

ドメイン分析・モデリングの利用法・研究法
ドメインに特化した概念モデルについて

廣田豊彦

九州工業大学情報工学部

1 はじめに

知的設計支援システムを開発するためには、設計対象の知識を分析、整理し、記述することが不可欠である。これに対して従来のソフトウェア工学では、OMT (Object Modeling Technique)[1]などのオブジェクト指向方法論や、各種のCASE (Computer Aided Software Engineering) ツールなど、汎用性の高いものが提案され、使用されてきた [2].

設計対象の知識は、そのドメインの専門家が直接記述することが望ましいが、ドメインの専門家が上述のような汎用性の高い方法論やツール、言語を使いこなすことは容易ではない。そこで、知識を記述するための概念モデル記述言語をドメインに特化し、ドメインの専門家に使いやすいものにする必要がある [3].

しかし、ドメインの専門家は言語設計の知識は持たず、また知識工学者やソフトウェア工学者はドメインの知識を持っていない。このため、それぞれが単独ではドメインに特化した言語は設計できない。そこで、著者らはこの3者が協調して知識ベースシステムを開発するプロセスを、4フェーズ、すなわち協働体制確立、プロトタイプ開発、実用システム開発、最適化のフェーズに分けることを提案した [4]. その上で建築設計のドメインを事例としてCAD実験システムを開発中である。汎用性の高い概念モデルを、建築専門家から獲得した知識を用いて、建築物の構造を表すための概念モデルに特化し、それをプログラミング言語C++で実現した第1フェーズはすでに終了した。現在、第2フェーズを実施中であり、その特化された概念モデルに基づいて、建築物の構成要素である部材や、その依存関係、部材の各属性間の制約を、建築設計の知識として記述するため、オブ

ジェクト指向かつ制約指向の言語 BDL (Building part Description Language) を規定した。

本稿では以下、2節で建築設計のためのドメインモデルについて述べ、3節でそのドメインモデルに基づく言語BDLの特徴について述べる。

2 建築設計のためのドメインモデル

著者らは以前に建築設計の特性を分析し、建築設計を統合的に支援するCADシステムとして、IBDS(Integrated Building Design System)[5]を開発した。このIBDSの建築部材モデルを採用し、ドメインの専門家でも容易に記述できるような言語の開発をすすめている。ここではIBDSの開発に際して、著者らが着目した建築設計の特性と、それに基づく建築部材モデルについて述べる。

2.1 建築設計の特性

建築設計は、構造設計や意匠設計などの専門の設計作業に分業化されているが、それら専門の設計作業は完全に独立に行えるわけではない。たとえば、意匠上の理由で柱が1本削除されたとすると、構造上は何らかの対策を講じる必要がでてくる。したがって、意匠の概要が決定した後に、構造の基本設計が行われるなど、適切な手順に従って設計がすすめられる必要がある。しかし、建築設計においては、設計変更が行われることもまれではなく、その場合に専門の設計作業相互の一貫性を保つことは容易ではない。

従来から建築設計の一部分を支援するCADは様々なものが開発されてきたが、上述のような一貫性を支援することはできない。そこで統合的な建築設計支援システムが必要とされているが、そのようなシステムでは、1) 建築オブジェクトを

表 1: 知識ベースシステムの開発サイクル

フェーズ	1	2	3	4
目標	協力体制の確立	プロトタイプの開発	実用システムの開発	最適化
ドメインの専門家の役割	知識表現の勉強	小規模例題の作成	知識ベースの開発	知識ベースの発展
知識工学者の役割	ドメインの勉強	知識ベースの調査	知識記述の確立	—
ソフトウェア工学者の役割	—	要求分析と仕様化	知識記述言語と処理系の開発	品質, 性能, 保守性の改善

一元管理する, 2) それぞれの専門の設計者に適切なドメインモデルを提供する, ことが要求される。

建築物の形状は3次元であるが, 設計者がいつでも3次元を意識して設計をすすめるわけではない。一般の建築物では, X, Y, Z, それぞれの方向の基準線ならびに補助基準線上に部材を配置し, そしてすでに配置されている部材に相対的にさらに部材が配置されて, 設計がすすめられる。したがって, 建築オブジェクトは, 1) 部材の集合として扱われる, 2) 部材は基準線や他の部材との相対的位置関係によって配置される, などの性質をもつ。

建築設計を知的に支援するためには, 設計対象に関する知識や, 設計作業に関する知識を整理し, 記述しなければならない。ここでは, そのような知識の一例として, 存在従属性と属性従属性を取り上げる。

存在従属性とは, ある部材が存在するためには, それに接続されている別の部材が存在しなければならないことを言う。逆にある部材が削除されると, それに依存する部材も削除されなければならない。たとえば, 梁はその両端の柱に依存し, 柱は下の階の柱に依存する。

属性従属性とは, ある部材の属性が別の部材の属性の値に依存することを言う。たとえば, 基準線を移動すると, その上にある柱の座標が再計算され, その柱が移動される。そして今度はその柱に架けられている梁の長さが再計算されることになる。このように属性の従属性を維持するためには, ある属性が変更されたときに, それに依存す

る属性を再計算しなければならない。

2.2 建築部材のドメインモデル

IBDSは属性モデリング[6]の考え方にに基づいており, 各部材は属性の集合として扱われる。これはソフトウェア工学におけるオブジェクト指向に対応しており, 各部材は一つのクラスとして記述され, 部材の属性はそのままクラス属性となる。各部材に対しては, 構造設計, 意匠設計など, それぞれの専門の設計毎に必要な属性があるが, それらをすべて一つのクラスの属性として統合的に扱う。一般にクラス属性は, 外部からはクラスメソッドを通じてのみアクセスされる。そこでそれぞれの専門の設計に必要なメソッドを準備することによって, 設計者は直接関係のない属性やメソッドを気にすることなく設計を行うことができる。すなわち, メソッドの集合によって, 適切な概念モデルを提供することができる。

部材の接続関係を表すために, IBDSではプラグとソケットの概念を導入している。一例として, 柱と梁の接続関係について説明する。梁の両端に柱と接続するためのプラグがあり, 柱には梁を接続するために4方向のソケットがある。このプラグとソケットを接続することによって, 柱と梁が接続されることになる。また, 柱などは基準線上に配置されるが, それを統一的に扱うために, 基準線も部材とみなす。基準線は任意の数のソケットを持つことができ, 必要な数だけの柱を配置することができる。

プラグとソケットは接続関係だけではなく, 存在従属性をも表している。すなわち, ソケットを

持つ部材が削除されると、そのソケットに接続されているプラグを持つ部材も削除されることになる。一方、プラグを持つ部材が削除された場合でも、ソケット側の部材は直接的な影響を受けない。

属性従属性も、プラグとソケットの接続関係に依存する。具体的にどのように属性が他の属性に依存するかはそれぞれの属性によって異なるので、属性の従属性は、クラスのメソッドあるいは従属性制約で記述する。

3 概念モデル記述言語

3.1 建築部材記述言語 BDL

建築設計の専門の設計作業の内、構造設計の分野について、部材の仕様を記述するために建築部材記述言語 BDL (Building parts Description Language)[7]を規定した。BDLは、建築物の構成要素である部材の接続関係や、部材の各属性間の制約を、建築設計の知識として記述するための言語である。

図1にBDLによる柱の記述例を示している。ここでは、柱のプラグとソケットの他に、柱に固有の属性 (USER_SPECIFIED) および他の属性に従属する属性 (DERIVED) を記述している。

3.2 言語の比較

建築設計のための概念モデル記述言語 BDL を、汎用性の高い概念モデルを適用した仕様記述言語 PSDL[8] およびプログラミング言語 C++ と比較するために記述実験 [9] を行った。この実験の結果を分析して、最小性、記述性、理解性、拡張性、形式性、適用性の6個の評価基準 [10] に従って、BDLとPSDL、C++を相互に比較した。その結果を表2にまとめている。表2から分かるように、BDLはその適用性を狭め、最小性を増すことにより、記述性・理解性や拡張性を増している。

BDLは、特別なプログラミング知識を必要としないので、設計者自身による設計知識の記述が期待される。これらのことから、それぞれの作業に特化した言語を規定し、設計者自身に知識を記述させることは、設計支援システムの要求分析に有効な手段である。

```
//
// column class (BDL)
// 柱

PART column ( part )
PLUG
  xbase_xline INTO xline::connect_col;
  ybase_yline INTO yline::connect_col;
  floor_zline INTO zline::connect_col;
  ceil_zline INTO zline::connect_col;
  lower_col INTO column::upper_col;
END
SOCKET
  upper_col TAKE column::lower_col;
  west_xgird TAKE xgirder::east_col;
  east_xgird TAKE xgirder::west_col;
  north_ygird TAKE ygirder::south_col;
  south_ygird TAKE ygirder::north_col;
END
USER_SPECIFIED
  xdiam INT DEFAULT default_xdiam; // X方向の柱の幅
  ydiam INT DEFAULT default_ydiam; // Y方向の柱の幅
  live_flag BOOLEAN DEFAULT Live; // Live or Dead
END
DERIVED
  xdisp INT // X方向の柱のずれ
  WHERE xdisp = IF xbase_xline->xlocate = East_end
  THEN wall_depth / 2 - xdiam / 2
  ELSE BEGIN
    IF xbase_xline ->xlocate = West_end
    THEN xdiam / 2 - wall_depth / 2
    ELSE 0
  END
  . . .
```

図 1: BDL 記述例

3.3 ドメインに特化した言語の特徴

本研究では、建築CADのプロトタイプの開発を目的として、要求分析ならびに仕様化を行った。その結果、応用ドメインに特化した概念モデル記述言語の特徴として以下の三つの点が明らかになった。これらの特徴はシステム開発サイクル (表1) のフェーズ3以降で、記述言語が改訂された場合でも保存されると考えられる。

第一に、ドメインモデルの構造が言語に反映されることである。ERモデルやオブジェクト指向モデルは汎用性のあるモデルであるが、それが応用ドメインのモデルとして最適というわけではなく、最適なモデルを選択することによって、言語の最小性、記述性、理解性を増すことができる。すでに述べたように、BDLにおいては、プラグとソケットが大きな役割を果たしている。

第二に、制約記述の集約ならびに制限がある。

表 2: 記述言語の評価

	C++	PSDL	BDL
最小性	×	△	○
記述性と理解性	×	△	○
拡張性	×	△	○
形式性	○	○	○
適用性	○	△	×

特定のドメインにおいては任意の制約が記述できる必要はなく、それを制限することによって、拡張性や形式性の面ですぐれた言語となる。BDLでは、部材の存在従属性制約が接続関係と一緒にになっており、さらに属性従属性制約が接続関係上に限定されている。

第三に、適切な計算機構の選択がある。データフロー計算が中心となるようなドメインでは単一代入が有利であるが、挿入、削除や属性変更が重要であるときには、単一代入はむしろ記述が困難になる。また制約記述を限定したことによって、必要な再計算をシステムが自動的に起動することが可能になる。このようにドメインにふさわしい計算機構を備えることによって、特に記述の最小性を高めることができる。

4 おわりに

本稿では、建築 CAD に特化した概念モデル記述言語 BDL の特徴を分析し、ドメインに特化した概念モデル記述言語のあり方について述べた。

BDL は、建築の専門家にも容易に理解でき、知識の追加や修正ができることを目指して開発をすすめている言語であり、建築物の構成要素である部材の接続関係に基づいて部材の存在従属性が自動的に仮定され、さらに属性従属性の記述も容易であるという特徴を持っている。

数多くの分野で知的設計支援の必要性が言われているが、それに対して単発的な自動設計ツールなどを開発するのではなく、設計知識を整理して活用するようなツールを開発する必要がある。そのためには本稿で示したようなドメインの分析とモデル化が不可欠である。従来、このような作業はドメインの専門家の作業と考えられていたが、著者らのグループの経験から、知識工学やソフト

ウェア工学の専門家がドメインの知識にまで踏み込む必要があると考えている。

参考文献

- [1] Rumbaugh, J., Blaha, M., Premerlani, W., Eddy, F. and Lorenzen, W.: *Object-Oriented Modeling and Design*, Prentice Hall (1991), (邦訳:『オブジェクト指向方法論—モデル化と設計』羽生田栄一監訳, トッパン, 1992年).
- [2] Bell, R.: Choosing Tools for Analysis and Design, *IEEE Software*, Vol. 11, No. 5 (1994).
- [3] Prieto-Diaz, R.: Domain Analysis for Reusability, in *Proc. IEEE Computer Software and Application Conference*, pp. 23-29 (1987).
- [4] 長澤勲: ポスト・マシプロダクション・パラダイムと IMS, 精密工学会知識工学と CAD 専門委員会第 23 回例会 (1993 年 6 月).
- [5] 長澤勲, 手越義昭, 牧野稔: IBDS: 建築物の統合化設計支援システム, 情報処理学会論文誌, Vol. 30, No. 8, pp. 1058-1067 (1989).
- [6] (社)日本設計学会: 高度技術化に対応する機械製図システムの標準化のための調査研究 (第 6 年度) 報告書 (1991).
- [7] 佐藤俊孝, 千原博司, 廣田豊彦, 橋本正明: 建築物設計支援のための概念モデル, 電子情報通信学会技術研究報告 (知能ソフトウェア工学), Vol. 93, No. 407, pp. 25-32 (1994).
- [8] 橋本正明: データ中心のプログラム仕様記述法, 井上書院 (1988).
- [9] 廣田豊彦, 千原博司, 佐藤俊孝, 橋本正明: 要求分析のための概念モデル記述言語に関する一考察, 電子情報通信学会技術研究報告 (知能ソフトウェア工学), Vol. 94, No. 130, pp. 1-8 (1994).
- [10] Liskov, B. H. and Zilles, S. N.: Specification Techniques for Data Abstraction, *IEEE Trans. Softw. Eng.*, Vol. SE-1, No. 1, pp. 7-19 (1975).