

要求フレームモデルに 基づくオブジェクト指向分析支援手法

大西 淳、野呂一仁、宇野 勝

立命館大学工学部
525-77 草津市野路町 1916

要求フレームモデルに基づいた日本語要求記述から、オブジェクト指向分析におけるオブジェクトモデルを構築するとともに、他のモデルとの整合性を検証する手法を提案する。最初に、あいまいさや抜けといった誤りを含む日本語要求記述から要求フレームモデルを用いて誤りを除去する。次に日本語文からオブジェクトの候補を抽出し、良いオブジェクトの指針に基づいて候補を絞り込み、オブジェクトモデルを得る。さらに状態遷移図として表された動的モデルやビジュアルな要求言語 VRDL によって表された機能モデルとオブジェクトモデルの間の整合性を検証することによって、3つのモデルを、より完全なものにしていく過程を支援する。

A Supporting Method for Object-Oriented Analysis based on Requirements Frame Model

Atsushi OHNISHI, Kazuhito NORO, Masaru UNO

Department of Computer Science, Ritsumeikan University

1916 Noji, Kusastu 525-77, JAPAN

e-mail: {ohnishi,noro,masaru}@selab.cs.ritsumei.ac.jp

A supporting method of both deriving object model of object-oriented analysis (OOA) from Japanese requirements description and verifying the consistency among the three models of OOA is proposed. This method enables (1) to eliminate errors in Japanese requirements description with Requirements Frame model, (2) to derive candidate of class objects, (3) to construct an object model, and (4) to verify the consistency among the object model, a dynamic model described with a state transition diagram and a functional model described with a visual requirements language VRDL.

1 はじめに

Rumbaugh 等の OMT[11], Coad/Yourdon の OOA[1], Shlaer/Mellor の手法 [12] などに代表されるオブジェクト指向分析では、対象システムを①システムの要素となるオブジェクトの階層構造とオブジェクト間の関連を表すモデル、②システムの動的な振舞いを表すモデル、③システムにおけるデータの変換過程や構造と言った機能を表すモデルの3モデルを導くことを目標としている。まず対象システムに関する記述からオブジェクトを認識するが、この記述に誤りがあるとはならない。また最終的に得られる3つのモデルは対象システムを3つの異なる観点から眺めたものであり、それぞれのモデルに誤りがあるとはならないし、さらにモデル間で矛盾があるとはならない。

しかしながら、最初から対象システムに関する正確な記述を用意するのは難しいといった問題や、オブジェクト指向分析で用意されている記法に基づいてモデルを記述しただけでは、モデルとしての妥当性やモデル間の整合性は保証されないという問題がある。これらの問題を解決し、オブジェクト指向分析の過程を支援するとともに、その成果となるモデルの正当性やモデル間の整合性をより高める手法の確立が望まれる。

本稿では、日本語記述からのオブジェクトモデルの導出とモデル間の整合性検証手法を中心に述べる。

2 オブジェクトモデルの導出支援

オブジェクト指向分析では対象システムに関する日本語記述の名詞や動詞に着目して、オブジェクトの候補を定めることから始めるが、分析の成果は記述の質に大きく依存してしまう。抜けがある記述や、あいまいな記述からは、それらの誤りを訂正しながら解析を進めない限り、得られるモデルも誤りを含んだものになってしまう。また、機能指向で書かれた記述からは、場合によっては属性(データ)がなくメソッド(機能)しか持たないようなオブジェクトとメソッドがなく属性しか持たないオブジェクトが作られてしまう。

ここでは、文法と語彙を限定した制限日本語を導入することによって、日本語記述の誤りを検出しながら解析を進めるとともに、オブジェクトに関する基準を設ける

ことによって、得られるオブジェクトの品質を、より高くする手法を紹介する。

2.1 日本語要求言語

我々は要求フレームと名付けた要求モデル、それに基づく日本語要求言語とその解析システムを開発してきた[5]。要求フレームは要求仕様の枠組を与えるものであり、①名詞と名詞の型、②動詞・形容詞と概念、③文と格フレーム、④文章と機能フレームのそれぞれについての対応関係を定める。この要求フレームによって誤った型の名詞の使用や、必須格の抜け、必須機能の抜けといった誤りを検出するとともに、省略された名詞の同定、代名詞の具体的な名詞への置換ができる[5, 6]。

日本語要求言語 X-JRDL はファイル処理分野に限定して開発しているため、名詞は人間(human)型、機能(function)型、データ(data)型、ファイル(file)型、制御(control)型、装置(device)型の6種の型のいずれかを持つものとし、動詞はデータの流れ、データや機能の構造、ファイルの操作など10種の動作に関する概念のいずれかに、形容詞は「大きい」、「小さい」、「等しい」といった6種の比較演算に関する概念に分類されるものとした。

より広い分野の記述に対処できるよう、名詞の型の種類を拡張したものを表1に示す。

表で最初の6つは具体的に存在する名詞の型であり、続く8つは抽象的な名詞の型である。型はそれぞれが独立しているわけではなく、

『具体的なもの』 ⊃ 『自然物』 ⊃ 『人間』

のように包含関係にあるものもある。

一つの名詞はいくつかの型を持ち、それだけを取り出すとどの型に属するかは一意には決められない。しかしながら文に表される時点では、どれか一つの型に属するものとする。例えば、「家に行く」と「家を建築する」では『家』という名詞は前者では場所・空間型に属し、後者では人工物の型に属するように、文に表された時点では、一つの型に属している。

また動作概念も10種類から拡張するとともに、動作概念間の包含関係を含めて定義している。文献[4]に基づいて動作概念を状態を表すものと動きを表すものに大別し、さらに動きを表す概念を10種類に大分類してい

表 1: 名詞の型

名詞の型	補足
人間	
組織	人間やシステムなどを含む
データ・ファイル	処理対象となるデータやファイル
装置・システム	
自然物	具体的に存在する自然物
人工物	具体的に存在する人工物
事象	
関係	オブジェクト間関係と因果関係、理由などの一般の関係
性質	
役割	
操作・機能	
場所・空間	
時間	
数量	数詞を含んだ名詞
ユーザ定義型	以上の分類に属さない名詞

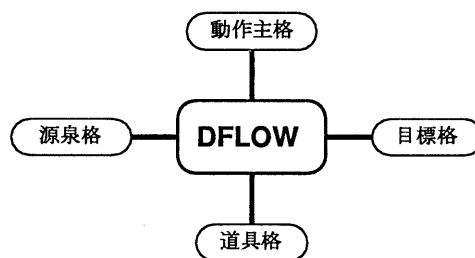
表 2: 概念の分類

動作概念	小分類
状態	存在や所有・関係の認定
関係付け	
時間に関わる動き	
状態の変化	移動・発生・消滅・生産・模様替え
構造	接続・繰り返し・選択
演算処理	
設置	
離脱	
力が加わる	
経済活動	
社会活動	
ユーザ定義概念	

る。状態そのものを表す概念や状態の変化を表す概念および構造を表す概念はもっと細かく分類している。これらを表 2 に示す。

各動作概念は対応する格構造を持つ。例えば、「状態の変化」の中の「移動」を表す動作概念は動作主、源泉、目標、道具という格を持っている。この「移動」の動作主格をデータ型に特化することによって図 1 のような『データの流れ』DFLOW という概念が得られる。DFLOW を具体化した動詞には「渡す」、「出力する」、「入力される」などがある。DFLOW の道具格を「プリンタ」という名詞に特化させた概念 PRINT は DFLOW の下位概念であり、その具体化した動詞には「プリントする」、「印字する」などがある。

X-JRDL では、動詞と名詞の他に、形容詞、接続詞、格助詞、接続助詞、指示代名詞を用いて、連体修飾節による複文、重文、単文を記述できる。解析系は、これらの文を動詞が一つしか現れない単文の集合に分割した後、格構造をもとに抜けた格や代名詞を前方照応により具体的な名詞に置き換える。この時点で不正な型を持った名詞や、必須格に対応するもので照応不可能な名詞が



概念	格	名詞の型
DFLOW	動作主格	data
	源泉格	functionかhuman
	目標格	functionかhuman
	道具格	device

図 1: 概念 DFLOW(データの流れ) の格フレーム

抜けとして検出できる。

このように制限された文法と語彙を持つ日本語を定め、それを用いることによって、記述の誤りを検出し、正しい要求を導くことができるとともに、以下に述べるようにオブジェクトの抽出が容易になる。

2.2 オブジェクトの候補の選定

オブジェクト指向分析ではオブジェクトの候補を選定する際、日本語記述を整形して、動作主体を主語になるように変形したり、代名詞や省略された名詞を具体的な名詞に置き変える必要がある。本手法では、これらの作業の殆どは日本語記述解析時に計算機によって自動的に行われる。また同義語を統一する必要があるが、名詞は型によって分類されているため、同義語をより見つけやすくなる。さらに、動きを表す動作の主体、動作の対象に相当する名詞、ならびに特定の型に分類された名詞をオブジェクトの候補とするが、これらについても、本手法では日本語記述解析終了時に、抜けている必須格や代名詞を照応することによってすべて判明している。

このように本手法では、オブジェクトの候補の選定についてはかなりの部分が自動化される。

2.3 オブジェクトの抽出

選定されたオブジェクトの候補はオブジェクトとして妥当であるかどうかを評価されなければならない。我々は以下に示すオブジェクトに対する基準を設けることによって、得られるオブジェクトの正当性を検証する。

1. オブジェクトは属性を持つ。
2. オブジェクトはメソッドを持つ。
3. メソッドと属性は密な関係がある。
4. オブジェクトは他のオブジェクトと関係 (関連, is-a, part-of) を持つ

1 番目と 2 番目の基準は、オブジェクトの特長の一つがデータと手続きが一体化していることであり、属性とメソッドを合わせ持つことが望ましいという事実を反映している。3 番目の基準は、単にメソッドと属性を持つだけでなく、互いに関係の深いものを取り入れることによってオブジェクトを強固なものとするために導入して

いる。オブジェクト候補の属性集合 A とメソッド集合 M をそれぞれ、

$$A = \{a_1, a_2, a_3, \dots, a_j\} \quad (1)$$

$$M = \{m_1, m_2, m_3, \dots, m_i\} \quad (2)$$

とした時、例えば属性集合 A とメソッド集合 M が次のように分割され

$$A_1 = \{a_1, a_2, \dots, a_l\}, A_2 = \{a_{l+1}, a_{l+2}, \dots, a_j\}$$

$$M_1 = \{m_1, m_2, \dots, m_k\}, M_2 = \{m_{k+1}, m_{k+2}, \dots, m_i\}$$

A_1 と M_1 、 A_2 と M_2 がそれぞれ相互関係があり、 A_2 と M_1 および A_1 と M_2 の間には相互関係がない場合には、もとのオブジェクト候補を A_1 と M_1 、および A_2 と M_2 をそれぞれ属性とメソッドとする二つのオブジェクト候補に分割することによって、よりオブジェクトを強固にできる。

3 モデルの表現

3.1 オブジェクトモデルの記述

本稿でのオブジェクトはクラスとインスタンスを総称したものとする。クラスオブジェクトは OMT の記法に準拠したものとする。オブジェクトとオブジェクト間の関係が明記されて、オブジェクトモデル (Object model) が形成される。本稿では関係は ① クラス間の概念的な包括関係を表す is-a, ② オブジェクト間の構造的な集約関係を表す part-of, ③ オブジェクト間の参照・利用関係を表す関連に分類されるものとする [1, 2]。

記述されたオブジェクトモデルは 2.3 節の基準と照らし合わせて検証される。

3.2 動的な振舞いのモデルの記述

動的な振舞いはある事象に伴う個々のオブジェクトの状態の遷移を図や表で表したものとする。状態遷移図の記法は OMT の動的モデル (Dynamic model) の記法に準拠したものとする。動的モデルのオブジェクトの状態はオブジェクトの属性の内の特定の属性がとる値の組み合わせによって決定するものとしている。従って状態を決定する属性の値が変わることによって状態が遷移する。

動的モデルの正当性検証では

- ① 各状態とそれを決定する属性の値が明示されているか、
- ② 各状態遷移での事象に伴う動作によって変化する属性の値が遷移前の状態と遷移後の状態とで矛盾していないか、
- ③ 日本語記述中の状態を変化させる概念に属する動詞が、その動作主に相当するオブジェクトの状態遷移での事象に伴う動作と矛盾していないか

を調べる。

3.3 機能を表すモデルの記述

機能を表すモデル (Functional model) は、筆者が開発してきたビジュアルな要求言語 VRDL (Visual Requirements Description Language) [8, 10] によって、データフローが表されているものとする。VRDL の特長を以下に示す。

1. アイコンの形状と意味を定義できる。
2. 複合的なアイコンを容易に作成できる。
3. アイコンと矢印をエディタ上で配置していくことによって要求を記述する。
4. VRDL 記述に用いられたアイコンの動作を与えることができ、その動作記述を解釈実行することによって、記述の中のアイコンの動作をアニメーションとして表示できる。
5. VRDL による記述を標準的なアイコンを用いた記述へ変換できる。
6. VRDL 記述に対して、要求定義環境 CARD [7] のツール群による検証・検索・フロー表示・設計支援ができる。

入力のない機能や出力のない機能がないか調べるといった静的な正当性検証に加えて、上の 4 番目の特長を活かして、次のような動的な正当性検証を行なう。データの流れや装置の動きの動作を動作記述言語によるシナリオとして記述し、シナリオを実行することによって、データの流れや装置の動きを逐次的にアニメーションとして表示する。これによりデータの流れる方向とデータ処理の順序が正しいかどうかを確認できる [9]。

OMT ではシナリオや事象トレースを考えて、それから状態遷移を導く [11] ので、最初に考えたシナリオをそのまま用いる方法も考えられるが、このシナリオは動的モデルを導くために一例をとりあげたものに過ぎず、すべての状態遷移に対応しているとは限らない。ここでは状態遷移表現の任意の部分集合の正当性を検証することを考慮して、新たにシナリオを導出する。

4 モデル間の整合性検証手法

得られた 3 つのモデルについて、それぞれ 2 つのモデル間での整合性を検証する。これによって誤りが検出されると、もう一度各モデルを修正することによって対処する。

4.1 オブジェクトモデルと動的モデルの整合性検証

まず動的モデルに対応するオブジェクトがあるかどうかを定めたいうで以下の整合性を検証する。

- ① オブジェクトモデルでのオブジェクトの持つ属性と動的モデルで状態を決める属性の間に矛盾がないか、
- ② 属性の取りうる値と状態を決める属性の値に矛盾がないか、
- ③ オブジェクトの持つ操作と状態の持つ活動や状態遷移に伴う動作の間に矛盾がないか

を調べる。

4.2 オブジェクトモデルと機能モデルの整合性検証

オブジェクトモデルにおけるオブジェクトと機能モデルのアイコンが対応するかどうかを定めたいうで以下の整合性を検証する。

- ① オブジェクト間の関連とアイコン間のデータフローに矛盾がないか、
- ② オブジェクトの持つ操作とデータフローという動作の間に矛盾がないか

を調べる。

4.3 動的モデルと機能モデルの整合性検証

機能モデルのアイコンに対応する動的モデルがあるかどうかを定めたいうで以下の整合性を検証する。

- ① 動的モデルにおける状態遷移に伴う入出力動作で、その入出力データと機能モデルにおけるデータフローに矛盾がないかどうか

を調べる。これは機能モデルの正当性検証で述べたシナリオ作成段階で調べられる。

5 おわりに

オブジェクト指向分析におけるモデルの正当性とモデル間の整合性検証のための手法を紹介した。現在のところ、要求言語の文法についてその拡張を検討しているとともに、拡張した要求フレームモデルに基づいた日本語処理系の開発を進めている。一方、VRDLの処理系と「動作シナリオ」の解釈実行系については、それぞれのプロトタイプを開発済みであり、これらの機能強化に加えて状態遷移図からの「動作シナリオ」導出部分のプロトタイプの開発と評価を進めている段階である。

今後の課題としては、

1. 数多くの例題に適用して手法を評価する
2. モデル表現の誤りを発見した場合の対処法を確立する。

があげられる。2番目の課題については、一つのモデルを修正・変更した場合、他のモデルへの変更の波及を検出する仕組みを研究したいと考えている。

参考文献

- [1] Coad, P., Yourdon, E.: Object-Oriented Analysis, 2nd ed., Prentice-Hall, Englewood Cliffs, NJ, 1991. (羽生田栄一監訳: オブジェクト指向分析(OOA)第2版、トッパン、1993.)
- [2] 本位田真一、山城明宏: 「オブジェクト指向システム開発」、日経BP社、1993.
- [3] 情報処理学会: 「オブジェクト指向分析・設計チュートリアル資料」、情報処理学会、1993.
- [4] 情報処理振興技術協会技術センター: 「計算機用日本語基本動詞辞書 IPAL - 解説編 -」、61 技 -073, 1987.
- [5] 大西 淳、阿草清滋、大野 豊: 「要求フレームに基づいた要求仕様化技法」、情報処理学会論文誌 31 巻 2 号, 1990 (pp.175-181).
- [6] 大西 淳: 「要求定義のためのコミュニケーションモデル」、情報処理学会論文誌 33 巻 8 号, 1992 (pp.1064-1071).
- [7] Ohnishi, A., Agusa, K.: "CARD: A Software Requirements Definition Environment," Proc. of IEEE Int'l Symp. Requirements Engineering, San Diego, CA, U.S.A., Jan. 1993, pp.90-93.
- [8] 大西 淳: 「ビジュアルなソフトウェア要求仕様化技法」情報処理学会研究報告, Vol.93, No.13, SE90-8, 1993 年 2 月, pp.57-64.
- [9] 大西 淳: 「オブジェクト指向分析における機能モデル検証支援」情報処理学会研究報告 Vol.94, No.6, SE96-2, 1994 年 1 月, pp.9-16.
- [10] Ohnishi, A.: "A Visual Software Requirements Definition Method," Proc. of IEEE Int'l Conf. Requirements Engineering (ICRE'94), Colorado Springs, CO, U.S.A., Apr. 1994, pp.194-201.
- [11] Rumbaugh, J., Blaha, M., Premerlani W., Eddy F., Lorensen W.: Object-Oriented Modeling and Design, Prentice-Hall, Englewood Cliffs, NJ, 1991. (羽生田栄一監訳: オブジェクト指向方法論 OMT、トッパン、1992.)
- [12] Shlaer, S., Mellor, S.: Object-Oriented Systems Analysis, Prentice-Hall, Englewood Cliffs, NJ, 1988, (本位田真一、山口亨訳: オブジェクト指向システム分析、啓学出版、1990.)
- [13] Thayer, R., Dorfman, M.: "System and Software Requirement Engineering," IEEE Computer Society Press Tutorial, Los Alamitos, CA, 1990.
- [14] Wang, W., Hufnagel S., Hsia, P., Yang S.M.: "Scenario Driven Requirements Analysis Method," Proc. IEEE Int'l Conf. Systems Integration(ICSI), 1992, pp.127-136.