ソフトウェア仕様記述におけるカテゴリ論の応用について *

4M－6　篠原伸生　松本 吉弘 †
京都大学工学部‡

1 はじめに

ソフトウェア開発の要求分析段階において，対象となる実世界の情報構造を抽出するための作業としてデータモデリングがある．しかし実体関連モデルに代表される従来のデータモデルは，習熟するまでに時間を要する形式的仕様と比較して，そのグラフ構造により全ての人がシステム概観を直観的に把握できるという利点はあるものの，データの構造的側面の把握に重点が置かれているため，意味的側面の記述ができない，実現段階との連結度が低いなどの欠点がある．

そこで本研究ではデータモデル中に厳密な数学的理論であるカテゴリ論に基づいた構成を導入することを試みる．カテゴリ論の普遍写像性や可換関係の記述を付加することにより，データ間の制約や関係を抽象的に記述する枠組を提供でき，形式性を伴った厳密な新しいモデルが構築できる．さらにこの手法の導入により，データモデリングの段階でのプロトタイピングを支援するようなシステムの構築も可能であると思われるので，本稿ではその概観についても述べる．

2 データモデルへのカテゴリ論の適用

2.1 カテゴリ論

カテゴリ論とは純粋数学から派生した理論であるが，構文と意味との関係を明確に定義できるので，計算機科学の分野でも幅広く用いられている．カテゴリ論の特徴は，対象の内部構造の性質がその要素に言及することなく射の普遍性により特徴付けられ，普遍写像性の証明に図式による議論を展開できるところにある．

カテゴリは対象の集合と射の集合から成り，対象 A から対象 B への射 f は $f : A \to B$ のように表され，対象 A は定義域，対象 B は値域と呼ばれる．また基本公理として恒等射と射の結合の結合則に関する公理があり，射の結合は一方の射の値域と他方の射の定義域が一致するような射の対に対してのみ定義できる．さらにある対象から別の対象へ向かう合成可能な2つの経路が同じ合成結果をもたらす場合その閉じた図式は可換であるという．以上によりカテゴリは射の結合を可能とする図式とみなすことができる．簡単なカテゴリの例としては対象が集合で，射が集合間の関数を表すような集合のカテゴリがある．

*An Application of Category Theory in Sofware Specification
†SHINOHARA Nobuo, MATSUMOTO Yoshihiro
‡Faculty of Engineering, Kyoto University

2.2 実体関連モデルへのカテゴリ論の適用例

全ての関係が多対一になるように標準化した実体関連モデルについて考える．このモデルは節点が実体や属性に，矢印が実体間の多対一関係，もしくは実体とその属性との関係にそれぞれ相当する図式であると考えられる．さらにこのモデルの節点を集合，矢印を集合間の関数とみなすことにより集合のカテゴリとして扱うことができる．カテゴリ論ではその意味は集合のカテゴリへの関手を通して定義されるので，意図した実体関連モデルの意味は，モデル中に示されることになる．

次に扱うカテゴリを集合のカテゴリに限定し，可換関係や普遍写像性を用いて意味的制約が如何にモデル中に表現されるかを具体例を通して見ていく．本稿では普遍写像性として終対象 (terminal object), 積 (product), 余積 (coproduct), 単射 (monomorphism), 引き戻し (pull-back), を扱うが，それらの詳細については他の文献 [3] を参照されたい．

集合のカテゴリにおける終対象は一点集合であるので，終対象 T からある実体 A への射 $T \to A$ は実体 A のある要素を特定するために用いられる．集合 A と集合 B の積はそれらの直積 $A \times B$ とその構成要素を特定する A, B への二つの射により，また集合 A と集合 B の余積は A, B の直和 $A + B$ と要素の埋め込みを表す A, B からの二つの射によりそれぞれ定義される．集合のカテゴリにおいて，単射とは一対一関数を意味しており，$f : A \rightarrowtail B$ と表す．さらに集合のカテゴリにおける $f : A \to C$ と $g : B \to C$ の引き戻しは $\{(s,t) | \ s \in A, t \in B, f(s) = g(t)\}$ なる二つの要素から成る集合と A, B への射により定義される．引き戻しの表記については図2と図3を参照されたい．

カテゴリ論の可換関係を用いて記述できる制約の例を図1に示す．この例は設備への人の割付けを表すモデルであり，「人と設備が同じ部署に属する時のみ割付けが可能である」という制約が存在するとする．この制約は図1の閉じた図式に可換関係があることを明示することにより保証される．

次に普遍写像性を利用して記述できる例を以下に示す．

図2は本の貸出，返却，登録，削除，貸出者リストの作成などの事象が発生する図書館データベースシステムであり，利用者として学生と職員の二つのタイプを持つ．利用者は職員と学生の余積として表され，本から利用者への射が本の貸出状況を示している．余積はこのように階層構造を持つ実体に適用でき，汎化や特化といったデータの抽象化技法をモデル中に記述することができる．ま

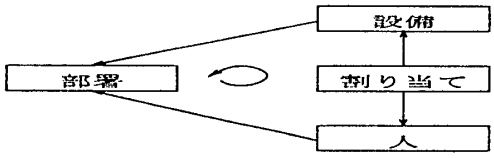

図 1: 可換関係を用いた制約の記述例

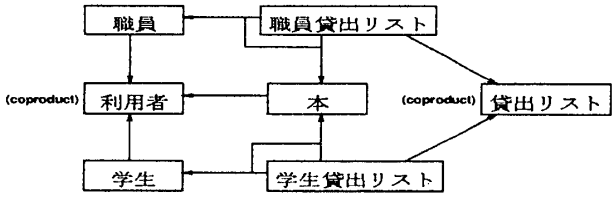

図 2: 図書館データベースシステムのモデル

た職員 (学生) 貸出リストは本から利用者への射と職員 (学生) から利用者への射の引き戻しとして表され，全体の貸出リストはそれらの余積となる．

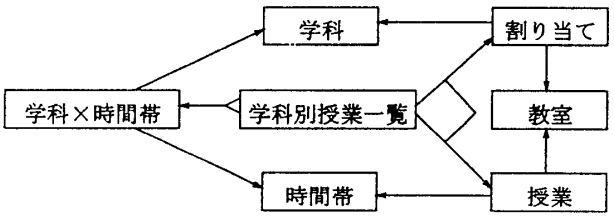

図 3: 京都大学工学部授業配当システムモデル

図 3 は京都大学工学部専門課程の時間割の状況をモデル化したものである．多対多の関係にある教室と学科を多対一の関係に直すために新たに割り当てを導入しており，そのインスタンスはある学科とその使用可能な教室を表している．図中の引き戻しにより各学科に配当される授業とその教室の一覧が得られ，そのインスタンスは学科，教室，授業の三つの要素から成る集合となる．また学科別授業一覧から学科と時間帯の積への射が単射であると明示することにより「同じ学科の同じ時間対に授業が重なることはない」という制約が満たされる．

3 支援システムの実装

まず本稿で提案しているモデルにより実際の問題を記述するために，カテゴリの普遍写像性・可換関係を記述できるエディタを作成する．そのエディタは描画機能のみならずモデル全体，つまり実体とその関係や普遍写像性を満たす部分などを内部表現化してその保存も行なう．さらにその内部表現と外部要求を入力とし，外部要求に応じてその問題の内部状態を変更・保存するプロトタイプシステムを構築する．図 3 の場合を例にとるとモデルの内部表現を参照し，そこに含まれている制約 (単射，積，引き戻し) を検証しながら，授業のある要素の追加や削除などの定義された外部要求に対して応答するようなシステムである．これは関数型言語 ML による問題の実現と同等の機能を持つシステムであると考えられ，ML プログラミングを行なうことなくプロトタイピングを支援することができる．このシステムにより同じ領域の問題のプロトタイピングは全て可能であると思われる．

このようにデータモデリングの段階で満たすべき制約を考慮に入れたプロトタイピングが可能であるのは，データモデル中にカテゴリ論の適用によりすでに制約が記述されているという点と，データモデルを有限集合のカテゴリとして解釈しているという点に大きく依存している．将来的にはエディタとプロトタイプシステムを連結させることにより画面上でプロトタイピングを行なうことも可能であると思われる．

4 おわりに

本稿では，実体関連モデルにカテゴリ論の普遍写像性・可換関係を適用することにより，データ間の制約が抽象的に記述できることを示した．これにより実世界の対象の構造を直観的に表現できる実体関連モデルに，カテゴリ論による曖昧性の少ない正確な記述能力を付加することができ，両者の長所を包含したモデル構築が実現できる．別の見方をするならば，開発初期の段階にできるだけ多くの情報をモデル中に埋め込るため，モデル自身を変換していくことでソフトウェア設計を行なうことになり，実現段階との連結度を高めることが可能となる．さらにそのモデルの利点を生かすために，外部入力を定義するだけでモデリングの段階でプロトタイピングが可能となるようなシステムを考案することにより，後戻りのないソフトウェア開発支援を目指した．

今後の課題としては，第一にエディタ，プロトタイプシステムの実装が挙げられる．さらには，他の問題領域への応用の可能性の追求などが考えられる．

参考文献

[1] 有澤博：意味データモデルの最近の動向，情報処理，Vol. 32, No. 9, pp. 1023–1031 (1991).

[2] Johnson, M. and Dampeny, C.: Category theory and information systems engineering, in *Proceedings of AMAST93*, pp. 95–103 (1993).

[3] Barr, M. and Wells, C.: *Category Theory for Computing Science*, Prentice Hall (1990).