

メタ階層アーキテクチャによる プロトタイプシステム開発とその考察

武野内 康寛 大島 俊泰 平井 譲 上田 賀一

茨城大学 工学部 情報工学科

〒 316-8511 茨城県日立市中成沢町 4-12-1

ソフトウェアシステムの開発において、要求者と開発者の意思の疎通を図る有効な手段としてプロトタイピング手法がある。本研究室ではこれの実現に向け、メタ階層アーキテクチャに基づくモデリングおよびシミュレーション支援環境の構築と、その実装言語であるモデル記述言語 Bramble の開発を進めてきた。本報告では、この環境の適用事例として大学における授業の履修申告表の作成/支援システムの開発を行う。この事例を通して、この環境の有効性と Bramble の実用性について考察を行ない、それらがソフトウェア開発において十分なものであることを示す。また今回、本研究室で提案しているソフトシステム方法論を利用したドメイン分析法により分析を行ない、その結果をこの環境での開発に利用することを試みた。この分析法とモデリング環境の適性についての評価も行なう。

Development of Prototype System based on Meta Hierarchical Architecture

Yasuhiro Takenouchi Oshima Toshihiro Yuzuru Hirai Yoshikazu Ueda

Ibaraki University

4-12-1 Naka-Narusawa, Hitachi, Ibaraki, 316-8511 Japan

In software development, there is a prototyping method which effects the mutual communication between requirers and developers. Toward the realization of this method, the authors' laboratory constructs the modeling and simulation supporting environment based on meta hierarchical architecture and develops a model description language: Bramble as a implementation language. This report presents the development of the supporting system of making self-curriculum as an application example of this environment. Through this example, the effectiveness of the environment and the usefulness of the language are assessed. And these features are shown to be sufficient in software development. In this report, the above system is analyzed by soft system methodology based domain analysis and modeling. The analyzed result is applied to the development in this environment. The correspondence between the analysis method and the modeling environment is also assessed.

1 はじめに

近年ソフトウェアシステムは、その要求の高度化に伴い年々大規模/複雑化している。そのようなシステムの開発において、要求者と開発者との間の意思的な隔たりは大きくなる一方であり、またその隔たりがソフトウェアシステムの品質に影響を及ぼすようになってきている。

これに対し我々は、ソフトウェアをモデル化し、そのモデルを通じて開発者と要求者の意思の疎通を図ることが有効であると考え、これにより、開発者側がシステムの実行可能なモデル（プロトタイプ）を作成し、それを基に要求者側からのフィードバックを得るというプロトタイピングや、要求者自らがソフトウェア開発に直接参加するエンドユーザコンピューティングが容易に行える。

本研究室ではこれらを実現する環境として、種々のモデリング過程を自在に構築できるモデリングおよびシミュレーション支援環境の開発を進めている [1][2]。この環境は、メタ階層アーキテクチャに基づいたモデリング支援を行い、オブジェクト指向であるモデル記述言語 Bramble によりその実装を行うというものである。

しかし、現在この環境における事例は少ないため早急に適用事例が求められている。そこで今回、この環境での適用事例として、現在その作業が煩雑化している大学における学生の履修科目届の作成を取り上げる。この作成を支援するソフトウェアシステムを開発し、この環境における適用事例として示す。

また現在、この環境における明確なモデリング方法が確立されていないため、試行錯誤を行ない進めなければならない。これを改善するために今回の適用事例において、本研究室で開発されたソフトシステム方法論に基づいたドメイン分析手法（以下、SSM based DAM と呼ぶ）[3]を用いて問題領域の分析を行ない、その結果をメタ階層アーキテクチャに基づく開発に利用することを試みた。

2 基盤技術

本稿で利用される SSM based DAM、メタ階層アーキテクチャ、モデル記述言語 Bramble についてそれぞれ説明する。

2.1 SSM based DAM

SSM based DAM は、システム工学から生まれた方法論である SSM(Soft Systems Methodology) をドメイン分析に応用した手法である。この手法は大きく分けて前半と後半部分の2つから成っており、前半部分の Domain Component Modeling(DCM) と後半部分の Domain Aggregation(DA) を図1に示す。

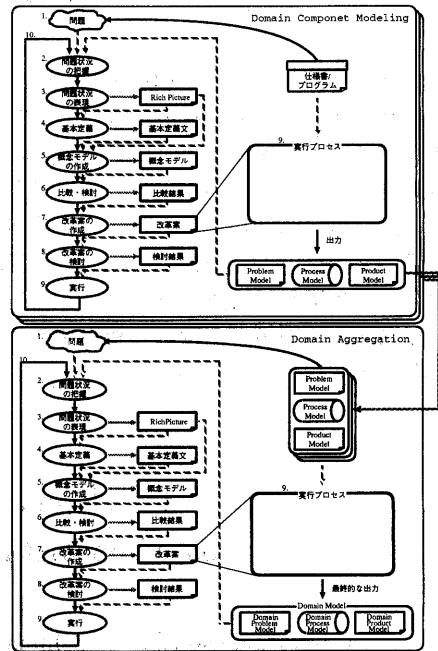


図 1: SSM based DAM

前半部分の DCM はドメインを分析し、問題、仕様書/プログラム、および問題から仕様書/プログラムを導出するプロセスという3つのモデルを作成する手順を示すものであり、後半部分の DA は、DCM によって作成された複数のドメインのモデルを集約または抽象化するものである。

DCM, DA の作業の流れは共に、次のプロセスを繰り返すことによりモデルが洗練されていく。

1. 「問題」
2. 「問題状況の把握」
3. 「問題状況の表現 (Rich Picture の作成)」
4. 「基本定義」

5. 「概念モデルの作成」
6. 「現実世界との比較・検討」
7. 「改革案の作成」
8. 「改革案の検討」
9. 「実行」

この方法論では、ドメイン分析にソフトシステム方法論を導入していることにより、ドメイン分析では取り入れられていなかった人間活動を含めた問題解決のプロセスを提供するという特徴を持っている。これによって、ドメイン分析しながらより理解を促進させることができ、洗練されたモデルを作成することができる。

2.2 メタ階層アーキテクチャ

ソフトウェア開発では、作成するシステムの全体像を把握するためにソフトウェアをモデル化することが有効である。そのモデルには記述したい目的によって様々な種類がある。それらを統合的に扱えるよう、メタ階層アーキテクチャではモデルを記述/解釈するものとしてメタモデルを定義する。さらにメタモデルの作成をより容易にするために、メタモデルを解釈するメタモデルを定義し、モデルのメタ階層を、ベースレベル、メタレベル、メタメタレベルの3つの階層に分割した。

これにより、モデル記述者が適切なメタモデルを選択しドメイン固有の問題の記述に専念することができ、また、ドメインのエキスパートがドメイン固有のメタモデルを構築することも可能となる。

2.2.1 モデルのメタ階層

モデルのメタ階層は、3階層から構成される(図2)。以下、メタメタレベルから順に述べる。

メタメタレベルには、メタレベルのモデルを記述するためのモデルである“メタメタモデル”が存在する。メタメタモデルはこのメタ階層アーキテクチャにおいて固定であり、後に述べるERFモデルがその意味的記述を担う。

メタレベルには、ベースレベルのモデルを記述するためのモデルが存在する(本稿では、メタモデルとはこのモデルを主に指す)。メタレベルのモデルは、その定義に従ってベースレベルのモデルを記述するツール(モデラ)を持つ。これにより、モ

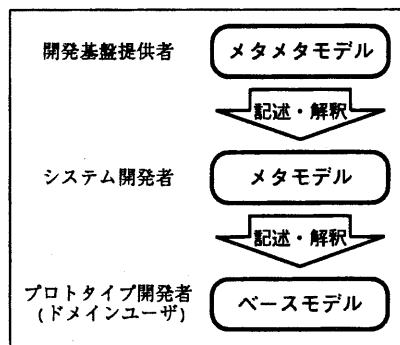


図2: モデルのメタ階層構造

デルのドメインに応じたモデリング方式を採用することができる。

ベースレベルには、メタレベルのモデルによって記述されるモデルが存在する。通常、要求者が求める成果物はこのレベルのモデルとなる。

2.2.2 ERF モデル

ERF(Entity-Relationship-Field) モデルは、ERモデルを基本とし、プロトタイピングにおけるオブジェクト指向の有用性からエンティティとリレーションシップをオブジェクトとして扱うこととしている。さらにオブジェクトの集約を扱うために、フィールドの概念を取り入れている。

また、ERFモデルは次の特徴も持っている。

- 各要素がオブジェクトであるので属性を定義可能
- 関連の多重度を定義可能
- フィールドを用いることによりモデルの階層的記述が可能

2.3 Bramble

本研究室では、メタ階層アーキテクチャに基づいたモデリングおよびシミュレーション支援開発環境の構築を進めている。この開発環境は前述のメタ階層構造を持ち、各階層を各レベルのユーザが受け持つ形態を取る。この開発の有する概念を損なわずに記述・解釈できる記述言語として、オブ

ジェクト指向であるモデル記述言語 Bramble を開発している。

Bramble には、次のような特徴がある。

- すべての要素がオブジェクトである
- コピー操作を主体としたオブジェクト指向概念を用いている
- 型付けがない

Bramble で実装する利点として、次のことが挙げられる。

- メタ階層を同一環境上で扱うことができる
Bramble にはメタの概念は含まれていないが、Bramble の言語仕様は拡張性に富んでいるため、メタの概念を実装することですべてのオブジェクトを同一環境上で扱うことができる。
- データベースを用いてオブジェクト管理が可能である

Bramble は分散オブジェクト技術 CORBA に対応しており、データベースとの通信手段としても利用している [4]。また、そのデータベースとの通信手段を Bramble のオブジェクトとして提供しているのでデータベースとの通信が容易となっている。これにより、Bramble によってオブジェクトの管理をすることができ、再利用性の向上が図れる。

- GUI オブジェクトの利用が可能である
Bramble には GUI オブジェクトが存在する。これを使うことによって GUI を用いたアプリケーションを簡単に構築することができる [5]。

3 履修科目届作成の分析

プロトタイプシステム適用事例として、現在煩雑化している履修科目届の作成に着目した。履修科目届とは、大学の講義を履修するために記入する専用の用紙のことを指している。その履修科目届の作成を支援する履修科目届作成支援システムを開発することでこの開発環境における適用事例を示すことにした。

3.1 現在の履修科目届作成とその問題点

現在学生が科目を履修するためには、学務係から配布された履修要項と講義要目を参考にその科

目に対応するコードを専用用紙に記入するという作業を行っている。その中で、必ず履修しなければならない科目、ある特定の科目の中から一定の単位を履修しなければならない科目等が存在し、また卒業研究着手・卒業条件等の様々な制約があり、さらに再履修、他学科科目の受講等の要素が存在し絡み合っているため、履修科目の作成は非常に複雑化/煩雑化している。このために取り忘れや書き間違いがでてしまい、学生は訂正に、学務係はその対応に追われることになる。

3.2 SSM based DAM による分析

現在メタモデルを作成する際にその作成方法はまだ確立されていないため、試行錯誤を行ない進めなければならないが、今回メタモデルの作成を支援する分析法として、本研究室で開発されたソフトウェアシステム方法論に基づいたドメイン分析モデリングの使用を試みることにした。

その SSM based DAM を使用することにより得られた結果の一部を次に示す。

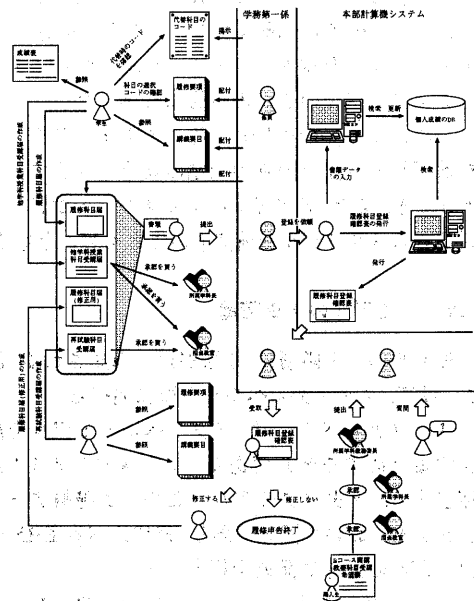


図 3: 履修科目届作成の Rich Picture

問題の状況を把握するために、現在の履修科目

届作成の状況を Rich Picture で示すと図3のようになった。

現実世界をモデルとして取り入れることができない項目については改革案を提示し、それをもとに Rich Picture を修正した。そして最終的に結果として出力された Problem Model を図4に示す。

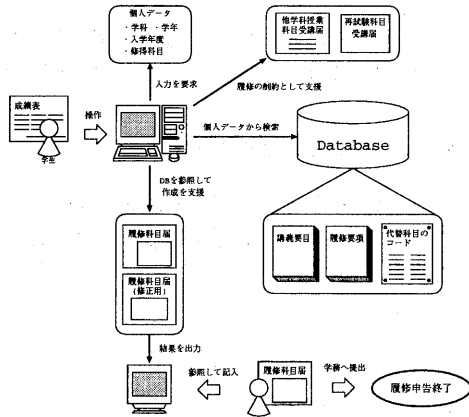


図4: Problem Model

今回、対象としたドメインが1つしかないということから集約の必要はなく、SSM based DAMの後半部分である DA は行わなかった。

4 履修科目届作成支援システムの設計

前章で行った分析に基づき、履修科目届作成支援システムの設計を行った。ここではその設計について述べる。

4.1 メタ階層アーキテクチャに基づいた設計

今回の履修科目届作成支援システム(以下、支援システムと呼ぶ)では、目的とする成果物を「個人の履修科目届」とし、それをベースレベルのモデルと定めた。これによりメタレベルには、それを記述/解釈するメタモデル、すなわち履修科目届のメタモデルが置かれることになる。また、そのメタモデルに基づき個人の履修科目届を生成するツール

(履修科目届作成支援ツール、以下支援ツールと呼ぶ)がメタレベルのツールとして位置付けられる(図5)。これに基づき、まずメタモデルの作成を行い、次に支援ツールの設計を行う。

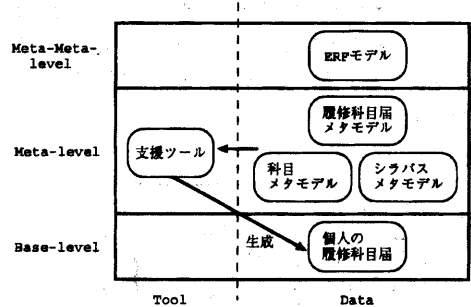


図5: 支援システムにおける各モデルの位置付け

4.2 メタモデルの作成

メタ階層アーキテクチャに基づいたソフトウェアの開発において、このメタモデルの作成が最も経験を要し難しい作業と言える。そこで、前章で述べた分析過程で作成された Rich Picture および Problem Model などを基にメタモデルの作成を行なう。

4.2.1 履修科目届メタモデル

ベースレベルのモデルとして定められた履修科目届モデルについて、それを記述する履修科目届メタモデルを作成した。作成したメタモデルを図6に示す。なお、この図では省略しているが、Entity や Relationship には、例えば曜日は月から土までであるといった制約が記述される。

4.2.2 科目メタモデル

履修科目届メタモデルにおいて、各欄は「科目」を持つ。しかしここでの「科目」は具体的な情報を持ったものであり、それはメタ階層におけるベースレベルのモデルであると言える。ここで、科目のメタモデルも当然存在することが考えられる。つまり、履修科目届メタモデルで科目モデルが必要とされ、履修科目届に関するメタ階層と科目に関するメタ階層が一段ずれることとなる。しかし、メ

タ階層アーキテクチャにおいて階層のずれは特に規定されておらず、その様な記述は避けなければならない。

そこで、ここでは履修科目届メタモデルと科目メタモデルを同じ階層に置くこととした。つまり、履修科目届メタモデルに科目メタモデル(科目 Field)を含ませることにより一応の解決とした。これらから生じる問題等は、支援ツールが吸収することとなる。

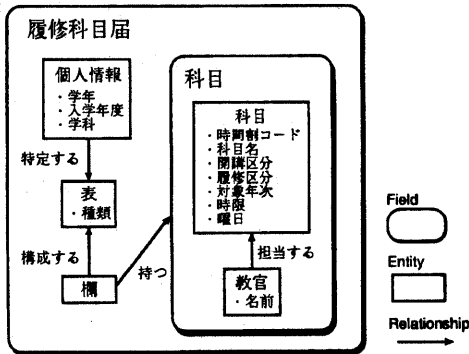


図 6: 履修科目届メタモデル

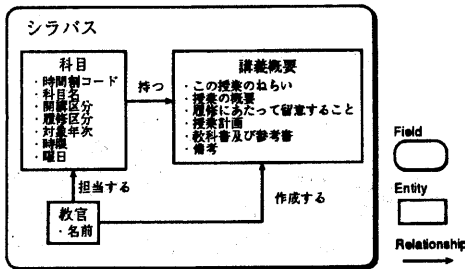


図 7: シラバスメタモデル

4.2.3 シラバスメタモデル

SSM based DAM による分析より、履修科目届の作成には科目に対する詳細な情報、つまりシラバス(講義要目とも呼ばれる)の参照が必要であることがわかる。そのためシラバスのメタモデルも作成した。シラバスメタモデルを図7に示す。

ここで、科目 Entity および教官 Entity は図6の

科目とはほぼ同一である。これらから記述された科目モデルは、同一のものを示すこととしている。別の見方をすれば、科目 Field とシラバス Field は同一のものであり、科目 Field はその一部のみを表していると言える。

4.3 履修科目届作成支援ツールの設計

前節で作成したメタモデルに基づき個人の履修科目届を作成するように支援ツールの設計を行った。

4.3.1 必要な機能の割り出し

Problem Model に基づき、支援ツールに必要なとなる機能を洗い出した。以下にそれらの機能について示す。

- 個人データの入力
- 修得済み科目の入力
- 履修科目の選択・決定
- シラバス情報の表示
- 履修における種々の制約の適用
- 実物の履修科目届に記述する情報の表示

4.3.2 オブジェクト図の作成

支援ツールのオブジェクト図の作成を行った。作成したオブジェクト図を図8に示す。

4.3.3 各オブジェクトの設計

前に作成した Problem Model 等に基づき、以下の各オブジェクトの設計を行った。

- ユーザオブジェクト
- 履修科目届オブジェクト
- データベースオブジェクト

また、次のオブジェクトには GUI を操作するメソッドを持たせた。

- 個人情報入力 GUI オブジェクト
- 修得科目選択 GUI オブジェクト
- 履修科目選択 GUI オブジェクト
- シラバス GUI オブジェクト

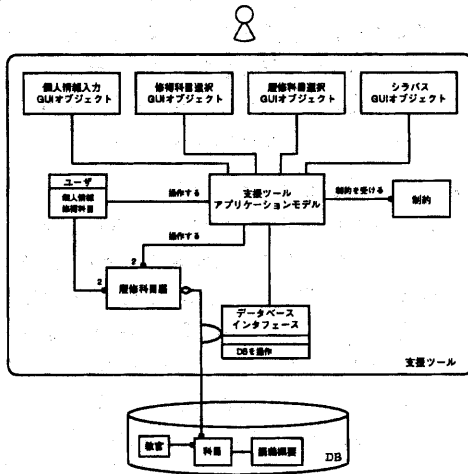


図 8: オブジェクト図

5 履修科目届作成支援システムの の実装

支援システムをモデル記述言語 Bramble により実装をした。

現在までに次の機能が実装されている。

- 個人情報の入力
学年, 学科名, 入学年度を入力する。
- 修得科目の入力
現在までに修得した科目について入力することができる。所属学科における必修/選択科目について、それぞれの科目に対し修得/未修得科目の分類をすることができる。教養科目, 共通科目, 他学科科目については修得単位数を入力することができる。
- 未修得科目の自動選択
学科と現在の学年より必修科目を自動で選択することができる。
- 履修科目の選択/決定
学科, 曜日/時限を選択することでその時間枠における取得可能な科目が, 教官名, 単位数等とともに表示される (図 9 参照)。
- 履修科目の詳細の参照
科目リストで選択されている科目についてシラバスを見る感覚で詳細を参照することがで

きる。

- 履修における制約
ユーザの現在の状況に応じて卒業/卒業研究着手条件等の制約が適用される。
- 履修科目届の記入に必要な情報の出力
現在選択されている科目について履修科目届の記入に必要な情報を出力することができる。

実際に支援システムを実行した画面を図 9 に示す。

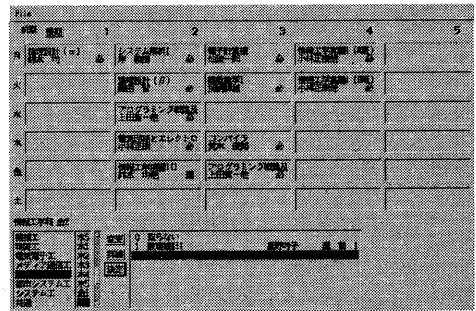


図 9: 履修科目の選択・決定画面

6 考察

本稿では, 我々の研究室で開発したモデリング環境により履修科目届作成支援システムを作成した。この作成過程の中で, SSM based DAM, メタ階層アーキテクチャ, モデル記述言語 Bramble 等について考察した。

- SSM based DAM の有効性について
今回, SSM based DAM を用いて分析・モデリングを行なった。これにより, ある程度メタモデルの作成が楽になり, モデリングに有効であることが分かった。しかし, SSM based DAM によってモデリングされたものと実際にメタ階層アーキテクチャで求めているものとはモデルの粒度に差があるため, 多少の手直しが必要である。SSM based DAM ではドメイン全体をモデル化しているため粒度が粗いのにに対し, メタ階層アーキテクチャでは粒度の細かいものに対して適用すると効果がある。このため, SSM based DAM により得

られたものからメタ階層に適用する際に、粒度を細かくする手法があれば、より有効であると考えられる。

- Bramble の実用性

今回、モデル記述言語 Bramble により履修科目届作成支援システムを実装した。このシステムでは Bramble の GUI オブジェクトを利用することによってユーザに分かりやすい環境を与えることができ、また、Bramble のデータベースオブジェクトを利用することによってデータベースから科目、講義概要を参照することができ、Bramble は実用的である。

- モデリング環境の有効性

今回は我々がシステム開発者とドメインユーザ(プロトタイプ開発者)のどちらも兼ねており、さらにドメイン分析者が存在しなかったため試行錯誤を繰り返すこととなってしまった。しかし、我々が考えている環境では、メタ階層のレベルごとのユーザの分離をし、ドメイン分析者により分析をすることにより、SSM based DAM による分析、メタモデルの設計、Bramble による実装という流れがスムーズに行われていくことになり、このモデリング環境及びモデル記述言語 Bramble の実用性は、プロトタイプ開発として十分に利用可能であると思われる。

7 まとめ

本稿では、我々の研究室で開発したモデリング環境における適用事例として履修科目届作成支援システムの開発を行い、この環境の有効性やその実装言語であるオブジェクト指向モデル記述言語 Bramble の記述性についてその検証を行った。

本稿で提示した適用事例は一例だけであるが、このモデリング環境およびモデル記述言語 Bramble は、プロトタイプ開発において十分に有益であるという感触を得た。

今後の課題としては、以下のようなものが挙げられる。

- 種々の方面への適用

モデリング環境における適用事例の提示は、い

まだ十分とは言えない。今後このモデリング環境において、種々の方面へソフトウェアシステムの適用を行う必要がある。それにより、この環境がプロトタイプ開発において有益であることを示さなければならない。

- メタモデル作成方法の確立

このモデリング環境において、適切なメタモデルを作成することは経験を必要とし、まだ困難となっている。今回は SSM based DAM を用いることによりある程度困難を回避することができたが、完全には回避することはできていない。この問題を解決することもまた、種々の方面へ適用し、評価/検討することにより、これをサポートするようなモデリング方法を確立する必要がある。

これらを解決することにより、モデリング環境はソフトウェア開発にとってより有益なものになると考える。

参考文献

- [1] 高橋 大輔, 上田 賀一: 「メタ階層に基づくモデル構築とその支援環境」, 近代科学社, ソフトウェア工学の基礎 III, 日本ソフトウェア科学会 FOSE'96, 146-149 (1996).
- [2] 中野 喜之, 上田 賀一: 「メタ階層アーキテクチャによるモデリング環境の構築」, 近代科学社, ソフトウェア工学の基礎 IV 日本ソフトウェア科学会 FOSE '97, pp.146-149 (1997).
- [3] 齊藤 恵子, 上田 賀一: 「ソフトシステム方法論を利用したドメイン分析法の提案」, 情報処理学会研究報告 '96-IS-58 1996年3月
- [4] 藤田 充典, 小銅 敬, 上田 賀一: 「メタ階層モデル記述言語 Bramble の分散オブジェクトへの対応」, 情報処理学会研究報告 '98-SE 1998年3月
- [5] 庄司 龍一, 中野 喜之, 上田 賀一: 「メタ階層モデル記述言語 Bramble のポータビリティの実現」, 情報処理学会研究報告 '98-SE 1998年3月
- [6] J. ランボー, M. プラハ, W. プレメラニ, F. エディ, W. ローレンセン (羽生田 栄一 監訳): 「オブジェクト指向方法論 OMT」, トッパン (1992)