

Title	多重視点設定可能な、OMTに対するCASEデータベースの設計とPCTE上での実装
Author(s)	西山, 雄; 落水, 浩一郎
Citation	Research report (School of Information Science, Japan Advanced Institute of Science and Technology), IS-RR-94-0004S: 1-37
Issue Date	1994-02-02
Type	Technical Report
Text version	publisher
URL	http://hdl.handle.net/10119/8422
Rights	
Description	リサーチレポート（北陸先端科学技術大学院大学情報科学研究科）

多重視点設定可能な、OMT に対する
CASE データベースの設計と PCTE 上での実装

西山雄 落水浩一郎

1994 年 2 月 2 日

IS-RR-94-0004S

北陸先端科学技術大学院大学

情報科学研究科

〒 923-12 石川県能美郡辰口町旭台 15

nisiyama@jaist.ac.jp

ochimizu@jaist.ac.jp

©Masaru Nishiyama, 1994

ISSN 0918-7553

もくじ

1 はじめに	4
2 多重視点	4
3 視点	5
4 各モデルの導出方法	9
5 分析例	12
5.1 オブジェクトモデルの導出	12
5.2 動的モデルの導出	12
5.3 機能モデルの導出	18
6 計算機による手法の支援	21
6.1 インスタンス図のスキーマ	21
6.2 オブジェクトモデルのスキーマ	22
6.3 事象トレース図のスキーマ	23
6.4 事象フロー図のスキーマ	25
6.5 動的モデルのスキーマ	25
6.6 パラメータフロー図のスキーマ	26
6.7 機能モデルのスキーマ	28
6.8 スキーマの合成	29
7 PCTE への実装	31
7.1 PCTE のオブジェクトベース	31
7.2 オブジェクトベースへの定義	31
8 手法の評価	34
9 おわりに	34
A 「キャッシュディスペンサーシステム」に関する問題記述	36

図一覧

1	OMT における問題世界の仕様化	5
2	OMT の 3 つのモデルとモデル間の重複	6
3	インスタンス図 記法	8
4	パラメータフロー図	9
5	中間的視点を用いたモデルの導出	10
6	キャッシュディスペンサーシステム インスタンス図	12
7	キャッシュディスペンサーシステム オブジェクトモデル	13
8	顧客、CD、銀行、口座に対する事象トレース図	14
9	顧客、CD、銀行、口座に対する事象フロー図	15
10	送信事象からオペレーションを明らかにする	16
11	クラス CD に対する動的モデル	17
12	顧客、CD、銀行、口座に対するパラメータフロー図	18
13	CD クラスに対する最上位の機能モデル	19
14	銀行クラスに対する最上位の機能モデル	20
15	口座クラスに対する最上位の機能モデル	20
16	インスタンス図を構成する要素	21
17	インスタンス図が表現する要素のスキーマ	22
18	オブジェクトモデルを構成する要素	22
19	オブジェクトモデルが表現する要素のスキーマ	23
20	事象トレース図を構成する要素	23
21	事象とオペレーションの関係	24
22	事象トレース図が表現する要素のスキーマ	24
23	事象フロー図を構成する要素	25
24	事象フロー図が表現する要素のスキーマ	25
25	動的モデルを構成する要素	26
26	動的モデルが表現する要素のスキーマ	26
27	パラメータフロー図を構成する要素	27
28	パラメータフロー図が表現する要素のスキーマ	27
29	機能モデルを構成する要素	28
30	機能モデルが表現する要素のスキーマ	28
31	問題世界を表現する要素のスキーマ	29
32	OMT の視点	30
33	問題世界を表現する要素のスキーマ	32
34	OMT を構成する要素	33

表一覧

1	7つの視点	6
2	関係型の分類 [鯨坂 93]	31

1 はじめに

Rumbaugh の OMT[Rumbaugh92] に代表されるオブジェクト指向モデリングは通常オブジェクトモデル、状態遷移モデル、データフローモデル等の仕様定義のための複数のモデルを持つ。これらはモデリングされる問題世界を、いくつかの側面から眺めた物である。ここで扱う OMT ではオブジェクトモデル、動的モデル、機能モデルの三つのモデルを最終生成物とし、それらは次のような側面を表現している。

- オブジェクトモデル：問題世界の静的構造
- 動的モデル：オブジェクトと関連の時間的变化
- 機能モデル：問題世界内部の計算動作

OMT ではこの三つのモデル間の相互関係の吟味を、3 つのモデルがほぼ完成してから行う。そのためモデル間の一貫性を保証しにくいという問題がある。特にオブジェクトモデルと機能モデルではその構築方法がボトムアップとトップダウンというようにまったく異なるため、その一貫性を取るにはたいへんな手間がかかる。

この問題に対し、本稿では OMT のモデルが表現しようとしている問題世界の要素¹について注目する。これによりモデル間の関係を明らかにする。この結果、モデル間の定義のオーバーラップ部分に対してそれぞれ「視点」を導入し、3 つのモデルの作成を連続的に行えるように方法論を改良をすることで、モデル間の一貫性の検証を分析者がより容易に行えるようにした。

また視点の構造をとらえ、視点間の関係からそれらの構造を合成することで、問題世界を表現することができる問題世界の要素スキーマを設計する。このスキーマは問題世界のすべての要素を表わしているが、分析者はその要素群の一部に注目した視点により分析を行う。分析者は視点を切りかえつつ要素を定義、検討することで、モデル間の一貫性を取ることができる。このようにどのモデルを使って分析を行うかではなく、問題世界のどの要素を見て分析をするかということを本稿では多重視点と定義する。この要素スキーマから注目する要素を視点として切り出すのが視点スキーマである。

本稿ではこのような要素スキーマと視点スキーマの PCTE 上のオブジェクトベースへの実装も行う。

2 多重視点

本論文でいうところの視点とは、仕様化作業のプロセス [西村 93] に対応する用語ではなく、OMT と同じく問題世界を表現するモデルに対応する。

OMT では問題世界に対する視点、オブジェクトモデルと動的モデル、そして機能モデルをそれぞれ作り、最後に分析者が3つのモデルをつきあわせ、検討することで、3つのモデル間の一貫性をとる。そのため OMT の三つのモデルは図 1²が表すように、まったく異なる要素群を表現しているかのように構成されがちである。さらに分析者はモデ

¹例えばクラスや事象といった OMT のモデルで表現しようとしている問題世界の構成部品。

²オブジェクトモデルが表現している要素「オペレーション」はクラスの操作を表わす。これは機能モデルで用いられる操作とまぎらわしいため別に用語を定義した。

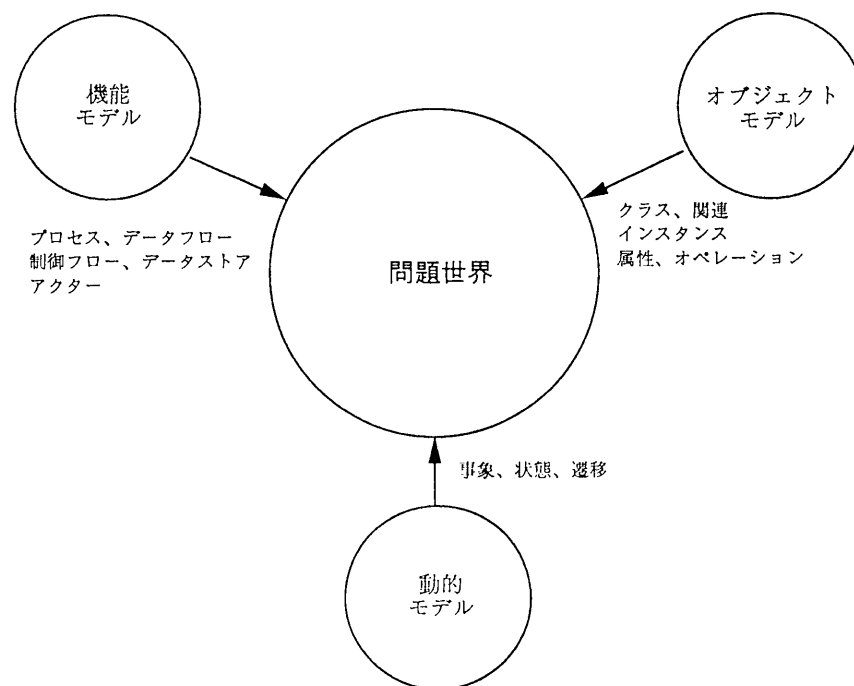


図 1: OMT における問題世界の仕様化

ル間の一貫性をとるさいに目的とする処理、すなわち機能モデルを優先し、オブジェクトモデルを変更することも OMT では許されている。このためオブジェクトモデルと問題世界との対応が悪くなる危険性も持つ。

しかしその要素ひとつひとつには、要素名は異なっても共通性のある物もある。例えばオブジェクトモデル上のオペレーションは、機能モデルにプロセスとして現われる。また、動的モデルの事象の半分以上はオブジェクトモデルのオペレーションと対応している。

これから三つのモデルは図 2 のように、重複部分をもちながら問題世界を表現していることがわかる。そしてその重複した要素がモデルにより一致しないとき、モデル間の矛盾が発生するものと考えられる。

これらの矛盾をなくすには、この重複部分の要素を含む中間的な視点を用いるという方法が一つの手段として考えられる。例えば動的モデルはクラス間の事象の流れを記述した事象フロー図を用いることで導出でき、中間的視点である事象フロー図で事象とオペレーションの対応を吟味することで、オブジェクトモデルと動的モデルの間の矛盾を検証することができる。このようにモデルとモデルの間に存在する要素に注目し、モデル間の橋わたしをするのが中間的視点である。

3 視点

本稿では表 1 にある 7 つの視点を基本的な視点として用いて分析をすすめる。

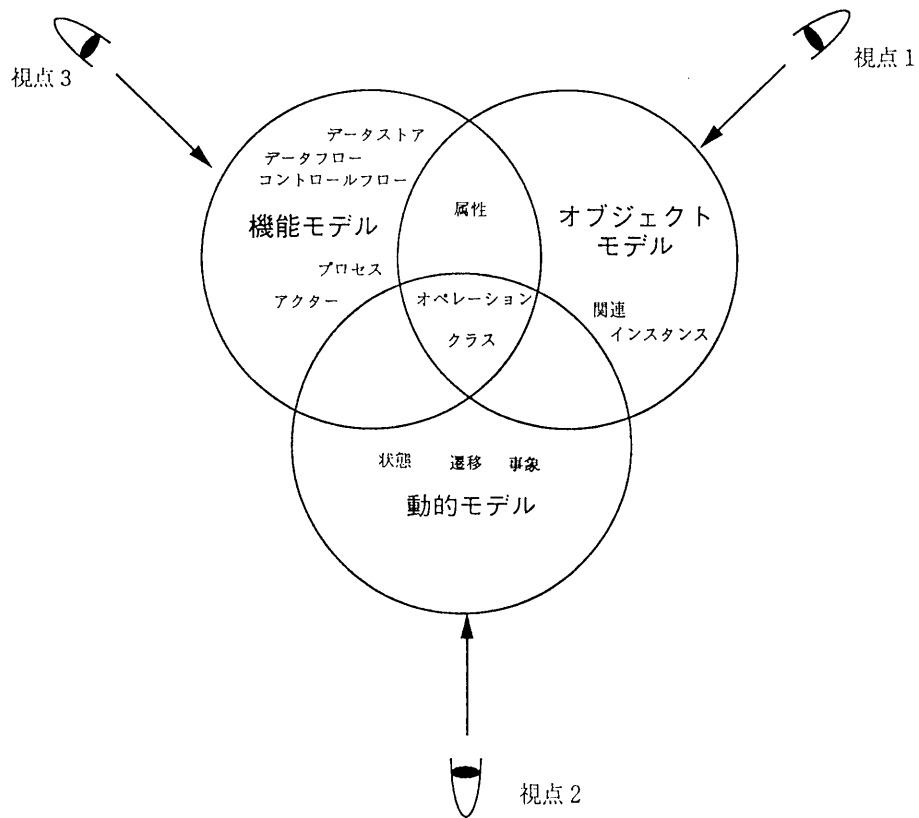


図 2: OMT の 3 つのモデルとモデル間の重複

視点	意味
インスタンス図	問題世界の一例
オブジェクトモデル	システムの静的構造
事象トレース図	オブジェクト間の事象の流れの一例
事象フロー図	クラス間の事象の流れ
動的モデル	オブジェクトと関連の時間的变化
パラメータフロー図	クラス間のデータ、信号の流れ
機能モデル	システム内部の計算動作

表 1: 7 つの視点

インスタンス図 オブジェクトとそのリンクを表した図である。これは問題記述書からオブジェクトモデルを導出する手がかりとなる。[Rumbaugh92] ではその記法をはっきりとは定義されていなかったなので、ここで図3のように定義する。

オブジェクトモデル オブジェクトの集合であるクラスとリンクの集合である関連、そしてクラスの属性とオペレーションを表す。なおオペレーションは事象フロー図から、属性は機能モデルから導き出される。また機能モデルでは見落としていたクラスの補完も行われる。

事象トレース図 インスタンス間の事象の流れを記述する。これはインスタンス図と次に説明する事象フロー図の間の中間的視点である。

事象フロー図 クラス間の事象の流れを記述する。この図はオブジェクトモデルと動的モデル、そして事象トレース図と動的モデルをつなぐ中間的視点である。

動的モデル クラス内の状態とその変化、そしてそれに伴う事象を記述する。

パラメータフロー図 OMT ではオブジェクトモデルと機能モデルをつなぐ視点は提供されていない。そこで本稿では図4のようなクラス間のデータの流れを記述したパラメータフロー図を導入する。これによりクラスへのデータの入出力がわかる。

機能モデル データの変換処理を記述する。この変換処理を行うプロセスのうち、クラスが行うサービスとして公開されるものがオブジェクトモデルのオペレーションである。またデータストアの一部はクラスの属性、ときにはクラス自身としてオブジェクトモデルで公開されている。

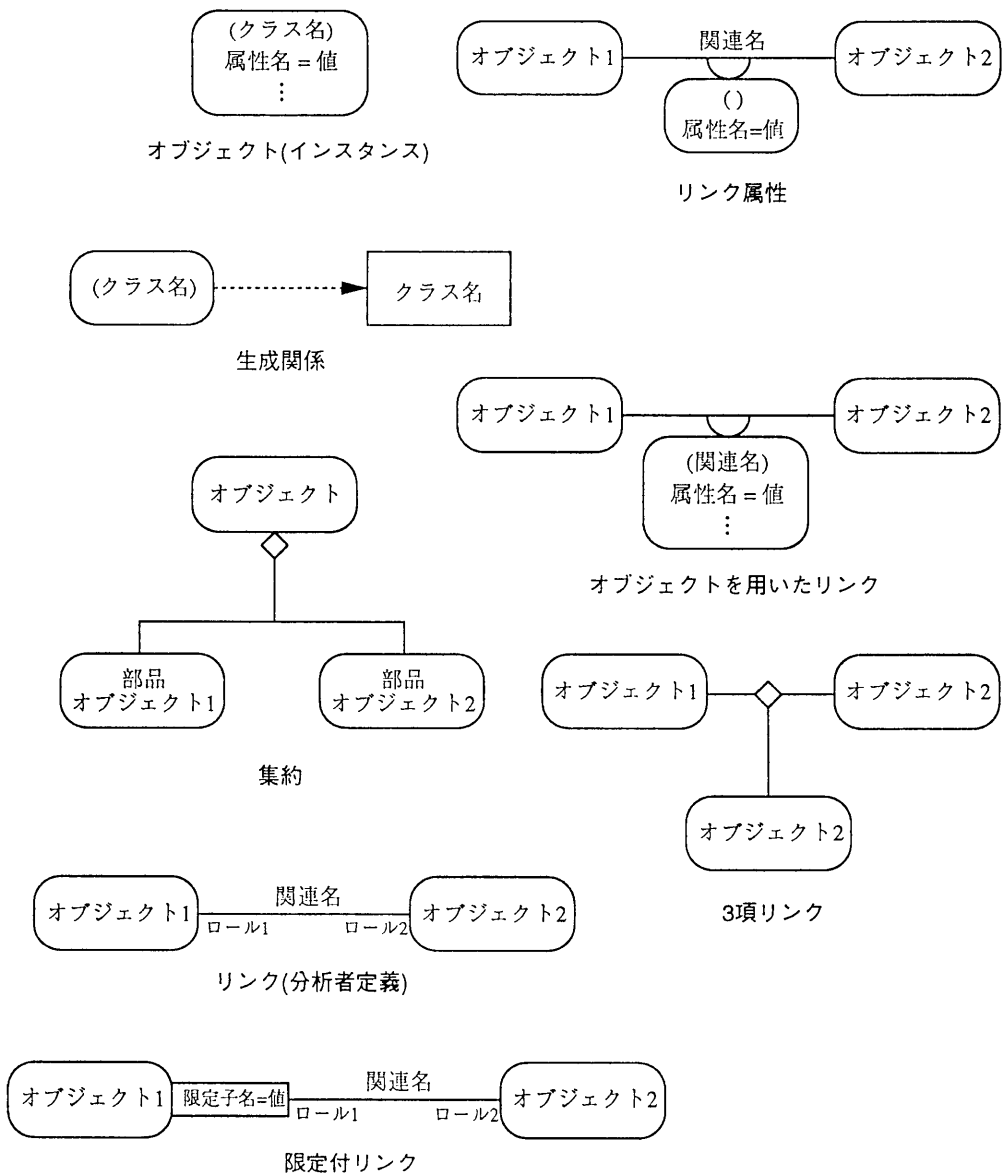


図 3: インスタンス図 記法

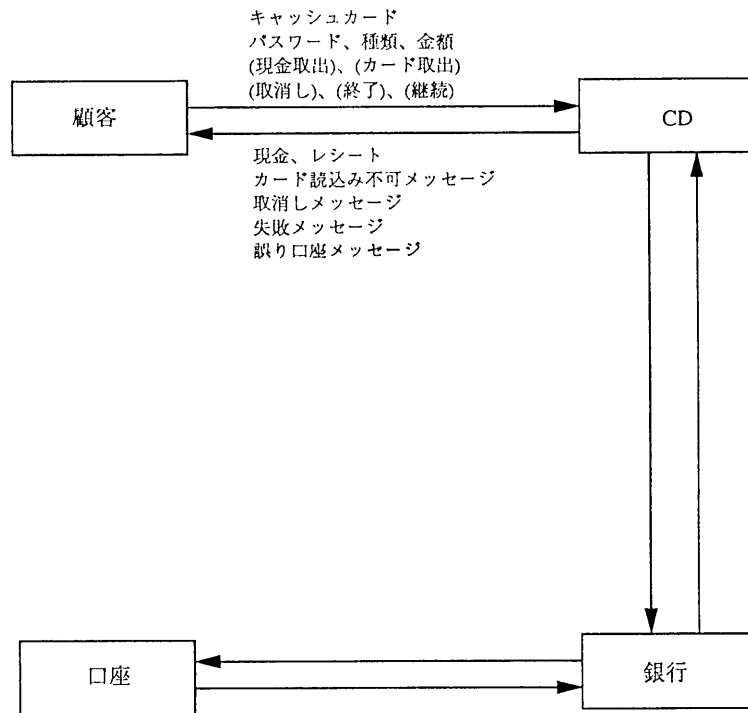


図 4: パラメータフロー図

4 各モデルの導出方法

図 5 は 3 つのモデルの導出に関わる視点とそのキーとなる要素を示したものである。

図 5(a) オブジェクトモデルの導出

オブジェクトモデルは問題記述書から直接導出するのではなく、「インスタンス図」をまず作成し、インスタンスの集合であるクラス、リンクの集合である関連を定義することで導出する。インスタンス図を介してオブジェクトモデルを導出するのは、

- オブジェクトモデルと問題世界との対応性を良くする。
- 分析者による問題世界の捉え方の例をインスタンス図として残しておく。

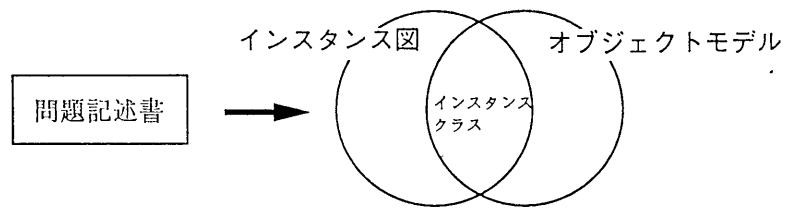
という 2 つの目的を満たすためである。

オペレーションについては動的モデルの導出の時に定義される。

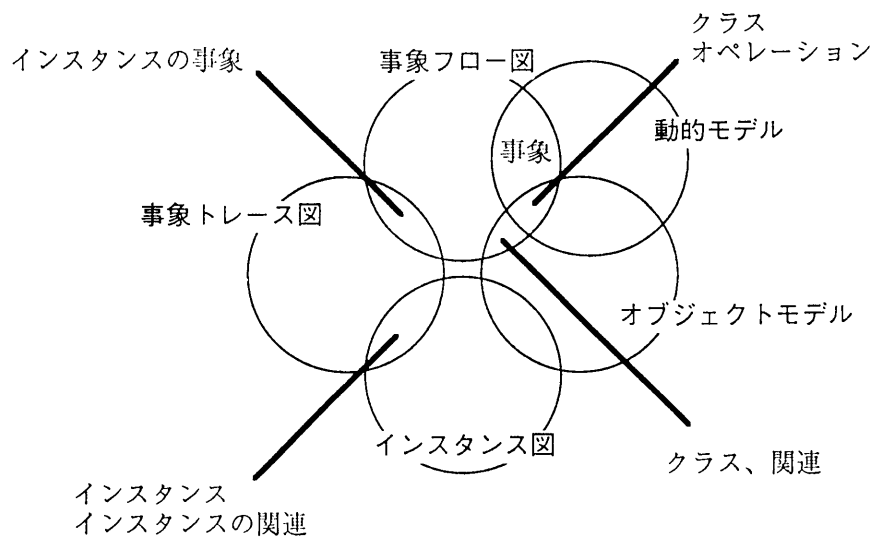
図 5(b) 動的モデルの導出

動的モデルの導出には、「事象トレース図」と「事象フロー図」を用いる。まず事象トレース図でインスタンス間の事象の流れを調べる。分析者はこの図における事象の流れと、問題世界で現実に行われている事象を比較することでインスタンス図の問題世界との対応を確認することができる。

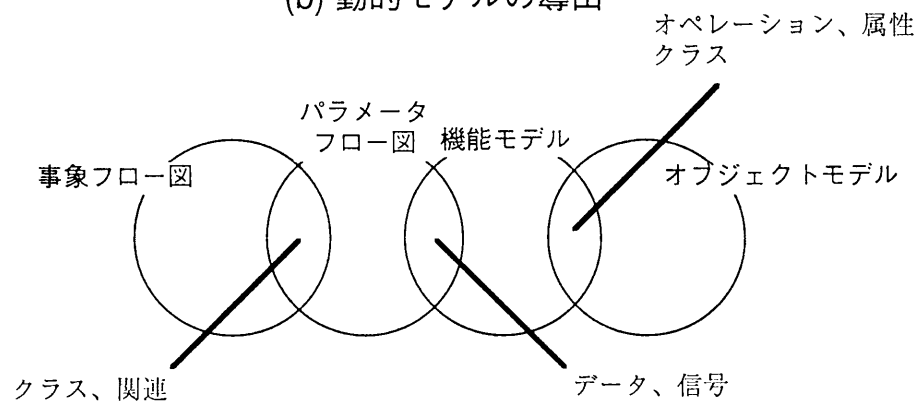
次にインスタンス間の事象の流れをもとにクラス間の事象の流れを記述する。これが事象フロー図である。クラス間の事象の流れはオブジェクト



(a) オブジェクトモデルの導出



(b) 動的モデルの導出



(c) 機能モデルの導出

図 5: 中間的視点を用いたモデルの導出

モデルで定義されたクラスとその関連、そして事象フロー図で定義されたインスタンス間の事象より定義される。この事象フロー図で、あるクラスへ入力事象があるということは、そのクラスに対応できるオペレーションをもつことを意味している。すなわちこの事象フロー図によって事象とオペレーションの対応をとることができる。この事象がクラス内でどのように発生していくかを表したのが動的モデルである。すでにオペレーションと対応のとれている事象を用いているので、これをもとにして定義された動的モデルはオブジェクトモデルと矛盾しない。

図 5(c) 機能モデルの導出

OMT では機能モデルは問題記述書から直接作成されていた。そのためモデル完成時にオブジェクトモデルとの一貫性を取るのに分析者は大きな労力がはらい、無理をして一貫性を取ろうとした結果、オブジェクトモデルをゆがめてしまうこともある。

そのようなことを防ぐため、本稿では事象フロー図から得られる「パラメータフロー図」を新たな視点として導入し、これによりオブジェクトモデル、動的モデルとの一貫性を検討しながら、機能モデルの分析を行えるようにする。

パラメータフロー図はクラス間を流れるデータを記述したものである。この視点は事象フロー図で表現されるクラス間でやりとりされる事象について、それに伴うデータや信号を記述する。これを事象フロー図の事象とおきかえることによりパラメータフロー図は視点を形成する。この視点によってクラスごとに入力、出力されるデータ、信号がわかる。

機能モデルはそのプロセスをオブジェクトモデルのオペレーションをベースに定義し、そして入出力されるデータフロー、制御フローをパラメータフロー図により定義する。このオペレーションとデータフロー、制御フローはもともになったのはともに事象であるので、オペレーションとこれらフローの対応もこの段階でとれている。そのため機能モデルではオペレーションとなっているプロセスの詳細化と、プロセスがアクセスするデータストアを明らかにし、そのデータストアのうちどれを属性、もしくはクラスとして定義するかを検討する。これによりオブジェクトモデルが補完される。

このように本稿で提案する手法では、視点を切りかえていく時に積極的に視点の要素の対応をとっていくので、視点間の矛盾は発生しない。

5 分析例

この章では7つの視点を使った分析例として、「キャッシュディスペンサーシステム」を分析する。

5.1 オブジェクトモデルの導出

オブジェクトモデルを作成するため、まずインスタンス図を問題記述書³から作成する。キャッシュディスペンサーシステムのインスタンス図を図6に示す。

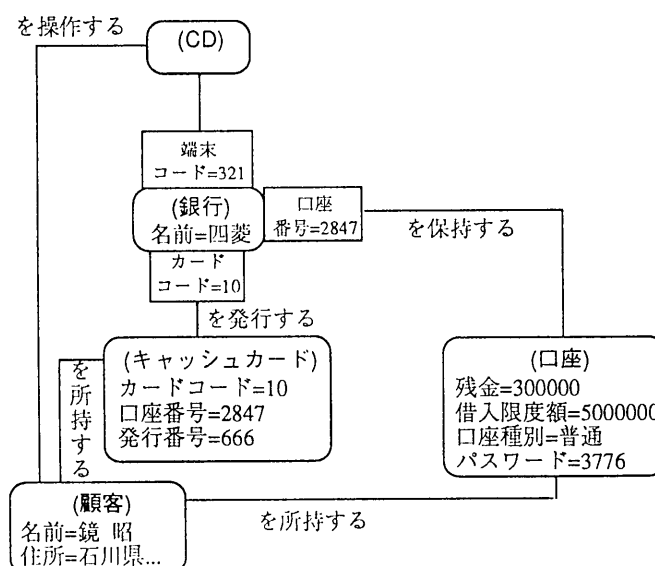


図 6: キャッシュディスペンサーシステム インスタンス図

このインスタンス図では、顧客がCD(キャッシュディスペンサー)を操作することで、銀行にある自分の口座にアクセスできることが表現されている。キャッシュカードはCDを操作するときにCDにデータとしてわたされるが、そのような機能的な制約はインスタンス図で書くべきでなく、機能モデルで記述される。

このインスタンス図のうち、クラスとリンクよりオブジェクトモデルを導出することができる。それが図7である。

5.2 動的モデルの導出

動的モデルはインスタンス図とオブジェクトモデルより作られる事象トレース図と事象フロー図によって導出される。事象トレース図はインスタンスの間をどのような事象がやりとりされるかを時間軸にそって並べたものである。図8に顧客、CD、銀行、口座オブジェクトに注目した例を示す。

ここで「ステップ」という考えを導入する。その定義は次の通りである。

³ Appendix A 参照

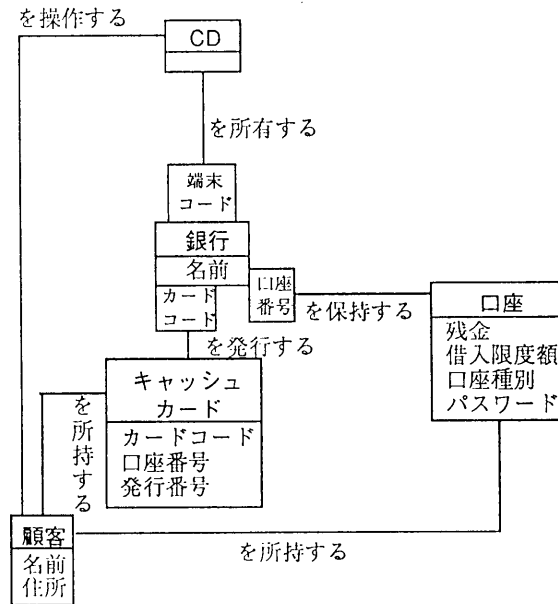


図 7: キャッシュディスペンサーシステム オブジェクトモデル

ステップ = { 送信事象 } + { 返信事象 }

すなわちステップとは他のオブジェクトへ処理を依頼するメッセージとその結果であるメッセージもしくはシグナルの組である。これによって事象を送信と返信の2つに分類し、ひとつの処理の単位を表す。

送信事象と返信事象は1対1に対応するのではなく、複数の送信事象に対してひとつの返信事象、ひとつの送信事象に対して複数の返信事象、送信事象のみで返信事象がないなどの可能性がある。しかし、返信事象のみで送信事象がない場合はない。それは処理を依頼されてもいないのに、返信するようなことは発生しないからである。

図8で事象をステップに組分けした。ステップは□で事象をくくることで表している。例えばステップ1で顧客はCDに対して「カード挿入」という送信事象を送っている。これによってCDとのやりとりを開始する。

どれが送信事象でどれが返信事象であるかは、複数の事象を一つの処理として組み合わせ、その処理がどの事象により開始されるかを考えることで決定される。例えば「カード挿入」は顧客とCDとの事象のやりとりを開始するキーとなる事象である。この事象の後にはCDが能動的に処理を進める。このため「パスワード要求」と「カード挿入」は別のステップとなる。これによりこの2つはともに送信事象であることがわかる。またその次の事象「パスワード入力」は「パスワード要求」によって発生した事象である。これよりこの2つの事象はひとつのステップを表し、「パスワード入力」は返信事象だということができる。

前述の通り、ステップとはある処理を他のオブジェクトに依頼し、その処理が終了し結果が帰ってくるまでの単位である。すなわちあるステップの間はひとつの状態にあり、ステップから次のステップに移るときに状態の遷移を起す。これより送信事象が状態の

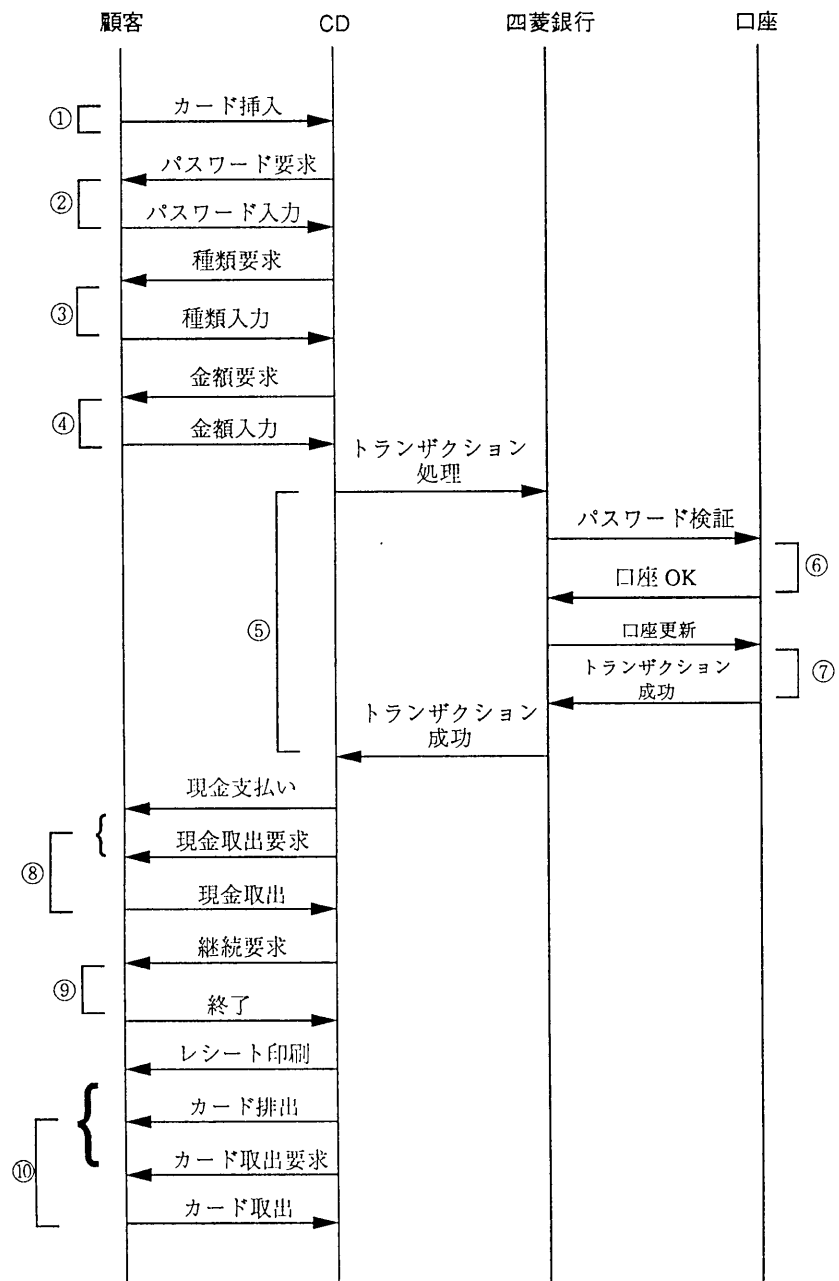


図 8: 顧客、CD、銀行、口座に対する事象トレース図

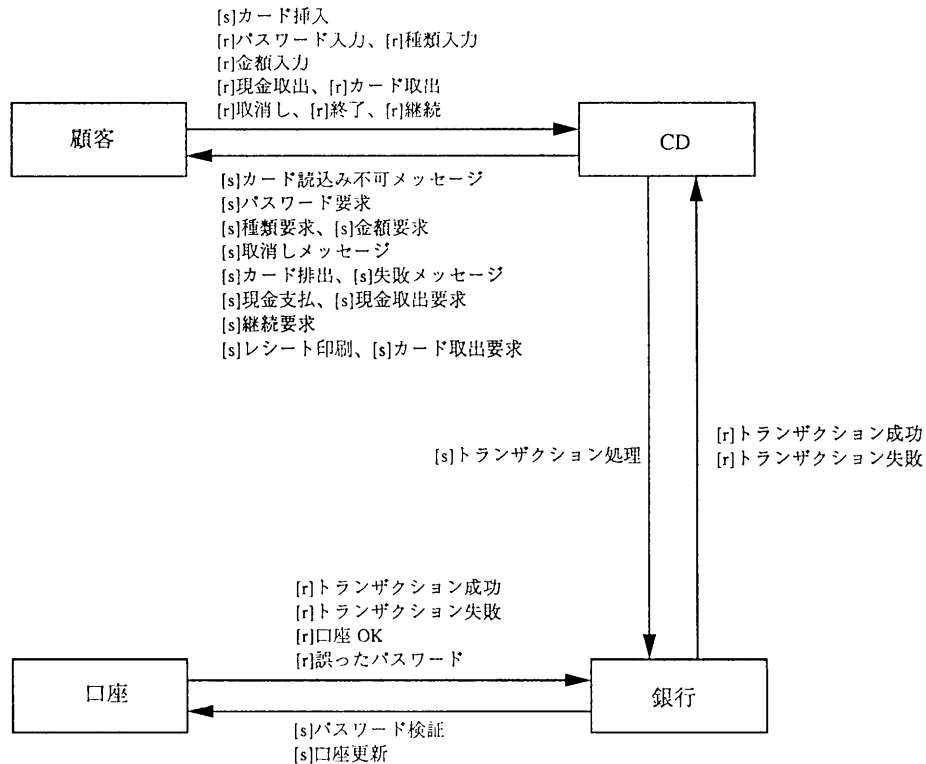


図 9: 顧客、CD、銀行、口座に対する事象フロー図

活動、もしくは内部動作となる。

次に事象トレース図をクラスの視点から見直し、事象フロー図を作成する。顧客、CD、銀行、口座に対する事象フロー図として図 9 を示す。この図では図 8 でわかった送信事象と返信事象を [s] 送信事象名、[r] 返信事象名というように表記している。

事象の送信は送信先オブジェクトがそのメッセージを受けとれるということなので、送信先オブジェクトには送信事象に対応するオペレーションが存在するはずである。また返信事象は送信事象に対する返り値、もしくは制御信号と考えられる。

これより図 10 のように事象からオペレーションを定義することができる。この図は事象フロー図を拡張し、クラスにオペレーションも表記している。CD クラスの上にあるのが、オペレーションと事象の対応である。他のクラスについては図が繁雑になるので省略する。この図 10 のように基本となる視点をもとにして要素を追加し、その要素の対応を検討できるのが多重視点の特徴である。

これらをもとに動的モデルを導出する。これには事象トレース図で定義したステップを利用する。その例として CD の動的モデルを図 11 を示す。

図の中で数字をふった網かけは図 8 のステップと一致している。ステップが状態とそこからの遷移を表わしているのがこれによりわかる。図 8 は正常動作のシナリオだったので、それに異常状態の遷移を加えることでこの動的モデルを得ることができる。

この動的モデルで表されている状態内の事象は、CD 以外のクラスに事象を送っているが、どのクラスへの事象かは事象フロー図ですでにわかっている。またその事象によっ

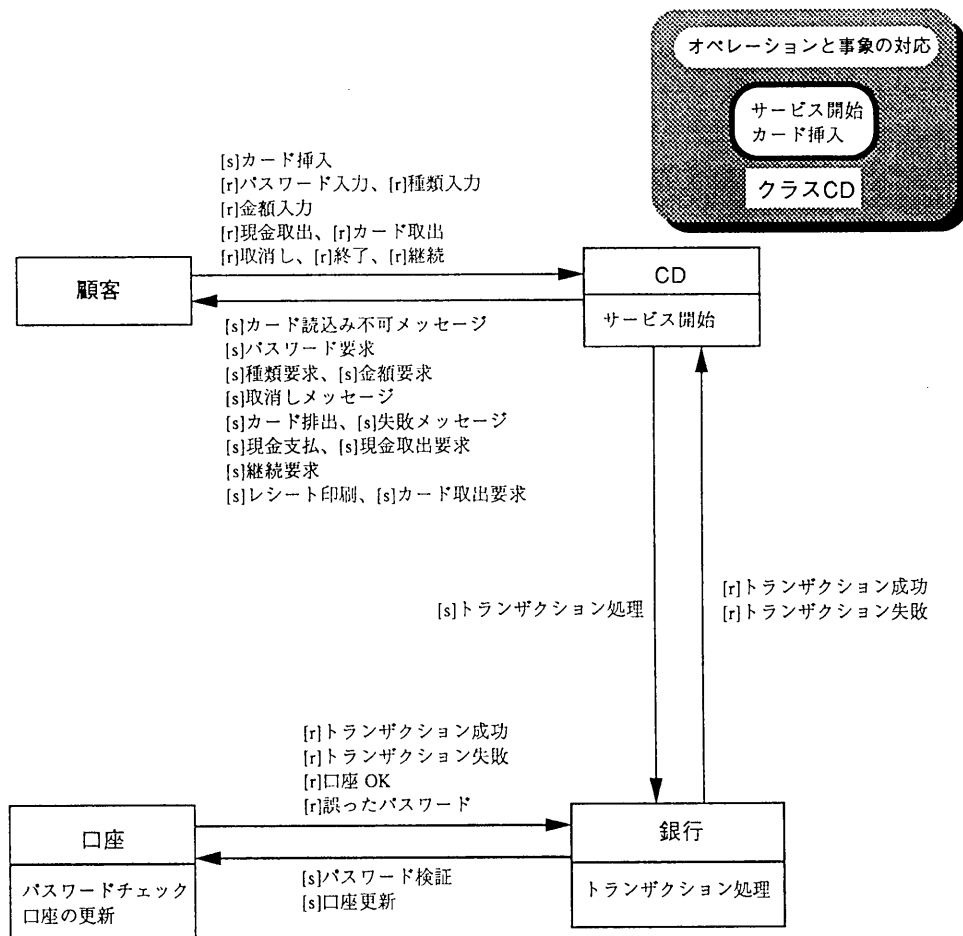


図 10: 送信事象からオペレーションを明らかにする

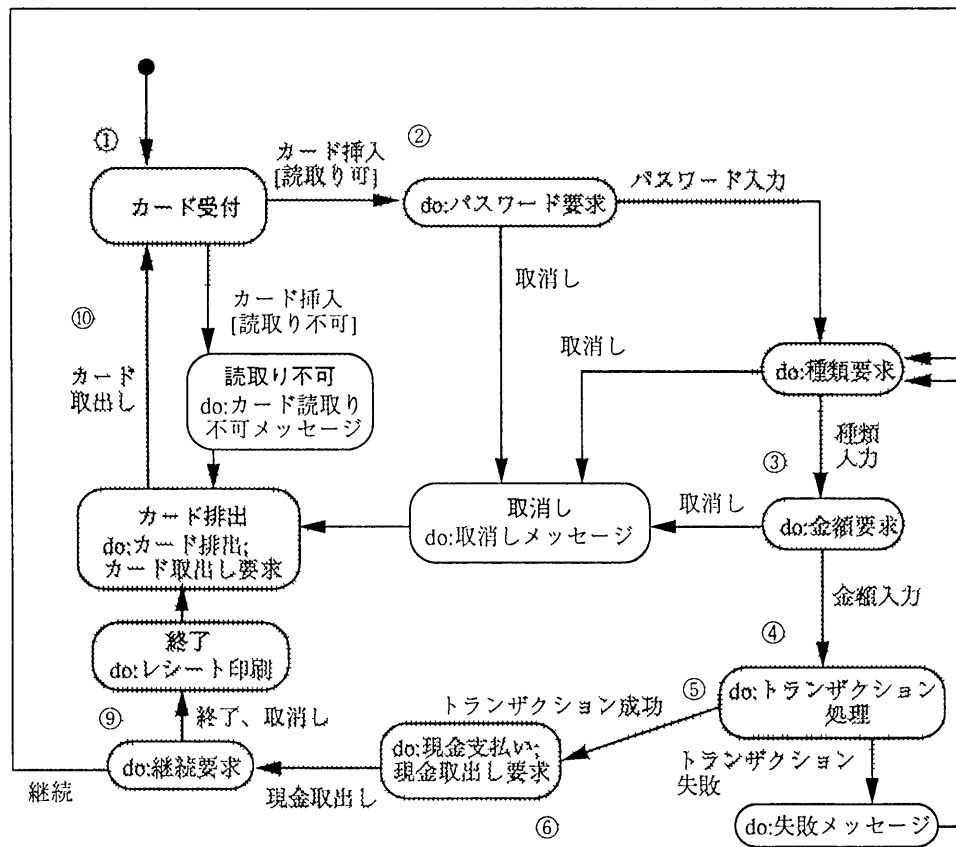


図 11: クラス CD に対する動的モデル

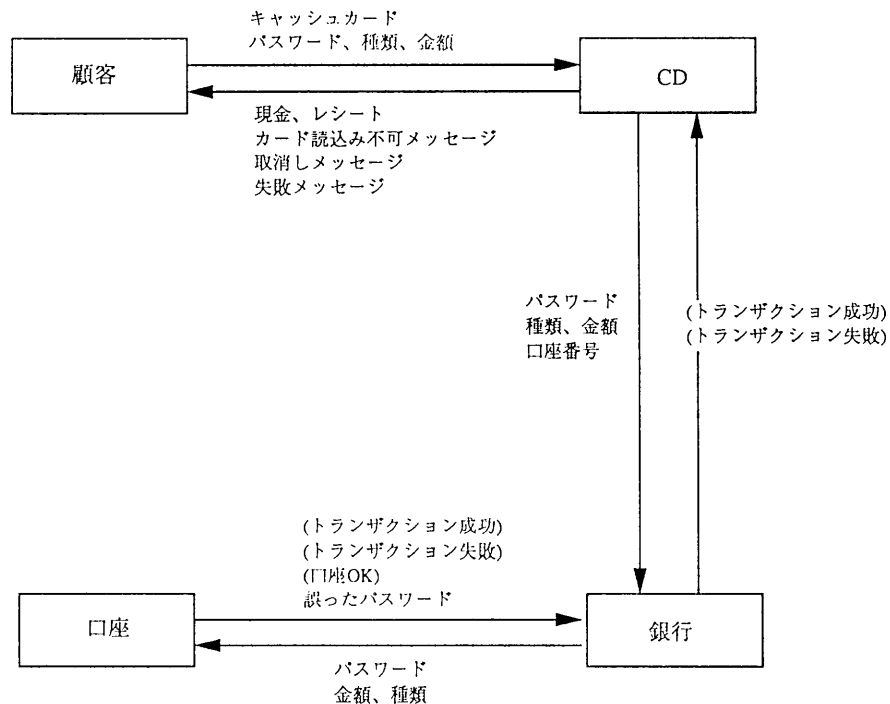


図 12: 顧客、CD、銀行、口座に対するパラメータフロー図

てクラスのどのオペレーションがサービスを行うかも動的モデルを作成するまえに確認してある。つまりオブジェクトモデルと動的モデルとの間には矛盾は発生しえない。

5.3 機能モデルの導出

OMT では機能モデルの生成プロセスに関しては、従来の構造化分析法に従っている。それはトップダウンに分析を行うものであり、ボトムアップに構築してきたオブジェクトモデルとの整合性が取りにくいという問題がある。しかし本手法ではすでにこの段階でオペレーションを導き出してある。このオペレーションがクラスで公開されるプロセスとしてまず定義される。

またクラスへの入力データとクラスからの出力データは、事象トレース図から簡単な変換で得られる「パラメータフロー図」という視点により調べることができる。パラメータフロー図は図 12 のように事象フロー図の事象のかわりに、クラス間のデータの流れを記述したものである。流れるデータはクラス間をやりとりされる事象にともなうパラメータである。また制御信号は (信号名) というように括弧で名前をくくって表している。

このデータはもともと事象から導き出されたものであるので、送信事象と対応するオペレーションからデータとオペレーションを対応付けることができる。これによりオペレーションの引数がわかる。また送信事象と対となる返信事象もわかっているので、オペレーションの返り値もわかる。

このパラメータフロー図からクラスごとの最上位機能モデルが書ける。それが図 13 である。これはクラス CD の機能モデルである。

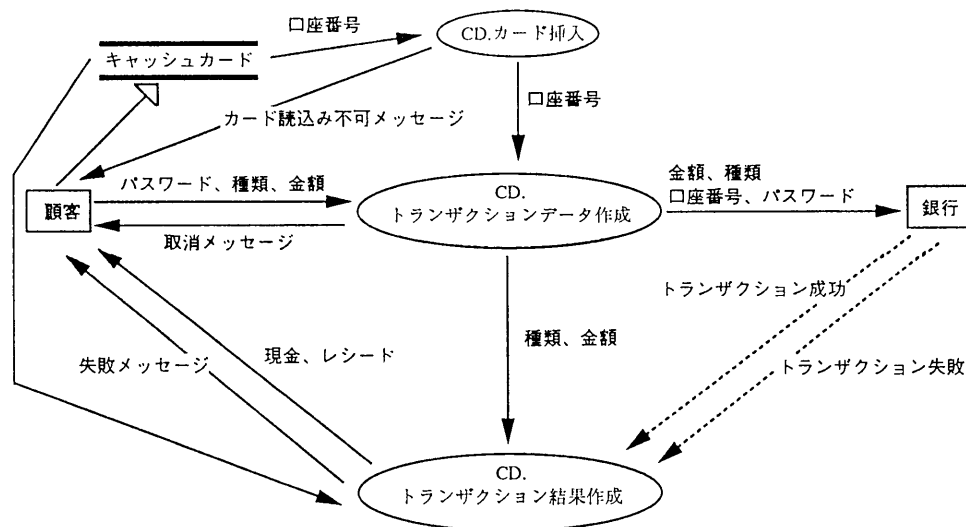


図 13: CD クラスに対する最上位の機能モデル

ここで本稿ではプロセス名の書きかたに拡張を加えた。プロセスを認識した時点でそれがどのクラスに属するかをプロセス名に記述する。例えば、

CD. カード挿入

というように「クラス名. プロセス名」というようにプロセス名を付ける。

「CD. カード挿入」がオペレーションとして公開されたプロセスである。またそれ以外のプロセスはクラス CD のプライベートなプロセスである。そしてクラス CD の入出力はパラメータフロー図のそれに従う。

同様に銀行クラスと口座クラスの最上位機能モデルを図 14、15 に示す。

この 3 つの図におけるクラスの入出力はすべてパラメータフロー図に従っている。これよりこれらの図を入出力より合成することで、キャッシュディスペンサーシステムの最上位機能モデルを得ることができる。分析はクラスをひとつずつ注目して行っても、クラス全体を見渡しながら行ってもよい。機能モデル全体から注目したいクラス部分を切り出した視点を作ることのひとつ、もしくは複数、そして機能モデル全体を分析者が自由を選択して分析することができる。複数のクラスについて分析する場合には、どのクラスに属したプロセスかを示すためにプロセス名の表記に拡張を加えた。

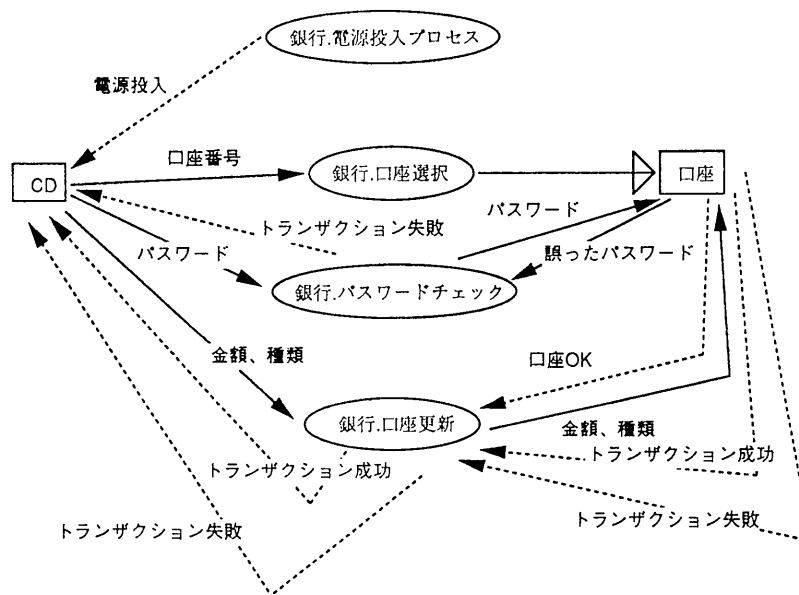


図 14: 銀行クラスに対する最上位の機能モデル

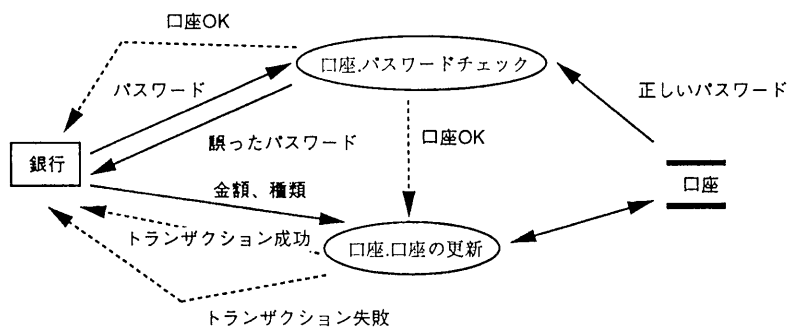


図 15: 口座クラスに対する最上位の機能モデル

6 計算機による手法の支援

本稿で提供する手法は視点を切り換えながら分析をすすめるため、その視点を紙に記述していくのは前章の分析例を見てもあきらかである。特に要素の間にある対応関係については、分析例でもそのすべてを記述は繁雑になりすぎるので省略した。実際にすべての対応関係の定義を行う場合、この程度の分析でもかなりの文書量になることがわかっている。

この問題については要素とその対応関係を維持するデータベースと、視点の切り換えを支援する視点ツールにより解決する。この章では7つの視点のデータベーススキーマをひとつひとつ定義する。視点は問題世界のある側面を満たたものであるので、それらのスキーマを重複部分から合成することで、問題世界全体のスキーマを定義できる。そこで視点ごとのスキーマを合成することで、問題世界のスキーマも定義する。

6.1 インスタンス図のスキーマ

インスタンス図は図 16 のような要素で構成される。

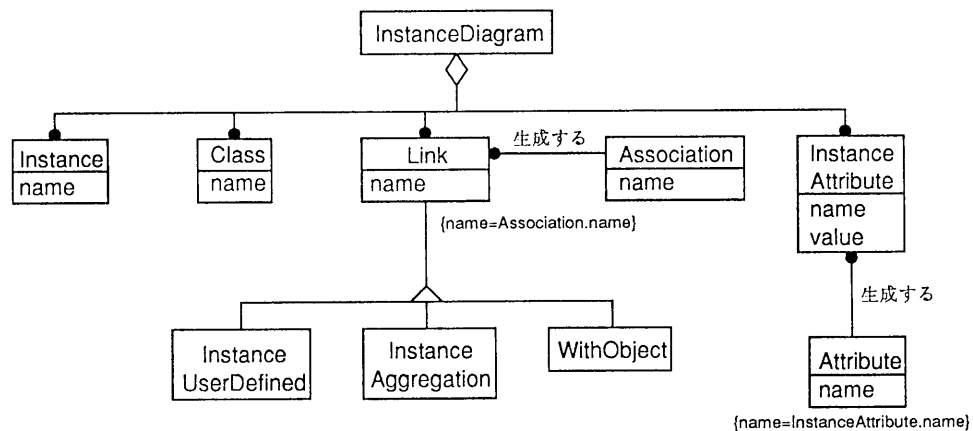


図 16: インスタンス図を構成する要素

インスタンス図はクラス (Class) から生成されたインスタンス (Instance) をリンク (Link) で関連付けたものである。リンクは参照リンクなどの一般的なリンクである分析者定義の関連 (UserDefined)、集約リンク (Aggregation)、オブジェクトを用いたリンク (WithObject) に分類される。なおこのリンクの種類は視点の切り換えでも用いられる。例えば集約リンクを視点作成のキーとして使用すれば、部品インスタンスを表示させないこともできるので、図が繁雑になるのを防ぐことができる。リンクはクラスを結ぶ関連 (Association) によって生成される。生成される時、リンクは生成した関連と同じ名前を持つ。それを表すため、図 16 ではクラス間にまたがる属性の制約を、

`{name=Association.name}`

という形で記述している。

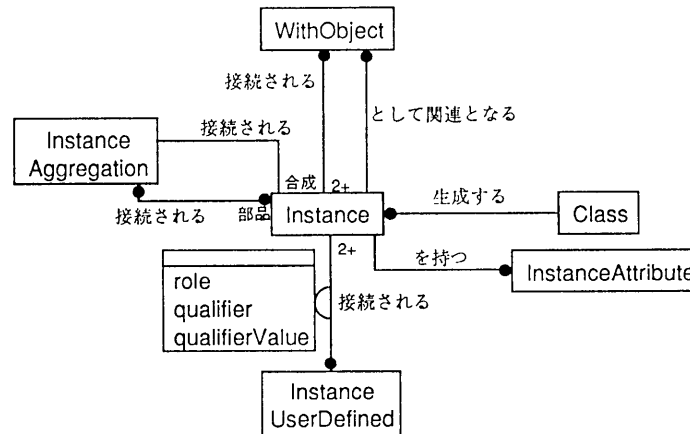


図 17: インスタンス図が表現する要素のスキーマ

またインスタンスはインスタンスの属性 (InstanceAttribute) を持つ。そしてインスタンスの属性もクラスが持つ属性 (Attribute) から生成される。これらから図 17 のようにインスタンス図が表現する要素のスキーマを表すことができる。

6.2 オブジェクトモデルのスキーマ

オブジェクトモデルは図 18 のような要素で構成される。クラスはインスタンスを生成し、クラス間には関連がはられる。関連には継承も含まれ、弁別子により二種類に分類される。またクラスには属性やクラスの操作であるオペレーションが含まれる。この構造を表わしたオブジェクトモデルが表現する要素のスキーマを図 19 に示す。

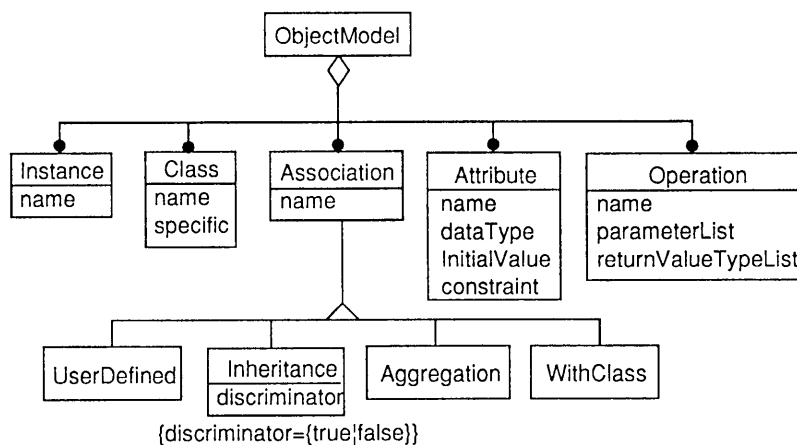


図 18: オブジェクトモデルを構成する要素

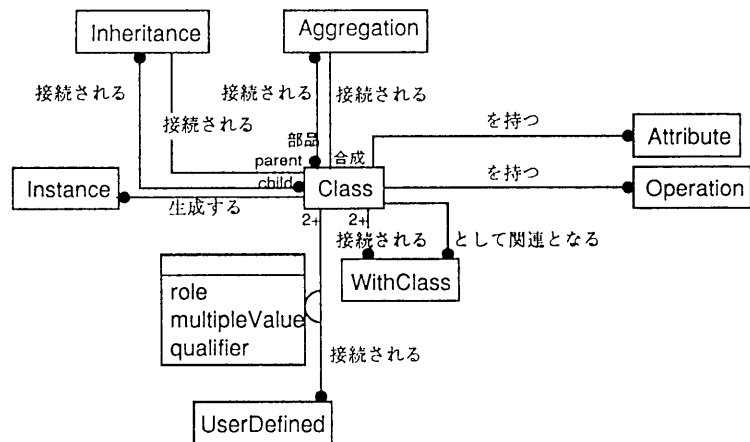


図 19: オブジェクトモデルが表現する要素のスキーマ

6.3 事象トレース図のスキーマ

事象トレース図は図 20 のような要素で構成される。まずその事象トレース図がどのようなシナリオからなるのかが必要である。そしてインスタンスとその関連、そして送信事象、受信事象を分類したステップ (Step) より事象トレース図は表現される。ステップとインスタンスの事象 (InstanceEvent)、クラスの事象 (Event)、オペレーションの関係は図 21 に示す。

これらより図 22 のように事象トレース図が表現する要素のスキーマは表わされる。

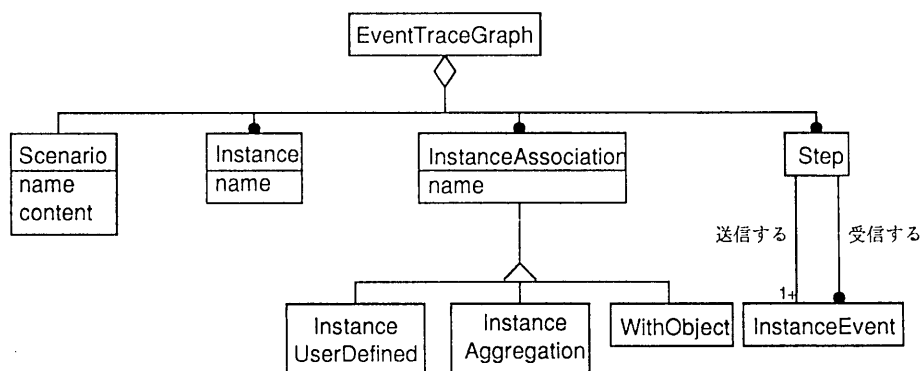


図 20: 事象トレース図を構成する要素

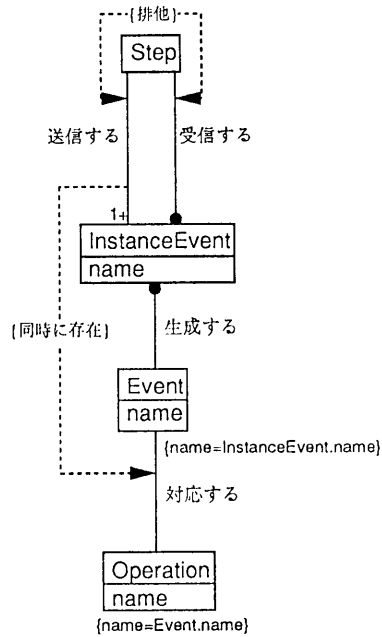


図 21: 事象とオペレーションの関係

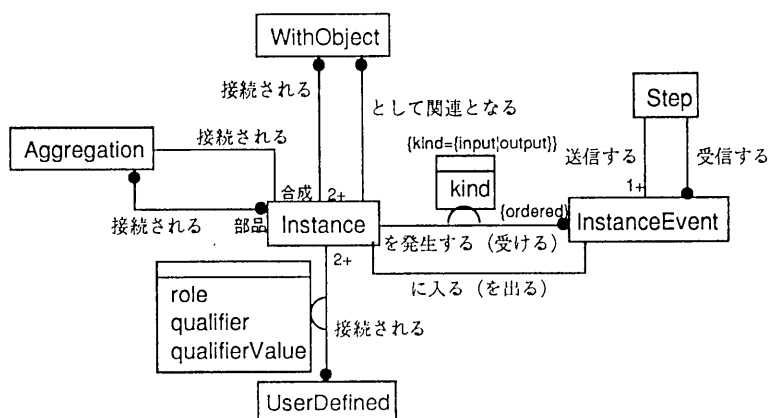


図 22: 事象トレース図が表現する要素のスキーマ

6.4 事象フロー図のスキーマ

事象フロー図は図 23 のような要素で表わされる。それは事象トレース図で表わされたインスタンスをグループ化しているクラス、そして関連と送受信する事象である。この構造は図 24 に示す。

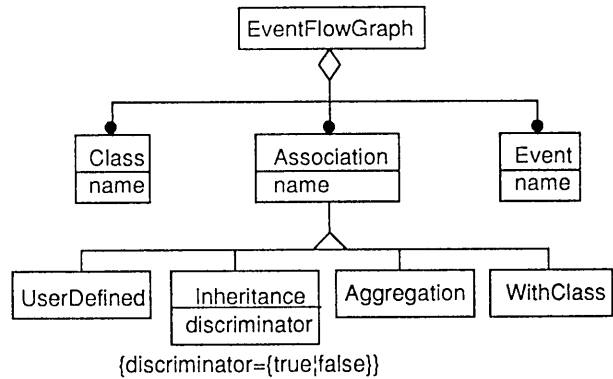


図 23: 事象フロー図を構成する要素

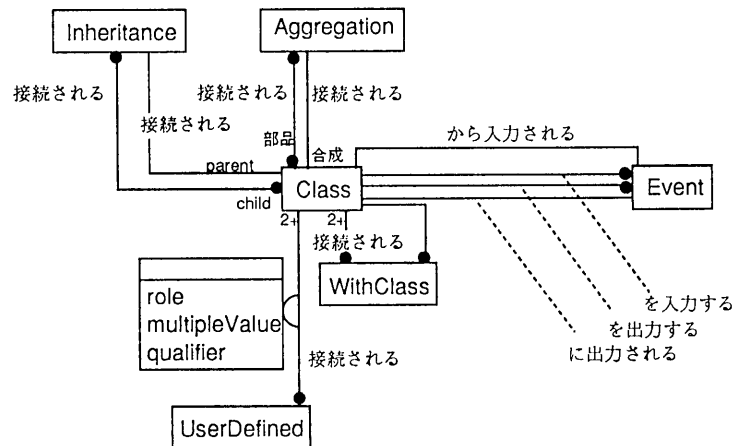


図 24: 事象フロー図が表現する要素のスキーマ

6.5 動的モデルのスキーマ

動的モデルの構成要素を図 25 に示す。クラスはひとつの動的モデルを持ち、そのクラスが生成するインスタンスの状態とその遷移、動作などの事象を表す。そして事象は属性リストを持つことがある。この構造を図 26 で示す。

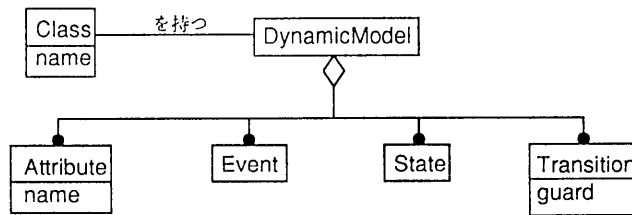


図 25: 動的モデルを構成する要素

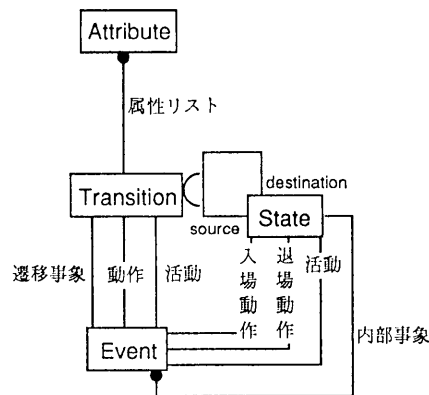


図 26: 動的モデルが表現する要素のスキーマ

6.6 パラメータフロー図のスキーマ

パラメータフロー図の構成要素を図 27 に示す。パラメータフロー図ではクラスや関連の他に流れるパラメータを表現する。このパラメータのデータ (Data) は機能モデルのデータフローに、信号 (Control) は制御フローに対応する。クラスにパラメータは入力、もしくはクラスから出力され、そのパラメータは別のクラスから出力されたもの、もしくは入力される。それを表わしたのが図 28 である。

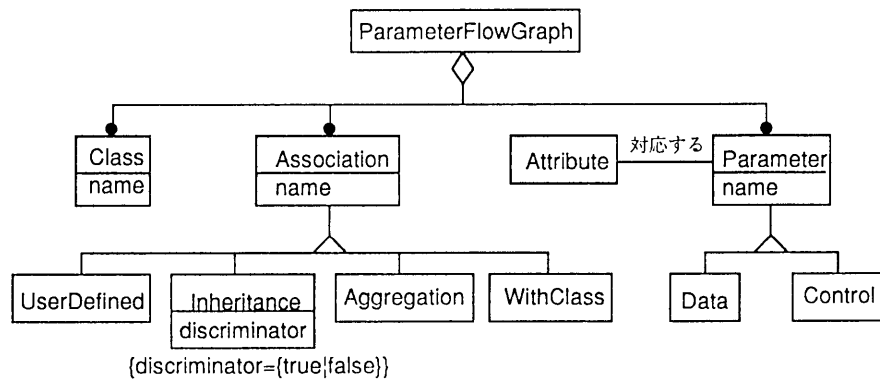


図 27: パラメータフロー図を構成する要素

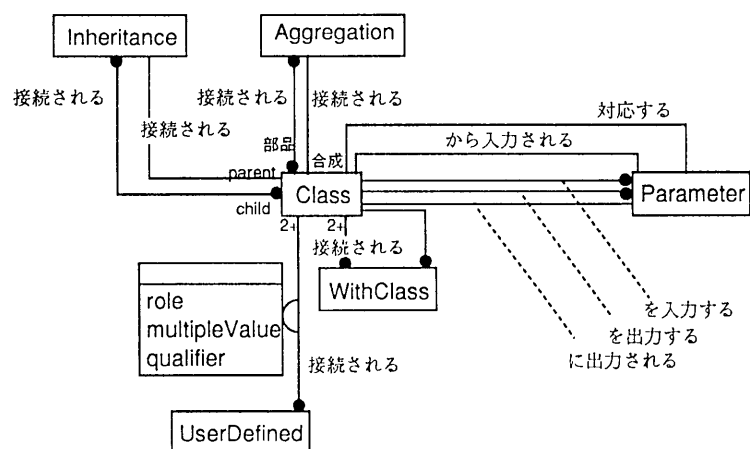


図 28: パラメータフロー図が表現する要素のスキーマ

6.7 機能モデルのスキーマ

機能モデルの構成要素を図 29 に示す。機能モデルはクラスに含まれるプロセス、プロセス間を流れるデータ、信号、そして属性、ときにはクラスとも対応するデータストアとこれもまたクラスと対応することがあるアクターにより構成される。またプロセスを詳細化することでもた機能モデルは階層化する。

これより機能モデルが表現する要素のスキーマを図 30 に示す。

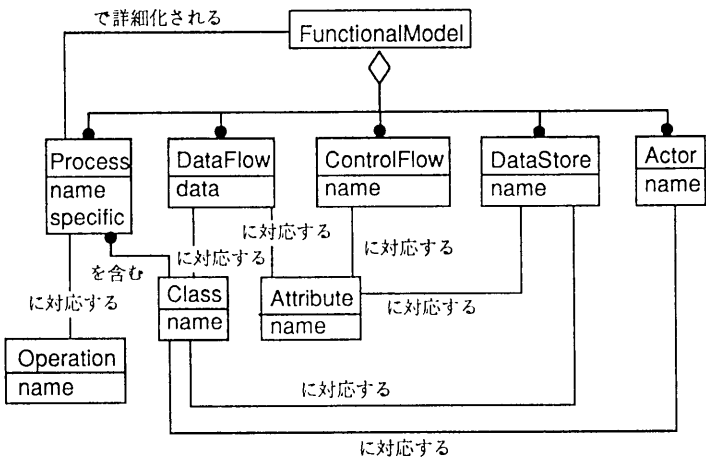


図 29: 機能モデルを構成する要素

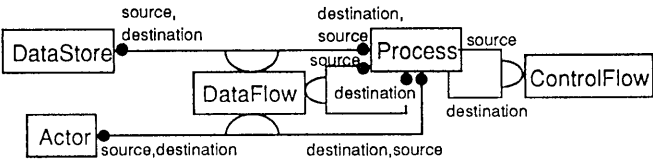


図 30: 機能モデルが表現する要素のスキーマ

6.8 スキーマの合成

これまでひとつひとつの視点に対して、その要素の関連を見てきた。その視点は問題世界の一部の要素について見たものである。ここでは今までひとつひとつの視点に対して定義されたスキーマを合成することで、問題世界全体を表現するスキーマを定義する。

合成にはまずひとつひとつのスキーマで重複部分を重ねあわせる。そしてオペレーションとプロセスのように対応する要素について「対応する」という新たな関連をはる。これにより図 31 が得られる。

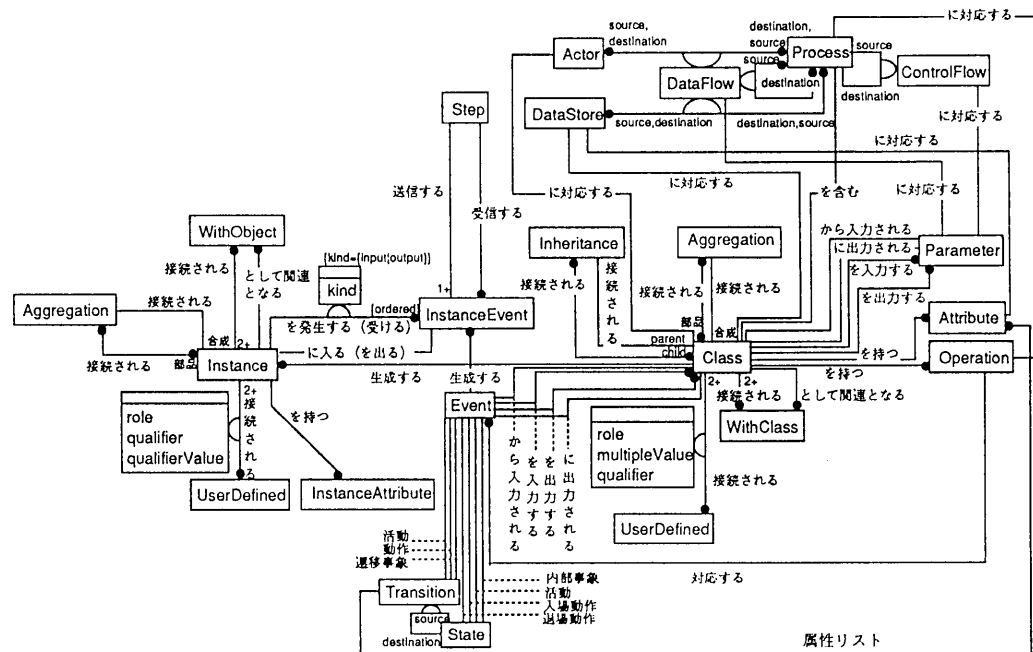


図 31: 問題世界を表現する要素のスキーマ

次にこの要素スキーマより視点を切り出す構造を定義する。これには視点が参照する要素へ視点クラスから集約関連を結ぶだけである。これは図 16 からの視点を構成する要素により定義されている。

また OMT の視点は図 32 のように形成される。

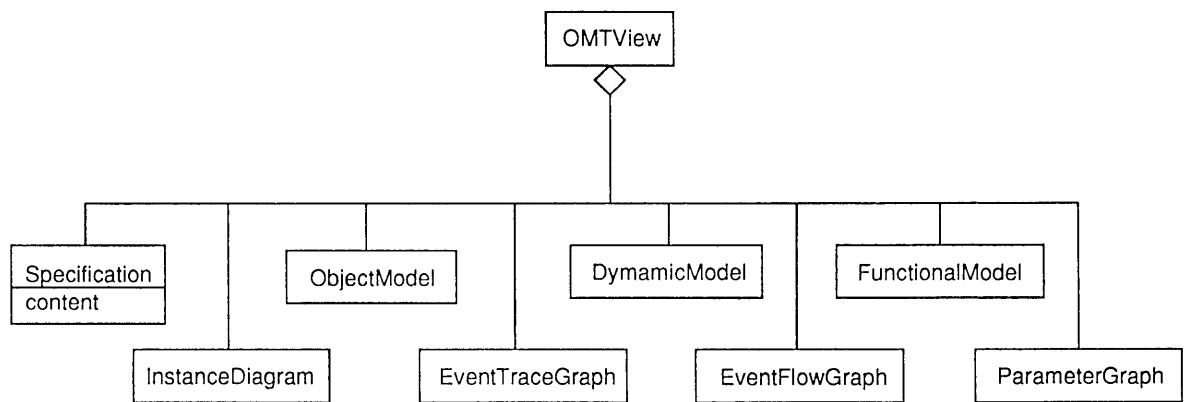


図 32: OMT の視点

7 PCTE への実装

PCTE(Portable Common Tool Environment) はソフトウェアエンジニアリングをサポートする環境の一部として、そこで扱われるデータの共通的な処理機能群に対する機械独立なアクセスを提供する。すなわちツールの可搬性を支援する解放型リポジトリのインターフェース標準である。

本研究では PCTE のソフトウェアパッケージである Emeraude 社の PCTE V12 を使用して、前章で定義したスキーマを PCTE のオブジェクトベースに実装を行った。

7.1 PCTE のオブジェクトベース

PCTE システムの中核となるのはオブジェクトベースである。そのデータモデルは、いわゆる ERA モデルの拡張である。オブジェクトがエンティティであり、関連にあたるものが双対のリンクである。このリンクにはカテゴリがあり、これによりエンティティの関係を表す。リンクには 4 つの種類があり、その中心の三角部分に表されている。リンクの種類を表 2 に示す。

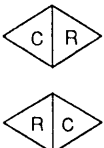
	関係型の性質
	視点オブジェクトが終点オブジェクトを要素とする複合構造を示す。終点オブジェクト側からこの関係を消去することはできない。
	複合構造の構成要素が明示的に構成主を参照することを示す。composition リンクの終点側からこの関係を消去することができるのは、そのオブジェクトに流入する他の composition リンクが存在する場合に限られる。
	一方的な参照関係を示す。参照しているのは視点オブジェクトであり、終点オブジェクトからこの関係を消去できない。
	対等な参照関係を示す。参照関係に主従はなく、終点側、視点側のいずれからでもこの関係を消去できる。

表 2: 関係型の分類 [鯨坂 93]

7.2 オブジェクトベースへの定義

要素のスキーマは図 33 のように拡張 ER 図に変換した。
そしてこの要素のスキーマより OMT の視点を切り出すスキーマは図 34 のようになる。

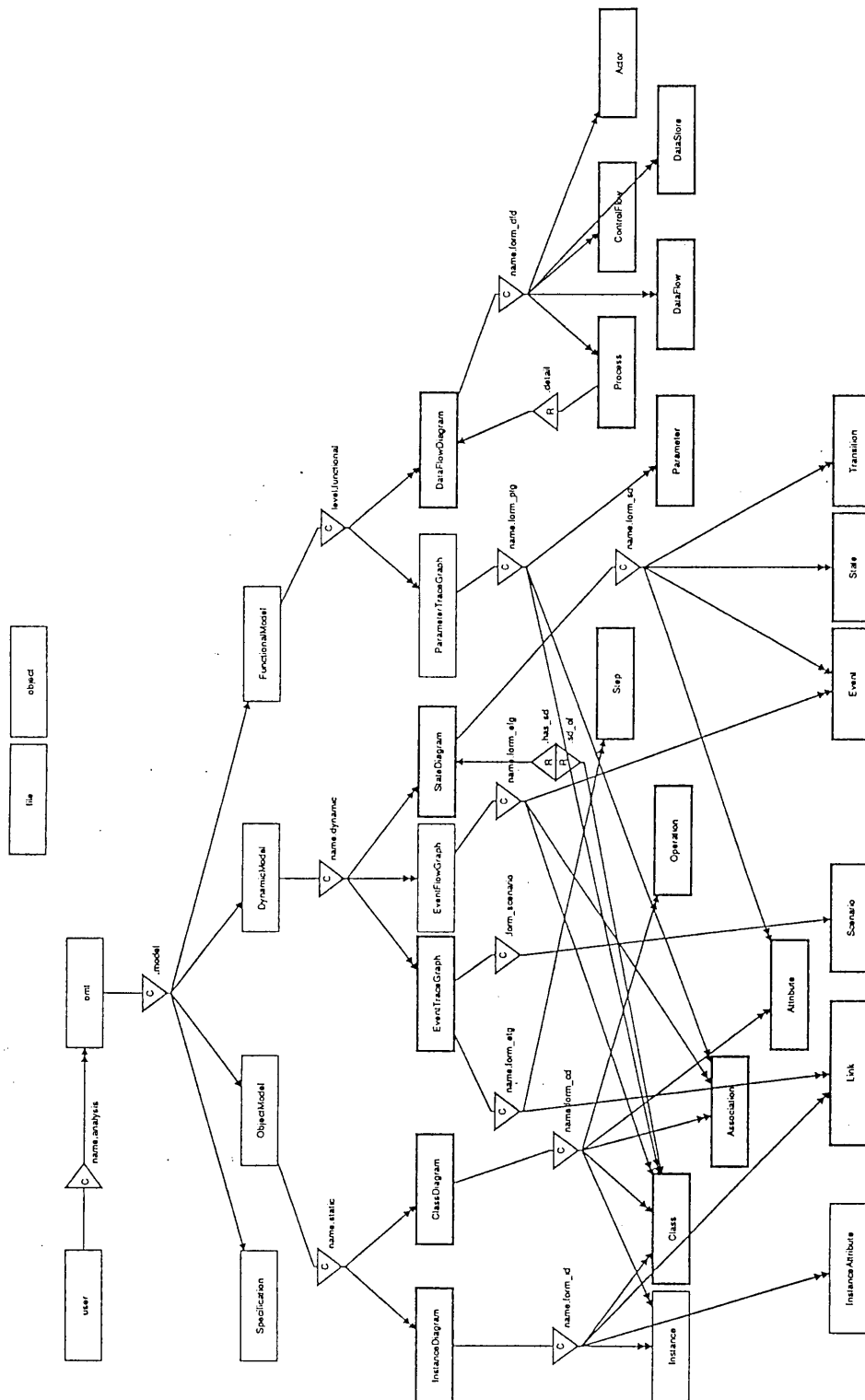


図 34: OMT を構成する要素

8 手法の評価

現在、より大規模な事例として [倉谷 93] で扱われている「実在する会社の生産管理部門の業務」を本稿で提案した手法により分析をすすめている。その結果、次の点が明らかになってきている。

- 分析は広く浅く行うのではなく、問題の中核をなす部分について特に注目する。
- 中核部分から周辺部分に分析の目を移していくにつれ、浅い分析にとどめておく。
- クラス構造が複雑になる時には大きな構造をまず認識し、それを集約関係を用いてその構造を構成する小さい構造を認識するのが有効である。
- オブジェクトモデルと現実世界との対応性の良さは、事象トレース図の事象の流れが現実世界での処理手順と一致することで判断できる。
- クラス間での事象の流れの複雑さが増すごとに、オペレーションを基礎とした機能モデルの導出の必要性が増す。

これらの結果より、本手法が教科書に記載されている例題のような閉じた世界を対象にするだけでなく、実際の業務世界をも対象にできると考えている。

9 おわりに

本稿では分析すべき問題世界を OMT が表現しようとしている要素に分解し、どの部分に集中して分析を行うかを視点と定義した。そしてひとつひとつの視点の構造をとらえ、それらの間の共通性をとらえることで問題世界の表現をモデルではなく、要素の集合である視点ととらえた。それにより連続的に分析をすすめることができ、OMT の 3 つのモデル間の間に存在していた矛盾を早期に排除できるようになった。

また要素間の対応を取るために大量の文書が必要となってしまう問題についてはデータベーススキーマを定義し、PCTE 上にそれを実装することで計算機支援を行える環境を整えた。

しかしこのスキーマはまだ完全には OMT のモデルを表現できていない。動的モデルの階層化などがその代表である。また PCTE 上のオブジェクトベースはクラスや関連間にはたらく制約を定義することはできない。これらの管理はそのスキーマの上にあるツールにより行われる。しかし新たな視点としてツールを加えるにあたって、スキーマ情報がツールに委ねられるのは好ましくない。そこで制約の管理をリポジトリの中で行える、オブジェクトベースの上にあるサービス層が必要である [鯉坂 93]。具体的には生成規則を持ったオブジェクトの生成関数である。これらの問題を解決した後、視点ツールをこのデータベーススキーマ上にのせることで、OMT を支援する CASE 環境を実現することができる。

Appendix

A 「キャッシュディスペンサーシステム」に関する問題記述

銀行には何台かのキャッシュディスペンサーが設置されている。銀行に口座を持つ顧客はその口座に対するキャッシュカードを銀行から発行してもらうことで、窓口からだけでなく、キャッシュディスペンサーからも現金の払い戻しや預金ができるようになる。顧客はカードごとに4桁のパスワードを持ち、それによってセキュリティ対策をしている。

また口座には少なくとも当座と普通を含むさまざまな種別がある。そして口座ひとつひとつに借入限度額が決まっている。

参考文献

- [Rumbaugh92] James Rumbaugh 他著, 羽生田栄一監訳. オブジェクト指向方法論 OMT. トッパン, 1992.
- [西村 93] 西村一彦, 本位田真一. 複合ビューポイントに基づく仕様化プロセスの分析. 情報処理学会論文誌, 34, 5, 1993.
- [酒匂 93] 酒匂寛. オブジェクト構造モデルと動的モデルの統合に関する考察. Software Symposium '93, 1993.
- [Nuseibeh93] Bashar Nuseibeh, Jeff Kramer, Anthony Finkelstein. Expressing the Relationships Between Multiple Views in Requirements Specification. Proceedings of the 15th International Conference on the Software Engineering, IEEE, May, 1993
- [Monarchi92] D. E. Monarchi and G. I. Puhr. A Research Typology for Object-Oriented Analysis and Design. Commun. ACM, 35, 9, 1992.
- [満田 93] 満田成紀, 鯉坂恒夫, 松本吉弘. オブジェクト指向分析/設計のためのソフトウェア基底モデリング, 第12回ソフトウェア研究会資料, 日本ソフトウェア科学会, 4, 1992.
- [ECMA90] ECMA. Portable Common Tool Environment (PCTE) Abstract Specification, Standard ECMA 149, 1990.
- [鯉坂 93] 鯉坂恒夫, 沢田篤史, 満田成紀. Emeraude PCTE, コンピュータソフトウェア, vol.10, No.2, Mar., 1993.
- [倉谷 93] 倉谷祥久, 門脇千恵, 西山雄, 落水浩一郎. 構造化分析の事例研究 (生産管理システム), Jaist Research Report, 9, 1993.