

ボトムアップな MVC アーキテクチャモデルの概念的な構成方法に関する考察

大木 幹雄

日本工業大学 工学部 情報工学科

〒345-8501 埼玉県南埼玉郡宮代町学園台 4-1

ohki@nit.ac.jp

あらまし

オブジェクト指向分析設計において、概念や MVC モデルに沿った基本クラス構造を抽出するには、経験と訓練が必要であるとされている。しかしながら、現在のオブジェクト指向方法論は、具体的な分析設計の判断基準を提供するまでにいたっていない。そこで、初心者であっても理解可能な分析・設計の視点や判断基準を CASE ツールの機能として盛り込むことを目標として、オブジェクト指向のもつ基本的な特徴からモデリングに関するいくつかの判断基準を明らかにした。概念モデルの判断基準については、その有効性が確認できているが、本論では次のステップとして、MVC モデルに View クラスの抽出に関する判断基準について提案する。

キーワード オブジェクト指向分析設計, 判断基準, 概念モデル, MVC モデル

A Consideration of the Modeling Criteria for MVC Model Building by Bottom-up approach

Mikio Ohki

Nippon Institute of Technology

4-1 Gakuendai Miyashiro Minami-Saitama ,345-8501

Abstract

To figure out Concepts or Primary Class Structures, Object-Oriented Analyst/Designer are required various experience and training. However, the present Object-Oriented Methodologies do not offer enough concrete Modeling Criterias. This paper describes Modeling Criterias, which are derived from fundamental features of Object-Oriented mechanism. Preliminary experiment clarifies that some those shows certain efficient. This paper proposes, for the next step, View Class Modeling Criterias used in MVC Class designing .

Key words Object-Oriented Analysis /Design , Modeling Criteria, Conceptual Model, MVC Model

1. はじめに

オブジェクト指向分析設計による情報システム開発の機会が増えるにつれ、オブジェクト指向分析設計を支援する CASE ツールの重要性が増している。しかしながら、現在のオブジェクト指向分析設計を支援する CASE ツールの多くは、各種の分析設計図の記述や設計内容の検証に関して簡便な機能を提供しているものの、分析設計をどのような視点で行うか、あるいはクラスをどのように抽出するか等の設計者の認識方法を支援する機能を盛り込むまでにはいたっていない。

例えばオブジェクト指向分析フェーズの第1段階である概念モデルの構築や設計フェーズにおけるクラス設計は、オブジェクト指向分析設計の初心者にとって、非常に難しい作業とされている。これは分析設計内容を形式的に表現する以前に、どのような視点で概念を抽出・整理すべきか、どのような基準で機能をクラスに分担させるべきかの判断基準を初心者がもたないためと考えられる。決して CASE ツールの機能や表記法が理解できていないためではない。

残念ながら、概念モデルの構築やクラス設計に関して、初心者にも理解できるような納得のゆく、分析設計の視点や判断基準を、現在のオブジェクト指向分析設計方法論は提供していない。もちろん、分析設計はケースバイケースによって異なる点が多いため、指針程度しか与えられないことは理解できる。しかし、例えば UML では概念モデルの構築において、指針として与えているものは「モデル作成者にとって意味のある概念を抽出し、図で説明すること」でしかない[1]。このような指針がいかにも具体性に欠ける。例えば、図1の(a)で示すように、「商品」の概念を取り上げたとき、一般に (a) より (b) の方が良い概念モデルの抽出であるとしている。これは「商品」のインスタンスが売り切れたときであっても、その商品に関する情報までも削除されない点で良いとしている。

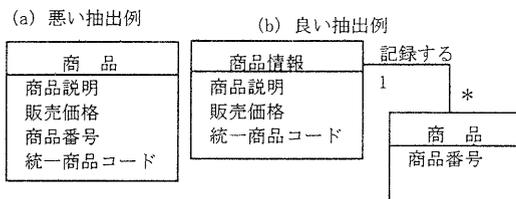


図1 指針のない「意味ある概念」の抽出のよし悪し例

しかし、本来なら最初から意味ある概念として「商品」と「商品情報」が抽出されるべきであって、売り切れると商品情報が削除されるからとい

った利用方法までも先回りして考慮し、「商品」概念を2つの概念として抽出せよとすることには無理がある。特に初心者にはこのような指針をほとんど活用できない。UMLではユースケース分析で得た仕様から名詞句に着目して概念を抽出する方法を推奨しているが、このような方法を適用したとしても、図1で示すような2つの概念をいかに抽出するかについて、具体的な判断基準が示されているわけではない。

本論では、このような現状から、業務を分析するとき比較的容易に得られる業務上のデータ群、およびビジネス活動のきっかけとなるイベント群にもとづいて、概念モデルを抽出するための判断基準について論じる。

次いで、クラスのもつ基本的な役割をオブジェクト指向の基本機構に立ち戻って考察し、MVC (Model-View-Controller) モデルにしたがったクラス構造を構築するときの属性とイベントの分類に関する判断基準について論じる。最後に、その有効性と課題について述べる。

2. 研究の目標

(1) 設計プロセス自動化に必要な機構の究明

本研究の発端は、経験と訓練を要する分析設計過程において、初心者であっても一定の成果が導き出せるような支援ツールの可能性を探ることにあつた。設計成果物は、設計内容の構成要素が織りなす協調動作から生み出されるとの考えから、オブジェクト指向機構に双方向コミュニケーション機能を追加した機構を提案し、その分析設計プロセスをそれらの協調動作で再現することを試みてきた [2]。現在、このような機構の一部は試作され、実験に用いられている。

(2) プロセスの設計に必要な判断基準の究明

協調動作機構の有効性を事前評価するため、具体的な事例に対して適用を重ねるにつれ、分析設計における人間の判断基準を明確化する必要性が生じてきた。そこで、協調機構開発と並行する目標として、分析設計における具体的な判断基準の明確化を設定している。

実際、UML等の方法論では、概念の抽出は発見的に行われるべきものとしており、初心者にも理解できる具体的な判断基準が存在しない。発見を容易にするため、ユースケースを洗い出し、活用するにしても、一般的な判断基準が存在しない以上、事態の大きな改善に期待はできない。

筆者は分析設計の判断基準は経験的に獲得されるものでなく、オブジェクト指向機構をもつ基本的な特徴から、必然的に導き出されるべきものとの考えから、判断基準の明らかにすることを試みた。

3. 基本的なアプローチ方法

オブジェクト指向がもつ基本的な特徴から、モデリングに関する具体的な判断基準を導き出すためには、まずクラスをどのように捉えるかが、その出発点になる。

機械的に判断基準を適用して、概念モデルやクラス構造を抽出するには、KJ法と同様に、まず概念やクラスがもつ構成要素を集め、その中から必要な要素を取捨選択し分類する基準を定める方が、概念やクラスに関する発見方法の基準を定めるより容易である。すなわち、クラスとは、取捨選択・分類された構成要素を格納する容器であると捉えて判断基準を見出す方が、直接、概念やクラスを発見する判断基準を見出すより容易である。

そこで、筆者はクラスを構成要素が格納された容器として捉え、ボトムアップにそれらを構成してゆくための判断基準を明らかにすることを、基本的なアプローチとして採用した。言い換えると、初心者でも比較的収集が容易な業務上のデータ群や業務処理のきっかけとなるイベント群を伝票や帳票の書式分析から洗い出し、それらから妥当なものを取捨選択し、分類することによって、概念モデルやクラス構造を導き出すアプローチを採用した。これと類似するアプローチをとる方法論として、他にもDATARUNがある[3]。

分類された概念候補やクラス候補が、どのような名称をもつかを決定する命名作業は、決定後に、既存の概念用語辞書の中から最もふさわしいものを選択して行うものとする。

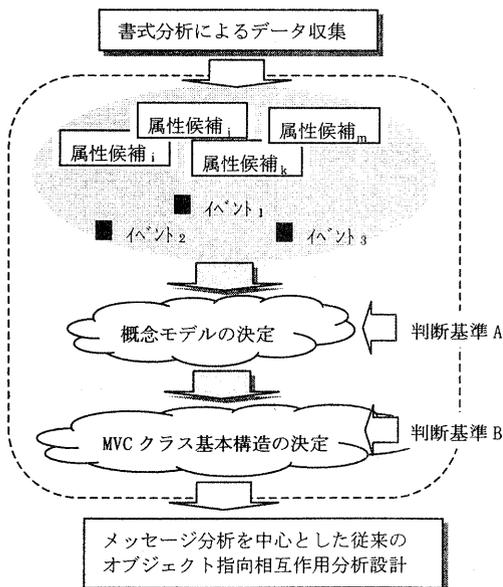


図2 ボトムアップなクラス構造決定のアプローチ

このようなアプローチをもとに、図2の破線内で示すように、概念モデルやMVCクラス基本構造を導き出す判断基準を明らかにすることにする。ただし、アプリケーション分野毎に分析設計内容の表記規則にしたがって判断する部分は、従来のUML等の方法論を用いて補完するものとする。そのため、クラスの基本構造決定以後の相互作用分析から各クラスがもつ操作を決定する手順は、両者とも同じになる。

4. 分析設計における視点とは

前述のように書式分析で収集したデータ群（これらはクラスの属性候補に見ることができる）と業務処理を始めるきっかけとなるイベント群から、概念やクラスの構成要素となる属性候補やイベント群を選択・分類し、ボトムアップにそれらを容器としての概念やクラスに格納してゆくには、前述のアプローチ法を具体化した次のような視点が必要になってくる。

- (視点1) 概念・クラスは属性を格納する実体
- (視点2) 概念・クラスはイベントを受取る実体
- (視点3) 概念・クラスは状態変化を通知する実体

このような視点は、あらためて述べるまでもないがオブジェクト指向の基本的な動作機構がもつ次のような特徴と対応している(図3参照)。

- (1) 動的なインスタンス生成
- (2) 外部からのイベントによる手続き駆動
- (3) インスタンス内部での非同期的なイベント発生による手続き駆動

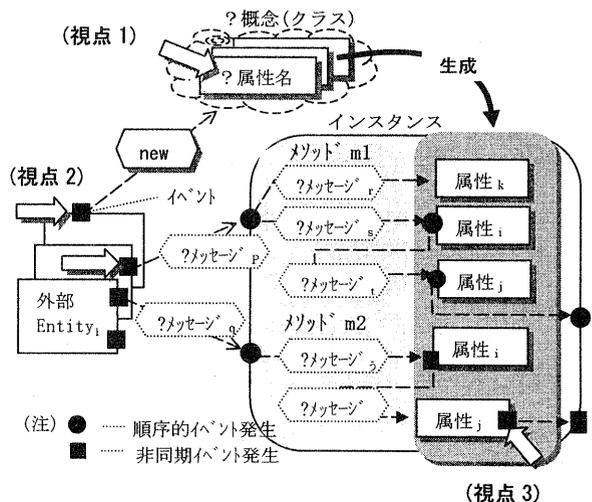


図3 オブジェクト指向の基本動作機構と着目すべき視点

もちろん、一般的なオブジェクト指向の特徴として、カプセル化やクラス継承、多相性等があげ

られるが、これらの特徴はクラス構造が構築された後にはじめて顕在化する特徴であり、概念やクラスそのものが抽出されていない状況では、二義的な役割しかもたないものとして位置付ける。

5. 判断基準の導出

3つの視点を原点に立ち返って検討してみると、データ項目とイベントから、図2で示したような概念モデルの決定とMVCクラス基本構造の決定を行うに必要な判断基準を導出することができる。なお、以後では、基本データとは洗い出されたデータ項目のうち、他から導出できないデータ項目を意味するものとする。また属性候補とは、それらを分類し、概念あるいはクラスに格納することが可能なものを指すものとする。

(1) 概念モデルの決定に関する基準

クラスは属性を格納する実体であり、クラスから生成されるインスタンスの属性値は次のような性質をもつ。

① インスタンス生成の同一時刻性

インスタンスのもつ初期値は、インスタンス生成時に同時に決定される。

② インスタンス属性値の単値性

インスタンスのもつ属性値は、すべて単値である。すなわち、インスタンスは一組の属性値の組をもつ。

これらの性質から、基本データから属性候補を選択・分類する判断基準が得られる。

【基準 A1】初期値の決定時点が同一な基本データは同じ概念やクラスの属性候補である可能性が高い。

初期値の決定時点とは、概念やクラスが未抽出な状況では、同じビジネスイベントの発生時点と解釈してもよい。

【基準 A2】多値属性は、同じが概念やクラスに格納できない。

【基準 A3】状況に依存しない属性候補は経書関係の上位に位置する。

このような判断基準を用いて具体的に概念モデルを抽出した事例と有効性の評価は、すでに報告しているが[3][4]、以下では判断基準を図1で示した概念抽出に適用した事例を示す。

i) 売上伝票や仕入れ伝票から次のようなデータ項目を洗い出す。

- ・商品説明
- ・販売価格
- ・商品仕入番号
- ・統一商品コード

ii) それらが決定される時点进行分析する。すると

それぞれのデータ項目の値が決定する時点(あるいはビジネスイベントの発生時点) t は一般に次のようになる。

[基本データ名]	[初期値の決定時点 t]
・商品説明	: t =取扱商品登録時
・販売価格	: t =取扱商品登録時
・商品仕入番号	: t =商品仕入時
・統一商品コード	: t =取扱商品登録時

インスタンス生成の同一時刻性の判断基準にしたがって、これらのデータ項目を分類すると、図1(b)に示すような2つの概念候補の属性候補に分類され格納される。分類後の属性候補は何かから実体の属性(=性質)を特徴付けるあることから、それらの性質を共通してもつ実体の名称を探し、“商品”と“商品情報”を命名する。

ここでは、もはや発見的に意味ある概念を見つけ出すといった観念的な指針は存在しない。ただ、値の決定される時点 t による分類操作が存在するだけである。

このほかにも、基準 A1 と基準 A2 を組み合わせることにより、注文と注文明細の関係のような識別子従属関係も自然に導くことができる。

例えば、次のようなデータ項目が注文伝票の書式分析から得られたとする。それぞれの初期値の決定時点 t 、およびそのとき記述されるデータ値の数を示すと、次のようになる。ここで*記号は、多値、すなわち同時に複数の値が記述されることを示している。

[基本データ名]	[初期値の決定時点 t]
・注文番号	: t =注文受付時
・日付	: t =注文受付時
・明細番号*	: t =注文受付時
・商品番号*	: t =商品登録時
・明細注文数*	: t =注文受付時
・商品名*	: t =商品登録時
・商品単価*	: t =商品登録時

これに基準 A1 を適用すると、以下の2つのグループに分類される。

《グループ1》

・注文番号	: t =注文受付時
・日付	: t =注文受付時
・明細番号*	: t =注文受付時
・明細注文数*	: t =注文受付時

《グループ2》

・商品番号*	: t =商品登録時
・商品名*	: t =商品登録時
・商品単価*	: t =商品登録時

グループ1は、多値属性候補が混在しているので、基準A2を適用すると、さらに分類することができる。またグループ2はすべての属性候補が多値であるので、同一グループが繰り返す形式に変形できる。このようにして最終的に以下の3つのグループに分類される。

《グループ1-1》	
・注文番号	: t=注文受付時
・日付	: t=注文受付時
《グループ1-2》*	
・明細番号	: t=注文受付時
・明細注文数	: t=注文受付時
《グループ2》*	
・商品番号	: t=商品登録時
・商品名	: t=商品登録時
・商品単価	: t=商品登録時

それぞれのグループに属性候補を参考にしながら名称を付けると、図4のような人手によるものと同様な概念モデルの構造を導くことができる。このとき、基準A1では、1つのグループであったが、基準A2によって強制的に分離されたグループ間は識別子従属関係に置き換えられることに注意する必要がある。

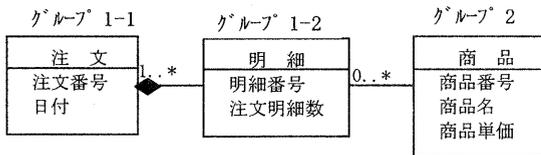


図4 基本データを基準にしたがって抽出した概念モデル

(2) MVCクラス基本構造の決定に関する基準

概念モデルが決定されると、概念モデルをMVCモデルのModelクラス候補として位置付け、次にVCクラス基本構造を決定する。これらの基本構造を決定する基準を考察するに先立ち、図3で示した基本的な動作機構を、一旦図5のように整理してみる。

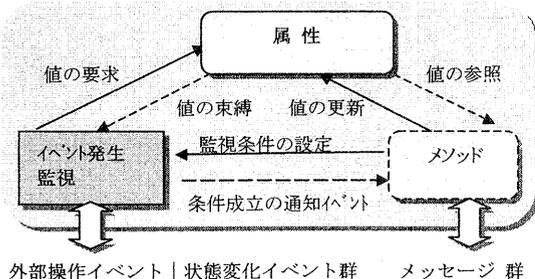


図5 基本動作機構の概念的構成

本来、クラスは図5で示す基本動作をすべて内蔵しているが、MVCモデルではそれらの動作を役割分担させ、その協調動作によってシステムの柔軟性を実現することが考え方の根底にある。

そこでMVCクラスの役割分担方法を追求してゆくと、図6で示すように、図5で示した基本動作をそれぞれペアにして、分離独立させたものと考えることができる。すなわちModelクラスとは属性と属性に対するメソッド(MVCの骨格を決める段階では更新、削除、参照の操作だけと考えてよい)をペアでもつクラスであり、Viewクラスとは、Modelクラスのもつ属性から導出されるような導出属性やインタフェースのために一時的に値を格納する属性、およびそれらの属性値を変化させるイベントの監視条件をペアにしたクラスと考えることができる。同様に、Controllerクラスとは、イベント監視条件とそれによって起動するメソッド(主としてModelクラスおよびViewクラスへのメッセージ送信を行う)をペアにしたクラスと考えることができる。ただし、図6では、一連のメソッド群は、MVC基本クラス構造が明らかになった後に、決定されるものとして破線で示してある。

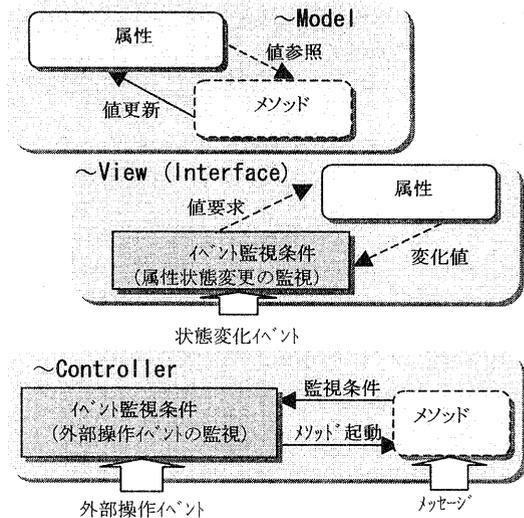


図6 基本動作をペアにして分離したMVCクラス

上記のようにMVCモデルを解釈することにより、Viewクラス、およびControllerクラスを抽出する基準のガイドラインを導くことができる。

●Viewクラス属性候補抽出の基準

図6で示したように、Viewクラスの特徴は、属性が何らかのイベント監視条件と密接に関連

していることである。例えば、Model クラスで非同期的に発生する値更新イベントを監視して導出属性“合計値”を再計算するときや、入力欄への値の記入イベントを監視して、その値を Model クラスの属性に代入したり、逆に Controller クラスからの操作イベントを監視して、Model クラスの属性値を表示するなどである。

このように View クラスの属性は、監視しているイベントの発生によって属性の値を変化させる特徴をもつ。これは View クラスが MVC クラスで構成されるシステムのインタフェース部分を役割分担する以上、システムの内部で発生するイベント、および外部からのイベントによって、ユーザとシステム内部との間で属性値の交換を必要があることから当然といえよう。したがって、オブジェクト指向のもつ動作機構の役割分担側から見ると、View クラスに格納する属性候補を決定するには、次のような判断基準が妥当といえよう。もちろん、ユーザ側からも View クラスに格納すべき属性候補を決定する必要があるが、これらは、ユースケース分析等から決定することが可能であるため、以下では割愛する。

《動作機構から見た View クラス抽出判断基準》

【基準 B1】値の導出性

Model クラスの属性候補は、他の属性から導出されない基本属性であるのに対して、View クラスの属性は、Model クラスの属性から導出される属性である。

【基準 B2】イベント監視条件の同一性

Model クラスの属性候補はその初期値が決定される時点进行分类基準としたが、インタフェースとしての View クラスの属性候補は、必ず値のやりとりのきつとなるイベント監視条件をもつ。そこで、監視すべき条件が同一のデータ項目は、同じ View クラスに含ませる。例えば、“売上が確定したとき”とか“明細数量が変更されたとき”、“クレームがあるとき”等である。

イベント監視条件は、また属性候補の値変化の原因あるいは理由と捉えることもできる。したがって、この基準は“イベントが同一の理由あるいは原因があるとき”は、同じ View クラスに含ませるべきであると読みかえることができる。

このような捉え方は、ユースケース分析と共通するものと考えられる。なぜなら、システムに対する具体的な利用法に関するシナリオは、その背景に、そのような情報を必要とする理由や原因が必ず隠されているはずであり、そのような理由や原因が発生した状況を洗い出すことは、すなわち、属性のもつ値のやりとりを起すイベントを監視する条件に他ならないからである。

これらの基準をすでに述べた基準 A と併せ、適用順序としてまとめると図 7 のようになる。

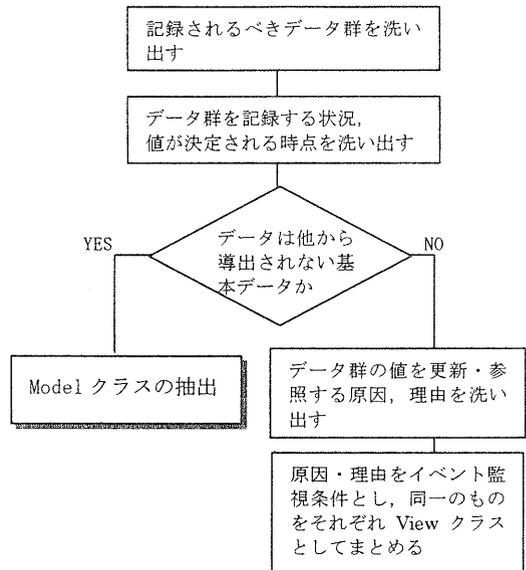


図 7 MV 基本クラス構造の抽出の手順

なお、図 7 における「Model クラスの抽出」は図 8 で示す概念モデルの抽出手順で抽出したものを流用するものとする。図 8 で示した判断基準と手順の有効性評価については、別稿[5][6]で行っている。

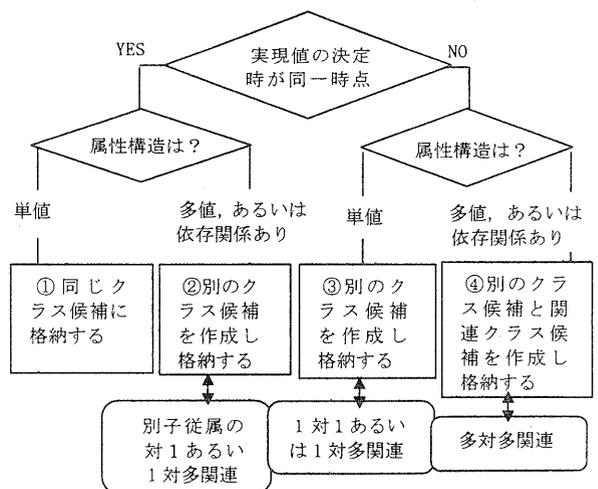


図 8 概念モデル抽出の手順

5. 考察と課題

以上に述べてきた MV 基本クラス構造決定の判断基準は、あくまでも基本的なクラス構造を抽出するガイドラインであり、メソッド内の操作を決定してゆくにつれ、副次的にクラスを追加する必要がでてくる可能性は十分ある。しかしながら、冒頭でも述べたように、分析設計の初心者にとって、基本クラス構造する決定できない場合が多く、図7で示したガイドラインであっても、十分判断基準としては機能する。

概念モデルに関する判断基準は、すでにモデリングツールの機能として実現しており、具体的な効果をあげている。MVC 基本クラス構造をオブジェクト指向がもつ動作機構の側面から抽出するには、さらに検討が必要であり、現在 UML や DATARUN 等の他の方法論と比較した妥当性の検証を継続して行っている。

6. おわりに

本論で述べた判断基準は、オブジェクト指向の基本機構から導いたものであり、オブジェクト指向分析設計で作成すべき各種のドキュメント(相互作用図, 状態図, イベントシーケンス図等)と整合性をとったものに至っていないが、今後明らかにしてゆきたい。

なお、本研究の一部は、情報サービス産業協会 HITOCC の補助金のもとに行われている。

[参考文献]

- [1] Craig Larman ; “Applying UML and Patterns: an introduction to object-oriented analysis and Design Prentice Hall, 1998
“実践 UML” プレンティスホール, 1998
- [2] 大木幹雄; “協調動作機構の実体関連分析プロセスへの適用”, 情報処理学会 オブジェクト指向シンポジウム’99, Vol.99 No.9, PP.127-136
- [3] 加藤貞行; “オブジェクト識別についての一考察とその効果; 情報処理学会研究報告, 信学技術, Vol.98, No.100, 98-SE-121, 1998/11
- [4] 加藤貞行; “失敗のないシステム開発入門”; 日経 BP, 2000
- [5] 大木幹雄, 稲田晃; “クラスモデリングにおける判断基準の分類と 3D-CASE ツールの試作” 電子情報通信学会 知能ソフトウェア工学研究会 信学技術 Vol.99 No.312 pp.9-16
- [6] 大木幹雄, 秋山構平; “多重継承を含むクラス自動抽出機構の検証”; 電子情報通信学会 知能ソフトウェア工学研究会 信学技術 Vol.100 No.90 pp.25-30