があるため , 類似した構造として抽出される .

$$X_{t_1} = \{c_{11}, c_{21}, \dots, c_{n1}\}, X_{t_2} = \{c_{12}, c_{22}, \dots, c_{n2}\}, r_i = \{c_{i1}, c_{i2}\}$$

$r_1, r_2, \dots, r_n$  がすべて存在する場合に類似した構造として抽出する

### (5) 具象度の測定方法

(4) で抽出した構造について具象度を定義する (図 6 の 4. 具象度の測定) . 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  $Av(X_t)$  と  $X_t$  間の関連の相違度  $D(X_j, X_k)$  によるものである . 相違度  $D(X_j, X_k)$  は手動で値を設け , クラス間の関連が異なるほど高くした . モデル  $M_1, M_2, \dots, M_n$  の類似した構造における具象度  $E(M_n)$  の定義を式 (6) に示す . 図 5 の場合  $D(X_1, X_4)$  の  $\leftrightarrow$  と  $\rightarrow$  の相違は 0.5 と設定した .

$$E(M_n) = \frac{\sum_{i=1}^{|X_t|} Av_i(X_t)}{|X_t|} \times \frac{\sum_{j=1}^{|X_t|-1} \sum_{k=2}^{|X_t|} 1}{\sum_{k=2}^{|X_t|} D(X_j, X_k)} \quad (6)$$

( $p$  : パターン構成クラス  $X_t$  間の関連  $R$  の数)

( $X_j$  と  $X_k$  に関連がない場合は  $D(X_j, X_k)$  を測定しない)

### (6) 類似した構造と要求の対応

類似した構造と入力した各要求との対応を調べる (図 6 の 5. 対応した要求の選択 , 6. 要求の定量化) . 各要求は句点や改行で分割し , シナリオ群として扱う . シナリオ  $s$  と類似した構造との関連の強さ  $N(s)$  の定義を式 (7) に示す . 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 . 図 3 の  $s =$  "自律飛行を開始する" と図 4 の「飛行制御」の場合 ,  $aw_s = \{ \text{自律, 飛行, 開始} \}$ ,  $w_s = \{ \text{飛行} \}$  であるので  $N(s) = \frac{1}{3} = 0.33 \dots$  となる .

$$N(s) = \frac{|w_s|}{|aw_s|} \tag{7}$$

( $aw_s$  : シナリオ  $s$  に現れる全単語)

( $w_s$  : 類似した構造のクラスに含まれる単語とシナリオ  $s$  に共通する単語)

#### 4 適用実験

本手法の実験として、同一ドメインにおける要求・モデル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要求・モデル資産からの汎用型パターンの抽出を行った。入力したモデル数は2、クラス数は15-25程度である。

##### 4.1 同一ドメインからの特化型パターンの抽出

エレベータシステムの対応した要求とモデルの組 [10] [11] を本手法の入力としてパターン抽出実験を行った。要求とモデルの組を図9, 10, 11, 12に示す。抽出したパターンを図13, 14に示し「エレベータパターン」とする。図13, 14からエレベータシステムに関する概念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。

エレベータシステムに関する要求記述 [12] について「エレベータパターン」を定性的に適用した。要求記述を図15に、適用して得られたモデルを図16に示す。図16における「エレベータ」「ドア」「エレベータボタン」について「エレベータパターン」が適用されている。「エレベータパターン」を適用することによって、図15の要求記述のみからの概念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、概念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。

・入力文書: 「閉ボタンを押すと、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。...

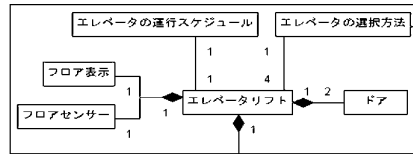


図9 要求 A [10]

図10 モデル A [10]

・M 階建のビルに 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。...

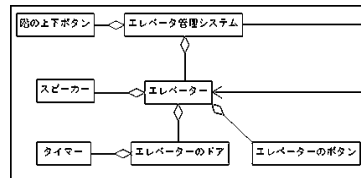
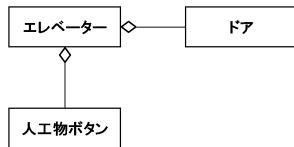


図11 要求 B [11]

図12 モデル B [11]



<要求 A >  
 [1.00] 閉ボタンを押すと、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 B >  
 [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13 エレベータパターンのモデル

図14 エレベータパターンの要求

##### 4.2 異なるドメインからの汎用型パターンの抽出

飛行船システムの図3, 4と図9, 10を本手法の入力としてパターン抽出実験を行った。抽出したパターンを図17, 18に示し「コントローラパターン」とする。図17, 18から組み込みソフトウェアに汎用的な対応したモデルと要求の記述が示されているため、汎用型アナリシスパターンが抽出できたと考えられる。

自動販売機システムに関する要求記述 [13] について「コントローラパターン」を定性的に適用した。要求記述を図19に、適用して得られたモデルを図20に示す。

・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。...

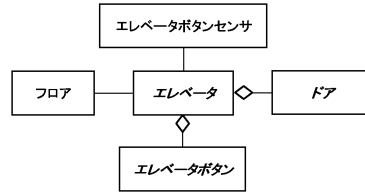


図 15 エレベータシステムに関する要求記述 [12]

図 16 エレベータパターンを適用した結果

図 20 におけるすべてのクラスについて「コントローラパターン」が適用されている。「コントローラパターン」を適用することによって、図 19 の要求記述のみからの概念モデル作成に比べてモデルの構造が再利用できるため、概念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。

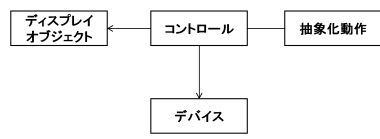


図 17 コントローラパターンのモデル

<要求 A >  
 [0.75] 開ボタンを押すと、エレベータのドアを開く。  
 [0.42] 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する。...  
 <要求 C >  
 [1.00] 飛行制御を行う。  
 [0.33] センサで受けた情報から飛行船の状況を画面表示する。...

図 18 コントローラパターンの要求

・投入されたお金が本物かどうかを判定する。  
 ・金額を表示する。  
 ・指定された商品を排出する。  
 ・お釣りを返却する。  
 ・売上金を取り出す。  
 ・釣銭を補充する。  
 ・商品を値上げする。...

図 19 自動販売機システムに関する要求記述 [13]



図 20 コントローラパターンを適用した結果

### 4.3 評価・考察

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図 16, 20 の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（入力したモデル資産と要求資産のドメインから定性的に判定）、具象度の関係を表 2 に示す。表 2 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」の 0.45 が「コントローラパターン」の 0.13 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、扱う問題領域に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。

本手法を用いることによって、特化型のアナリシパターンだけでなく汎用型のアナリシパターンも抽出することができた。適切な具象度によって、効率的な再利用が可能でアナリシ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る。

表 2 再利用率・適用領域・具象度の関係

適用パターン	再利用率	問題領域	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

## 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 [14]。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た [4]。ケーススタディとは、ある具体的な事例について詳しく分析・研究して、その背後にある原理や法則性などを究明し、一般的な法則・理論を発見しようとする方法である。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、抽出にかかるコストを削減することが可能である。

## 6 おわりに

アナリシスパターンの抽出は人手を介したものでありコストがかかる。我々は、対応付けられ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・モデル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、ロール名、多重度は考慮せずに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、パターン構成クラス名は手動で付けているため完全に自動化したい。

## 参考文献

- [1] 鷲崎弘宜・太田健一郎, "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, 2005年6月号, 2005.5
- [2] M.Fowler, "Analysis Patterns:REUSABLE OBJECT MODELS", Pearson Education,1996
- [3] パターンワーキンググループ, "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, 2005
- [4] L. Hagge, K. Lappe: Sharing Requirements Engineering Experience Using Patterns., IEEE Software, Vol.22, No.1, 2005.
- [5] 久保淳人, 鷲崎弘宜, 深澤良彰, "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 FOSE 2005 in 仙台, 2005
- [6] 和歌山大学・アーツテックラボ・徳島大学, "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), 2008
- [7] Princeton University, WordNet, <http://wordnet.princeton.edu/>
- [8] 形態素解析システム Sen, <http://chasen.naist.jp/hiki/ChaSen/>
- [9] Excite 翻訳, <http://www.excite.co.jp/world/>
- [10] 株式会社オージス総研, 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>
- [11] 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>
- [12] 牛尾遼平, "修士論文 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, 2007  
<https://dspace.jaist.ac.jp/dspace/bitstream/10119/3595/3/031paper.pdf>
- [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] P.Tonella, G.Antoniol: Object Oriented Design Pattern Inference, Proc. ICSM '99(International Conference on Science and Technology of Synthetic Metals)