

D_010

設計仕様書を用いた分析クラス図の生成に関する研究

Research for Generating Analysis Level Class Diagram with Software Resources

池辺 正典† 田中 成典‡ 古田 均‡ 中村 健二十 細島 啓史†
 Masanori Ikebe Shigenori Tanaka Hitoshi Furuta Kenji Nakamura Hirofumi Hosohata

1. はじめに

近年、システム開発では、UML (Unified Modeling Language) を用いたオブジェクト指向開発が盛んに行われている。それに伴い、オブジェクト指向開発を支援するために、ユースケース分析法[1]やCRC分析法[1]などのオブジェクト指向分析を補助する手法が考案されている。しかし、これらの手法を用いるためには、システム分析の知識や経験が十分に必要[1]であり、分析クラス図の品質は、作成者の技量に依存してしまう問題がある。そのため、作成者の技量に依存せずにクラス図を生成する方法として、要求仕様書にクラスの抽出ルールを適用し、自動的にクラス図を生成する研究[2]や熟練技術者のシステム分析作業時の思考過程を共有する手法の研究[3]が行われている。しかし、前者[2]では、クラスの抽出ルールが固定であるため汎用性が低いという問題がある。また、後者[3]では、分析作業時の思考過程の情報を作成者自らが手動で作成しなければならず、作成者に多大な負担をかけてしまう問題がある。さらに、それらの情報の品質は、作成者の知識や経験に依存するという問題がある。

新規システム開発を行う場合、既存資産の再利用により、開発効率の向上が可能であると考えられる。しかし、現在、既存資産を再利用する方法は確立されておらず[4]、既存資産が有効に活用されていないという問題がある。

そこで、上記2つの問題点を解決するために、本研究では、既存資産を有効に活用し、一定の品質の分析クラス図を自動的に生成する手法の提案を行う。

2. 研究の概要

本研究では、既存の設計仕様書を利用して、分析クラス図を自動生成することを目的とする。このことにより、経験に依存せずに一定の品質を保った分析クラス図を生成する。本提案手法の処理の概念を図1に示す。

本システムは、図1に示すように、1) 既存資産の検索処理、2) 分析クラス図の生成処理の2つの処理から構成される。既存資産の検索処理は、リポジトリ内に蓄えられている設計仕様書の中から、利用者が入力したシステムと類似する情報を検索する。分析クラス図の生成処理は、類似した既存資産を使用して、利用者が入力したシステムに適合した分析クラス図を生成する。

本論文の構成は、まず、3章と4章でそれぞれの処理について順に解説する。そして、5章で本提案手法の精度を検証するために、実証実験を行う。最後に、6章で得られた成果と今後の課題についての考察を述べる。

3. 既存資産の検索処理

既存資産の検索処理は、利用者が入力した要求仕様書とユースケース記述からリポジトリ内に蓄えられている類似システムを検索し、その類似システムの設計仕様書を抽出する。この処理は、類似システムの検索機能と類似ユースケースの検索機能の2つの機能により構成される。以下にこれらの機能について解説する。

3.1 類似システムの検索機能

類似システムの検索機能は、利用者が入力した要求仕様書とユースケース記述を用いて、リポジトリ内から類似システムを検索する。リポジトリには、既存の設計仕様書として、要求仕様書、ユースケース記述、分析クラス図、分析シーケンス図が蓄えられているものとする。また、分析クラス図と分析シーケンス図のデータ形式は、UMLモデリングツールに依存させないためにXMI (XML Metadata Interchange) とする。XMIとは、UMLモデリングデータをXML形式で表したデータ形式である。ここで、リポジトリ r に含まれるシステム s の要求仕様書、ユースケース記述をそれぞれ Drs , Urs とする。利用者が入力した要求仕様書とユースケース記述をそれぞれ D_i , U_i とする。類似システムの検索方法は、ベクトル空間モデルを使用し、要求仕様書とユースケース記述から意味的な類似度を判定して行う。ベクトル空間モデルは、特徴ベクトルの作成と類似度の判定の2つのステップにより構成される。以下に、これらのステップについて解説する。

3.1.1 システムの特徴ベクトルの作成

システムの特徴ベクトルは、対象となる文章に含まれる単語とその単語の重要度を元に作成する。まず、対象の文章 Drs , Urs , D_i , U_i に対して形態素解析を行い、文章内に含まれる名詞、動詞群を抽出する。ここで、 Drs と Urs から抽出された名詞、動詞群を $Trs = (trs_1, trs_2, \dots, trs_n)$ とし、 D_i と U_i から抽出された名詞、動詞群を $Ti = (ti_1, ti_2, \dots, ti_m)$ とする。さらに、任意の文章に含まれる単語

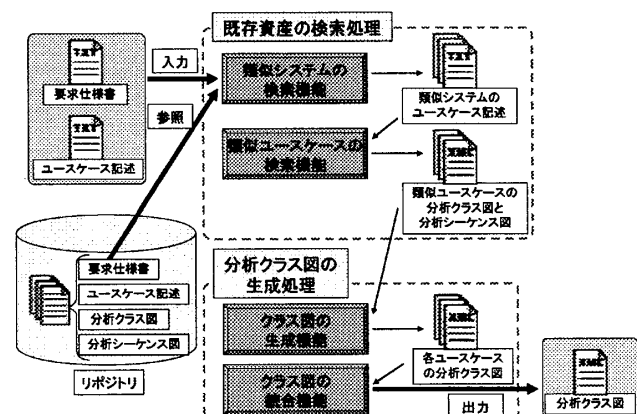


図1 本システムの概要

†関西大学大学院

‡関西大学総合情報学部

trs_n , ti_m の出現回数をそれぞれ $C(trs_n)$, $C(ti_m)$ とし、対象となる文章 Drs , Urs , Di , Ui の総数を Doc_{sum} とする。本提案手法では、形態素解析器として、Sen[5]を利用する。Senは、Java 言語で使用可能な形態素解析器である。次に、抽出した単語に対して文章における重要度を算出する。本研究では、重要度の算出に TF/IDF 法を利用する。TF/IDF 法は、文章中に同じ単語が多く出現すれば、その単語の重要度は高い、また、多くの文章に含まれる一般的な単語の重要度は低いという考えのもとで単語の重要度を算出する方法である。TF/IDF 法による、 trs_n についての重要度算出は、次式の通りである。

$$TF(trs_n) = \frac{C(trs_n)}{\sum_{k=1}^m C(trs_k)} \quad (1)$$

$$IDF(trs_n) = \log \frac{Doc_{sum}}{df(trs_n)} \quad (2)$$

$$W(trs_n) = TF(trs_n) \times IDF(trs_n) \quad (3)$$

式 (1) は、リポジトリ内の文章における任意の単語 trs_n の出現回数を算出し、その値を文章内の単語の総数で割ることによって、文章内における任意の単語の重み係数を算出する。式 (2) は、全体の文章数と任意の単語 trs_n が含まれる文章数の値により、任意の単語の重みを調整する。ここで、 df は、 Doc_{sum} 中において任意の単語 trs_n が 1 つ以上含まれる文章の個数を示す。式 (3) は、重み係数の積によって、リポジトリ内の文章における任意の単語 trs_n の重要度を算出する。ここで、 $W(trs_n)$ は、単語 trs_n の TF/IDF 値を示す。入力された文章における任意の単語 ti_m の TF/IDF 値 $W(ti_m)$ も同様の方法で算出する。そして、文章における各単語の TF/IDF 値から重要度を算出する。ここで、リポジトリ内の文章から作成される特徴ベクトルを $VTrs_n = \{W(trs_1), W(trs_2), \dots, W(trs_n)\}$ 、入力された文章から作成される特徴ベクトルを $VTi = \{W(ti_1), W(ti_2), \dots, W(ti_m)\}$ とする。

3.1.2 類似度の判定

利用者が入力した文章とリポジトリ内の文章の類似度を判定するために、利用者が入力した文章とリポジトリ内の文章との組み合わせについて、 $V(x, y)$ 次元の仮想空間に展開する。ここで利用する $V(x, y)$ は、 $VTrs_n$ と VTi を比較した場合に、式 (4) で得られる集合とする。

$$V(x, y) = (VTrs_n \cap VTi) \quad (4)$$

このため、仮想空間に展開された特徴ベクトルは、 $V(x, y)$ の要素を持つ必要がある。ここで、 $VTrs_n$ と VTi の共通の要素のみを抽出した特徴ベクトルをそれぞれ $VTrs'_1 = \{W(trs'_1), W(trs'_2), \dots, W(trs'_n)\}$ 、 $VTi' = \{W(ti'_1), W(ti'_2), \dots, W(ti'_m)\}$ とする。そして、文章間の類似度を判

定するために、補正した特徴ベクトル $VTrs'_1$, VTi' を使用して余弦値を算出する。余弦値の算出は式 (5) を用いる。式 (5) において、右辺の分子はリポジトリの文章と利用者の入力した文章との特徴ベクトルの内積を表し、分母は各特徴ベクトルの原点との距離の積である。特徴ベクトル間の余弦値は、文章の特徴ベクトル同士の比較を角度として算出するもので、0~1 の範囲で結果が得られる。そして、この余弦値が 1 に近いほど、2 つの文章は類似していることを示す。最後に、リポジトリ内の全てのシステムにおける余弦値を算出し、その余弦値が閾値以上のシステムにおけるユースケース記述を出力する。

3.2 類似ユースケースの検索機能

類似ユースケースの検索機能は、利用者が入力したユースケース記述に書かれている各ユースケースに対応する類似ユースケースを検索する。類似ユースケースの検索方法は、類似システムの検索機能と同様にベクトル空間モデルを使用する。類似度の判定方法は、類似システムの検索機能で出力されたユースケース記述と利用者が入力したユースケース記述を用いて、各ユースケースを対象として特徴ベクトルを算出する。そして、特徴ベクトル間の余弦値が、閾値以上のユースケースを全て類似ユースケースと判定する。最後に、類似ユースケースに対応する分析クラス図と分析シーケンス図をリポジトリから取得する。

4. 分析クラス図の生成処理

分析クラス図の生成処理は、既存資産の検索処理で抽出した設計仕様書を使用して、利用者が入力したシステムの分析クラス図を生成する。この処理は、クラス図の生成機能とクラス図の統合機能の 2 つの機能により構成される。以下にこれらの機能について解説する。

4.1 クラス図の生成機能

クラス図の生成機能は、利用者が入力したユースケース記述の各ユースケースに対する分析クラス図を生成する。この機能では、類似ユースケースの検索機能で出力された複数の分析クラス図と分析シーケンス図を使用する。この機能は、クラスの抽出、クラス間の関係の抽出、クラスの属性と操作の抽出の 3 つのステップにより構成される。以下に、これらのステップについて解説する。

4.1.1 クラスの抽出

クラスの抽出は、まず、各ユースケースの分析シーケンス図からオブジェクトのクラス名を全て抽出する。そして、EDR (Electronic Dictionary Research) 辞書[6]を用いてクラス名の一般化を行う。クラス名の一般化は、EDR 辞書からクラス名の 1 つ上の概念を取得し、2 つのクラス間においてクラス名または上位概念が一致した場合、その 2 つのクラスは同じ意味を表すクラスと考え、上位概念の単語でクラス名の一般化を行う。そして、それらの一般化したクラスの出現回数をクラスごとに算出する。最後に、合計回数が閾値以上のクラスを全てクラス候補として出力する。

$$V_{\cos}(VTrs'_1, VTi') = \frac{W(trs'_1) \times W(ti'_1) + \dots + W(trs'_n) \times W(ti'_n)}{\sqrt{W(trs'_1)^2 + \dots + W(trs'_n)^2} \times \sqrt{W(ti'_1)^2 + \dots + W(ti'_m)^2}} \quad (5)$$

4.1.2 クラス間の関係の抽出

クラス間の関係の抽出は、まず、クラス間の重要度を算出するために重み付けを行う。重み付けは、クラスの抽出で出力されたクラス候補を対象にする。クラス間の重み付けには、分析シーケンス図におけるメッセージの交換回数を使用する。ここで、利用者が入力した任意のユースケース u に対応する類似ユースケースの検索機能で抽出した分析シーケンス図の集合を $USD=(usd_1, usd_2, \dots, usd_n)$ 、クラスの抽出処理で抽出したクラス候補の集合を $UC=(uc_1, uc_2, \dots, uc_m)$ と定義する。また、任意の分析シーケンス図 usd_n に含まれるクラス uc_i と uc_j 間のメッセージの交換回数を $Mrel(uc_i, uc_j)$ 、 usd_n に含まれるメッセージの総数を $Msum$ と定義する。ユースケース u に対応するクラス uc_i とクラス uc_j 間の重みの値 $weight(uc_i, uc_j)$ は次式により算出する。

$$weight(uc_i, uc_j) = \sum_{k=1}^n \left\{ \frac{Mrel(uc_i, uc_j)}{Msum} \right\}_k \quad (6)$$

式 (6) は、まず、分析シーケンス図 usd_n におけるクラス uc_i と uc_j 間のメッセージの交換回数をメッセージの交換回数の総数 $Msum$ で割ることによって、 usd_n 内におけるクラス間の重みを算出する。ここでの重みづけは、2つのクラス間において、どのくらいメッセージが交換されているかという割合をもとに行う。次に、 USD に含まれる全ての分析シーケンス図において、同様にクラス間の重みを算出する。そして、それらのクラス間の重みの合計値を求める。この合計値が、ユースケース u における任意のクラス間の重み係数を示す。

式 (6) で求められたクラス間の重みをもとにクラス間の関係の抽出を行う。ここで、閾値の切り捨てによりクラス間の抽出を行った場合、抽出されるクラス間の関係に偏りが生じてしまうなどの問題が発生し、重要なクラス間の関係も切り捨てられる可能性がある。また、クラス間に関係のあるものは、全て関連付けてしまうと、クラス間の結合度が高くなってしまい保守性が劣ってしまうという問題がある。そこで、本研究では、クラス間の最適な組み合わせを抽出するために遺伝的アルゴリズム (以下、GA) を用いる。GA で用いる遺伝子は、2つのクラス間の関係とし、関係がある場合を1、関係が無い場合を0とする。制約条件としては、抽出した全てのクラスに対して1つ以上の関係を設定することとする。ここで、個体を $Ig_n=(ig_1, ig_2, \dots, ig_n)$ とし、個体 Ig_n の要素である遺伝子 ig_n をクラス uc_i とクラス uc_j 間の関係とする。また、個体 Ig_n において遺伝子 ig_n の値が1で、 ig_n によるクラス間の関係にクラス uc_m が含まれる個数を $Count(uc_m)$ 、 uc_m に含まれるクラス間の重みの和を $Wsum(uc_m)$ と定義する。評価値 EV_n は、次式の評価式により算出する。

$$EV_n = \sum_{k=1}^m \left\{ \frac{Wsum(uc_k)}{Count(uc_k)} \right\} \quad (7)$$

式 (7) は、まず、遺伝子の値を元に、各クラスにおける他のクラス間との関係数と重みの和を算出する。次に、

算出したクラス間の重みの和を他のクラス間との関連数で割ることによって、クラス1つ当たりの評価値を算出する。この評価式は、1つのクラスに関連が偏ると評価値は低くなるように設定している。そして、 UC に含まれる全てのクラスにおいて、同様に評価値を算出し、各クラスの評価値を合計する。最後に、この評価値が最大となる個体 Ig_n を最適なクラス間の関係として抽出し、そのクラス構造を出力する。

4.1.3 クラスの属性と操作の抽出

クラスの属性と操作の抽出は、クラスの抽出と同様の方法で行う。まず、クラスから属性と操作を全て抽出し、EDR 辞書を用いて属性名、操作名の一般化を行う。このとき、操作は、クラス間の関係の抽出で出力されたクラス構造に一致するように、分析シーケンス図から抽出する。そして、それらの属性と操作のそれぞれの出現回数を算出し、合計回数が閾値以上の属性と操作を採用する。最後に各ユースケースにおける分析クラス図を全て出力する。

4.2 クラス図の統合機能

クラス図の統合機能は、ユースケースごとに生成したクラス図の統合を行う。まず、クラスに設定されているステレオタイプでグループ分けを行う。リポジトリに蓄えられているクラスには、Boundray, Control, Entity の3つのステレオタイプの内1つが設定されているものとする。そして、グループ内のクラス名を比較し、クラス名が同様のものを重ねあわせクラス図の統合を行う。

5. 評価実験

本システムにより生成した分析クラス図の品質を評価するために、本システムにより生成した分析クラス図とオブジェクト指向分析の経験者が作成した分析クラス図との比較を行った。本実験では、共通問題[7]の在庫管理システムを対象とした。評価は、経験者 (a) が作成した分析クラス図と本システム (b), (c) により生成した分析クラス図における各要素の一致率により行う。評価対象とする分析クラス図ごとの要素としてクラス名、属性、操作、クラス間の関連の4つの要素を用いる。また、クラス間の関係の抽出において、GAを用いることによる有効性を確認するために次の2通りの方法でクラスの生成を行う。1つ目は、GAを適用しないで、クラス間の重みの値が閾値以下のものを切り捨てる方法 (b) である。もう1つは、GAの適用による抽出方法 (c) である。本実験では、在庫管理システムが15件、図書貸し出し管理システムが6件、オンラインショッピングシステムが6件、顧客管理システムが3件を蓄積しているリポジトリを利用した。また、本実験による閾値の設定は、ベクトル間の余弦値は0.8以上採用、クラス抽出は12未満のクラスを削除、クラス間の関係は12未満の関連を削除した。本実験による実験結果を表1、本システムが生成した分析クラス図とオブジェクト指向分析の経験者により作成された分析クラス図を図2に示す。また、図2において色のかかっている注文クラス、注文明細クラス、商品クラスは3つのクラス図において完全に一致した部分を示す。

実験結果が示すように、本システムを用いることにより、約70%の精度でクラスを抽出することができた。しかし、図2の経験者と本システムにおける分析クラス図の比較を行うと、本システムが生成した分析クラス図はクラス数が

少ないとの実験結果になった。これは本手法が、倉庫、コンテナといったシステム特有のクラスを抽出できなかったためであると考えられる。本手法では、一部のクラスは抽出できていないものの、請求、顧客、注文、注文明細、商品、在庫といった在庫管理システムに共通するクラスについてはクラス間の関係とともに正確に抽出することができた。また、属性の一致率が低いのは、図2の分析クラス図が示すように、必要なクラスが抽出されていないためと考えられる。クラス間の関係については、まず、経験者と本システム(b)の場合、一致率が38%であった。閾値による方法では、クラス間の重みのみで判断を行うため、図2の(b)が示すように必要なクラス間の関係までも切り捨てられている。一方、経験者と本システム(c)の場合、一致率が63%であった。GAによる方法では、63%となっているが、図2の(a)と(c)の比較からも(c)において抽出されたクラスについてのクラス間の関係については、正確に抽出することができた。これらのことから、GAを用いることによる有効性を確認することができた。

表1 実行結果

	経験者(a)	本システム		一致数		一致率(%)	
		(b)	(c)	(b)	(c)	(b)	(c)
クラス	クラス名	9	6	6	6	67	
	属性	31	18	17		55	
	操作	11	4	7	4	7	36
クラス間の関係	8	3	5	3	5	38	63

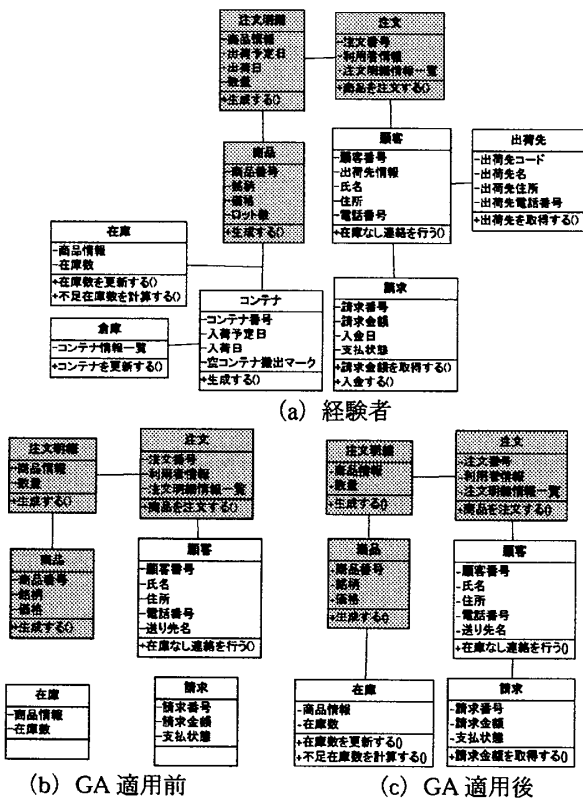


図2 分析クラス図

6. おわりに

本研究では、既存の設計仕様書を利用して、分析クラス図を自動的に生成する手法を考案した。実証実験の結果、本システムを用いることで、オブジェクト指向分析の知識や経験に依存せずに一定の品質の分析クラス図が作成可能であると証明した。また、既存資産を有効に活用するための1つの手法を確立することができた。

今後の課題は、リポジトリの品質向上である。本研究では、リポジトリがあらかじめ用意されていることを前提にしていたため、リポジトリの作成については特に考慮しなかった。しかし、リポジトリの品質は、分析クラス図の品質に大きく影響する。そのため、リポジトリに設計仕様書を登録する前にその設計仕様書の品質を測定し、一定の品質以下の設計仕様書は、リポジトリに登録しないという処理が必要である。また、今回の実証実験は、小規模な言語資源のみで行ったため、共通部分だけの抽出に止まったが、リポジトリの品質次第でさらに正確な出力が期待できると考える。そのため、本手法の実質的な効果を確認するために、上記の方法で大規模なりポジトリを準備し、実証実験を行うことが必要であると考えられる。

今後の発展としては、本システムが同様のシステムに共通するクラスを抽出できることに着目して、システムに応じたフレームワークを作成し、システム設計の支援を行う。このことにより、システムの品質を均一化することが可能となり、システム開発の生産性を向上させることができる。と考える。

参考文献

- [1] Craig Larman : Applying UML and Patterns - An Introduction to Object-Oriented Analysis and Design and the Unified Process -, Prentice Hall (2001).
- [2] 原田実, 野村佳秀, 山本幸二, 大野雅志, 田村浩樹, 高橋史郎 : 自然語要求仕様からオブジェクト指向設計図を自動生成するシステム CAMEO, 情報処理学会論文誌, 情報処理学会, Vol.38, No.10, pp.2031-2039 (1997).
- [3] 河井渉, 大山勝徳, 武内惇, 藤本洋 : オントロジーを用いた要求仕様記述からのオブジェクト抽出法 - 機能分割法 -, 知能ソフトウェア工学研究会技術研究報告, 電子情報通信学会, Vol.104, No.588, pp.25-30 (2005).
- [4] 堀内信吾, 井上正輝, 佐藤正崇, 吉田寛, 井上貴司, 杉本悟, 山村哲哉 : オントロジーを用いた OSS モデルの再利用, テレコミュニケーションマネジメント研究会技術研究報告, 電子情報通信学会, Vol.105, No.527, pp.1-6 (2006).
- [5] Sen. <http://ultimania.org/sen/>
- [6] EDR 辞書. http://www2.nict.go.jp/kk/e416/EDR/J_index.html
- [7] 山崎利治 : 共通問題によるプログラム設計技法解説, 情報処理学会誌, 情報処理学会, Vol.25, No.9, pp.934 (1984).