ソフトウェア工学 107－9
（1996．1．18）

ソフトウェアリポジトリにおける構造制約管理機能の設計について

伴 啓太† 沢田 篤史‡ 満田 成紀† 鯵坂 恒夫†

†京都大学工学部

‡奈良先端科学技術大学院大学 情報科学研究科

統合型CASE環境におけるデータ管理の統合に使用されるソフトウェアリポジトリにおいて，構造制約と意味制約の管理を分離することを考える．それによって，制約情報のよりきめの細かい，より抜けのない管理が可能となる．E-Rモデルを用いて情報をモデル化する際，用いる概念要素に制限を加えることにより，同意の情報に対して異なるモデルができる．これら複数のモデルを，加えた制限と定めた指標に対する値によって比較することにより，構造制約管理機能を実現するためのデータモデルの指針とする．

Towards a Design of Structural Constraints Management for Software Repositories

BAN Keita† SAWADA Atsushi‡ MITSUDA Naruki† AJISAKA Tsuneo†

†Faculty of Engineering, Kyoto University

‡Graduate School of Information Science,
Nara Institute of Science and Technology

Data integration facilities in CASE environments are provided by software repositories, which involve all aspects of software subject information with various kinds of constraints. For consistent management of them, structural and semantic constraints should be layered in a repository apart from tool programs.

In this paper, we examine several limitations on the E-R model to find a model that represents pure structural constraints and therefore that is independently plugable with semantic constraints. Those alternative E-R models are compared using some quantitative metrics, giving a guide for a universal structural management in software repositories.

1 はじめに

統合型CASE環境には、ツール間のデータ管理を統合化する目的で、リポジトリと呼ばれるソフトウェア開発用データベースが使用される。ソフトウェア・リポジトリには開発対象のソフトウェアを表現するための様々なデータが蓄積され、各ツールに対して、検索、分析、一貫性保持などのデータ管理機能が提供される。

情報をモデル化するためには，その構造的な制約に関する記述と，意味的な制約に関する記述が必要である．多くのソフトウェアの分析・設計方法論では，E-Rモデル[1]あるいはそれに類似するモデルがソフトウェアの情報構造を記述するために用いられている．リポジトリの国際規格となったPCTE[2][3]においても，データモデルとしてE-Rモデルを拡張したモデルが使われている．PCTEデータモデルでは，E-Rモデルの実体，関連，属性に対応する，オブジェクト，リンク，属性という概念が使われる．PCTEにおいて特徴的なことは，5種類のリンクカテゴリ，継承関係，リンクのカージナリティによって情報の持つ構造制約，意味制約を表現している点である．

しかし，集約関係と継承関係の相対性がない，各リンクカテゴリの制約の強さには大きな開きがある，など制約の構成は必ずしも満足できるものではない．さらに，PCTEデータモデルに反映できる制約だけでは，情報の持つ意味的制約を十分に表現できているとは言えない．例えば，オブジェクトの（インスタンスの）現存数やオブジェクトに対するリンクの現存数に関する制約，属性値とオブジェクトやリンクの存在の関係などを表現することはできない．これらの表現できない制約はPCTEによって管理することはできず，そのデータを利用するプログラム中に埋め込まなければならない[4][5]．このようにPCTEデータモデルが部分的に制約を含んでいることが，情報のモデル化をかえって分かりにくいする場合もある．

このことから，情報をモデル化する際に，その構造制約に関する情報と意味制約に関する情報を完全に分離して管理することを考える．それによって，次のような利点が生まれる．

- プログラム中に意味制約を埋め込むことがなくなり，構造制約，意味制約の両方に対して抜けのない管理をすることができる．
- 情報をモデル化するとき，特にその構造を考えるときに，データモデルの持つ制約に縛られることなく，構造のみを考えてモデル化することができる．
- 構造管理システムと制約管理システムを分離しコンポーネント化することにより，再利用性が高まる．

本研究では，意味制約に関する情報は別の層で管理することを仮定し，構造制約にのみ着目して，ソフトウェア・リポジトリにおける構造制約管理機能の実現のために，そのデータモデルが持つべき特徴を考えている．構造制約に限定しても，システム分析結果の表現にはゆらぎが生じ，設計の品質に影響を与える．このゆらぎをおさえるために，制約記述能力にさらにさまざまな制限を加え，モデルの比較検討を行う．

2 構造制約管理のためのデータモデルの特性

構造制約を管理するためのデータモデルが持つべき特徴を以下の3つの点から考えてみる．

1. 情報の構造をモデル化するという点
2. 意味制約は分離して管理するという点
3. ソフトウェア・リポジトリのためのモデルであるという点

第一の情報の構造をモデル化するという点について，まず用いられるデータモデルが，正確に外界の情報の構造を反映している必要がある．また，複数の開発者の間でモデルを共有できるためには，データモデルが一貫性を持つ必要がある．さらに，開発者は外界の情報を分析しモデル化する時，様々な観点からその情報を捉えているので，その思考に柔軟に対応でき，思考を直接に反映するデータモデルが望ましい．

第二の意味制約を別に管理するという点について，構造制約管理システムは，このシステム単独で利用しようとしているものではなく，意味制約管理システムと共に使用して，ソフトウェア・リポジトリとして情報を管理しようとするものである．従って，構造制約管理システムで用いるデータモデル上に，必要な意味制約を表現して貼り付けられることが必要になる．

第三の点については，ソフトウェア・リポジトリとして特に必要な機能を考える必要がある．ソフトウェア・リポジトリでは，概念要素のナビゲーションが機能の中心となる．例えばソフトウェアの再利用を考えたとき，検索すべきものを曖昧なくナビゲートする必要がある．

これらのことから，ソフトウェア・リポジトリにおいて構造制約を管理するためのデータモデルが持つべき特徴として

直感的な捉えやすさ 人間の思考に柔軟に対応できるモデルであるかどうか．

モデルの一貫性 構造制約を正確に反映するモデルであるかどうか．

意味制約の表現のしやすさ ソフトウェア・リポジトリとして持つべき制約を十分に表現できるモデルであるかどうか．

ナビゲーションの容易さ ソフトウェア・リポジトリの主な機能となるナビゲーションに向いたモデルになっているかどうか．

の４つが挙げられる．

3 E-R モデルによる情報構造の表現

先に述べたように多くのソフトウェアの分析・設計方法論で，E-R モデルは情報構造の記述法として採用されている．これは，E-R モデルが非常に直感的なモデルであり，人間の思考に柔軟に対応できること，また，実体，関連，属性，という３つの概念によってあらゆるシステムをモデル化することができるためであると思われる．

しかし一方で，E-R モデルは同じ情報に対しても様々な異なるモデルを考えることができ，同一の情報に対してモデルが一意に決まらない．この E-R モデルのゆらぎは，実体・関連・属性という３つの概念が完全に別の概念ではなく，それらの切り分けが曖昧であるためと考えられる．

そこで，E-R モデルによってシステムを表現する時に様々な制限を加える．異なる制限を加えることにより，できるモデルに違いが生まれる．それらのモデルを比較することにより，設計時に加えた制限がモデルの特性にどのような影響を与えているのかを考える．

3.1 E-R モデルに対する制限

E-R モデルで使われている概念は，実体，関連，属性の３つである．E-R モデルのゆらぎは，それらの概念の切り分けの曖昧さ，属性の適用方法，実体，関連の種類の数などによって現れる．

まず，何の制限も加えない標準の E-R モデルから出発する．これに属性の適用範囲による制限を加えると，実体にのみ適用できる，関連にのみ適用できる，という変形を考えることができる．

次に，実体，関連の種類に関しては一種か多種が考えられる．実体を縮退させてしまったものは属性をもつ関連の集合，すなわちい

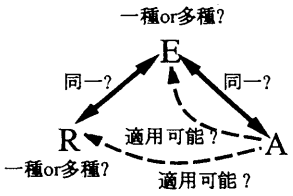

図 1: E-R モデルのバリエーション

わゆるリレーショナルモデルとなり，ここでは議論しない．関連については，一種，多種の別が考えられ，一種の場合は，属性は関連には適用できない．

概念の切り分けについて，実体と関連を同じものと考えるとやはりリレーショナルモデルになる．実体と属性を同一視するということは考えられる．この場合，関連は一種，多種ともに考えられるが，より極端な例にするためにここでは関連が一種の場合について考える．

これらのことから，以下の 5 通りの制限を加えた場合の E-R モデルについて考えることにする．

制限 1. 実体多種，関連多種，属性は実体，関連ともに適用可 (制限なし)

制限 2. 実体多種，関連多種，属性は関連にのみ適用可

制限 3. 実体多種，関連多種，属性は実体にのみ適用可

制限 4. 実体多種，関連一種，属性は実体にのみ適用可

制限 5. 実体多種，関連一種，属性＝実体（属性はなし）

3.2 モデルを評価するのための指標

前節で述べた制限の下で考えたモデルを比較し評価するために以下の指標を定める．

- 実体の種類数
- 関連の種類数
- 関連の数
- 属性の数
- 流入する関連の数の平均
- 最長経路
- 平均経路長

ここで，最長経路とは任意の 2 つの実体間を結ぶ最短経路の中で最長の経路とする．また平均経路長は任意の 2 つの実体間を結ぶ最短経路の平均である．

4 例題によるモデル比較

例題に対して，実際に制限 1. ～制限 5. までの制限の下で E-R モデルを用いてモデル化を行う．

4.1 例題の仕様

図書室の管理システムを設計する．図書室の業務として，学生への図書貸し出しと業者からの本の納入だけを考える．学生は，学籍番号，学部名，入学年度，氏名で管理される．納入業者は，会社名，住所で管理される．

図書室には，実験指導書のように，同じ本が複数冊備え付けられる場合がある．学生は，同時に幾つかの本を借りることができる．貸出においては，同じ本は区別しない．同じ本は一つの業者から購入する．

4.2 例題の制限 E-R モデル

4.2.1 実体多種，関連多種，属性は実体，関連ともに適用可

これは，何の制限もない E-R モデルである．これを図 2に示す．このモデルを制限 2. ～制限 5. のモデルを考える際の基本としている．

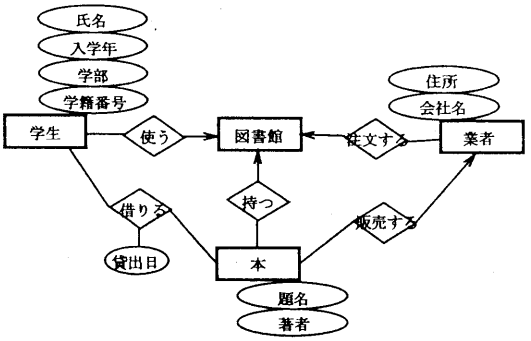

図 2: 制限 1. の場合のモデル

4.2.2 実体多種，関連多種，属性は関連にのみ適用可

この場合のモデルを図 3に示す．このモデルでは，図 2の実体に適用されていた属性をその実体が関わっているすべての関連に適用し，実体から属性を取り除いたモデルになっている．したがって，同一属性の一貫性管理が必要になる．

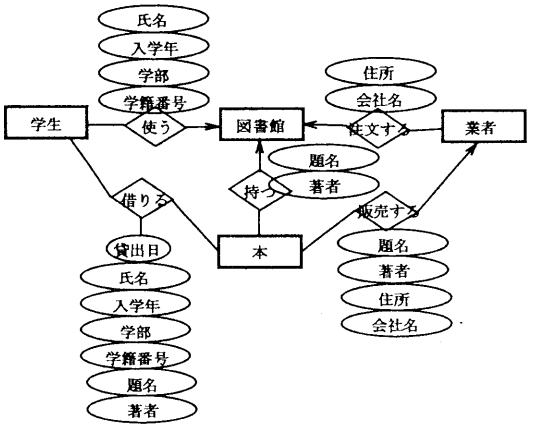

図 3: 制限 2. の場合のモデル

4.2.3 実体多種，関連多種，属性は実体にのみ適用可

この場合のモデルを図 4に示す．このモデルでは，関連に属性を適用することができない．したがって，図 2の関連が属性を持つ場合，それが 1 対 1 の関連の場合は，属性をどちらか一方の実体に適用する．1 対多の場合には，多の側の実体に適用する．多対多の場合は，その関連を新たな実体として表し，その実体に属性を適用する．また，元の関連の両側の実体のキー属性を新しく定義した実体に適用する．

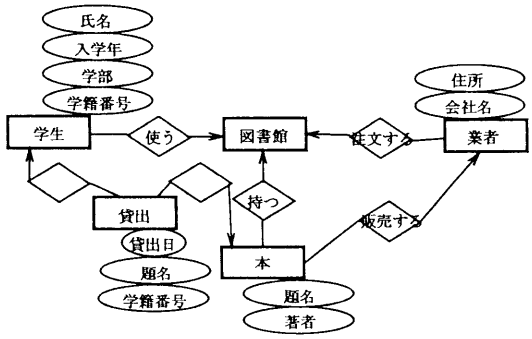

図 4: 制限 3. の場合のモデル

4.2.4 実体多種，関連一種，属性は実体のみ適用可

関連は一種類で has_a のみである．したがって，そのままでは has_a 以外の関連を表せない．has_a 以外の関連を表すためには関連に相当する実体を定義しその実体を has_a の関連で結んでいる． この場合のモデルを図 5に示す．

4.2.5 実体多種，関連一種，属性＝実体（属性はなし）

この場合は，実体と属性を同一のものとして考えている．つまり，それぞれの実体は一

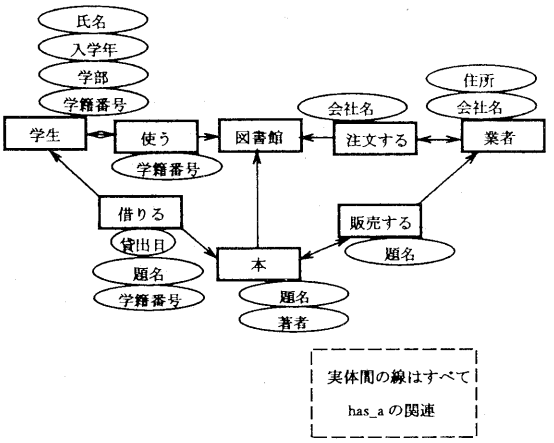

図 5: 制限 4. の場合のモデル

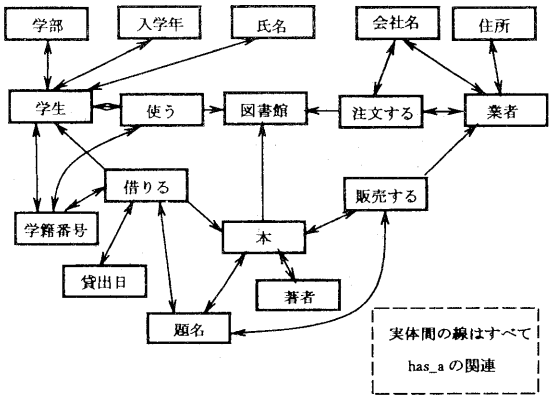

図 6: 制限 5. の場合のモデル

つの値と対応関係を持っていることとしている．関連は一種（has_a）だけとしているので，図 2での実体に適用された属性はそれぞれ実体となり，元の実体に has_a されていることになる．この場合のモデルは図 6の様になる．

5 評価と考察

各モデルに対する 3 章で与えた指標の値を表 1に示す．この表を元に各制限によって作られたモデルを比較する．

表 1: 各モデルの指標の値

指標＼制限	1.	2.	3.	4.	5.
実体の種類数	4	4	5	8	17
関連の種類数	5	5	4	1	1
関連の数	5	5	6	9	23
属性の数	9	19	11	14	0
流入する関連の数の平均	2.5	2.5	2.4	2.25	2.71
最長経路	2	2	2	4	6
平均経路長	1.17	1.17	1.4	1.93	2.72

5.1 各モデルの比較による考察

まず，制限 1. と制限 2. を比べてみると，制限 2. は制限 1. の属性の適用範囲を関連のみに制限したものである．表で比較してみると，値が違うのは属性の数だけある．属性数が増加しているのは，一つの実体に適用されていた属性が流入（流出）するリンクすべてに適用されたためである．このとき同じ名前の属性が複数現れており，情報の重複を管理しなければならない．

次に制限 3. について考えてみると，属性数は制限 2. の時ほど増加していない．多対多の関連に属性が適用されている場合にそれを実体として扱う必要があり，属性数の増加は，その時に両端にあった実体のキー属性を新たに適用する必要があるために起こる．一対一，一対多の関連が属性を持つ場合には，一方の実体あるいは多の側の実体にその属性を適用することによって関連を実体として扱うことは避けられるが，本来関連が持つべき属性を実体に適用する必要があることから，図 2と比較してモデルがその元になる情報を的確に表現しているとは言えない．

制限4.は，制限3.の関連を一種に限定したもので，これにより先にのべたような，関連を実体としても関連としても捉えることはできなくなり，has_a以外の関連はすべて実体として考える必要が出てくる．その結果，このモデルでは実体の種類数は増加している．また，図2での関連を実体として扱うため，元々関連に適用されていた属性を対応する実体に適用できるという点では元になる情報を反映できていると言えるが，逆に元々関連が属性を持たない場合でも，一対多，多対多の関連を実体として扱った場合には，多の側の実体のキー属性を関連に対応する実体に適用する必要がある．これが，属性数増加の原因となっている．

制限5.は，制限4.で実体に適用されていた属性をすべて実体にしたものである．制限4.から実体が複数の属性の組を表現する能力をなくしたモデルとも考えられる．このため個々の属性が組としての制限から解放されたため，重複がなくなっている．表一の実体の種類数，関連の種類数，属性数の合計は，制限1.と制限5.において最小であり，情報の重複が最も少なくなっていると考えられる．

5.2 各指標からの考察

次に，各指標の値から考えられることについて述べる．まず，先にも述べたように属性数は情報の重複の多さを表していると思われる．制限5.は属性なしとしているので他のモデルについて考えると，情報の重複は，多い方から制限2，制限4，制限3，制限1となる．属性数が多ければ多いほど，重複した属性の値の一貫性を保つ負担が大きい．

流入する関連の数の平均は，ある実体を検索するためのパスの数に関係していると思われる．同一数の実体に対してより多くの関連を持つという事は，それだけ柔軟に実体を検索することができると言うことである．そうであるならば，制限5，制限1と制限2，制限3，制限4の順でナビゲーションが容易という事になる．ただし，制限5では，実体の表すもののレベルが他のモデルとかなり違う．制限3，制限4で，この値が小さくなっていることから，関連を実体として扱うことによってこの値は小さくなるいえる．

平均経路長は，ある実体を起点として別の実体を検索するときにたどる必要な関連数に関係する．したがって，この値は検索にかかるコストに関係すると考えられる．この値は，制限1～制限5まで実体の種類数，最長経路とともに増加している．実体の種類数が増えると，経路は長くなる．

流入する関連の数の平均と平均経路長を見比べてみると，制限3，制限4では平均関連数は減少し，平均経路長は増加している．これは，ナビゲーションの柔軟さは減少し，更にナビゲーションのコストは増加しているという事になる．制限3，制限4は，制限1のモデルの関連を実体として扱ったモデルであるから，ナビゲーションに関しては，関連を実体として考えることは好ましくないと考えられる．

属性の種類数は表には書いていないが，制限5以外ではすべて9であり，これに，実体の種類数，関連の種類数を加えると，すべてのモデルで18になっている．制限1のモデルで，図書館と本の間の「持つ」（has_aと同じと考えている）という関連が別の関連であったなら制限3，制限4，制限5でのこの値は1変ったはずだが，それでもだいたい同じ数である．これは，同じ情報をモデル化するとき概念として考えられる物の総数はあまり変わらないと言うことだろう．

6 おわりに

ソフトウェア・リポジトリを構造制約と意味制約を完全に分離して管理するシステムとして実現することによって，モデル化されるべき情報に対してよりきめの細かい管理をすることが可能になる．

本研究では，構造制約を表現するためのデータモデルが持つべき特徴について考えた．また，E-Rモデルを用いて情報をモデル化する際に，分析段階で幾つかの制限を加えることによって，結果として得られるモデルにどのような影響を与えたか，どのようなE-Rモデルのゆらぎに対応するのか，について考察を加えた．また幾つかの指標を与え，各モデルを量的に比較することによってそのモデルの持つ（あるいは，その制限の持つ）特徴についても考察した．

ただし，ここでも用いた指標は基本的にこう雑用素数を計数するもののみである．今後，意味制約との親和性を考慮し意味制約管理のベースとして効果的な構造管理モデルを求める必要がある．

参考文献

[1] Chen, P.P.: The entity relationship model: toward a unified view of data,*ACM Trans. Database Syst.*, Vol.1,No.1,pp.9–36

[2] ISO/IEC: 13719-1 Information technology – *Portable Common Tool Environment(PCTE)*, 1995.

[3] 鯵坂恒夫，沢田篤史，満田成紀：ソフトウェア評論 Emeraude PCTE, コンピュータソフトウェア，10巻，2号，pp.65–77, 1993.

[4] A. Sawada, N. Mitsuda, T. Ajisaka and Y. Matsumoto: Utilities for Avoiding Constraints Violation in Creating and Updating the Data in PCTE, Cambell, I. (ed.), *Proc. PCTE '93 Conference, PCTE Technical Journal*, No.1, pp.93–113, PIMB Association, 1993.

[5] 沢田 篤史，鯵坂 恒夫：オブジェクト管理システムにおける制約管理について，ソフトウェア工学の基礎 II(日本ソフトウェア科学会 FOSE'95），レクチャーノート / ソフトウェア学 15，pp.153–158，近代科学社，1995.