

統合設計支援環境におけるオブジェクト指向データベース 4V-5

石川 孝

べんてる株式会社 中央研究所

takashi@crl.pentel.junet

1. はじめに

近年、Drawing CADから問題解決を支援するCADへ脱皮するための計算機利用を高度化する研究において、新しい概念が提案されている。それらは設計過程と設計対象のモデリングに関するものおよび問題解決のための計算モデルに関するものであり、現在のCADシステムの課題を解決する、統合設計支援環境を構築するための基礎となるものと考えられる。本稿では、設計対象のモデリングに関して、統合設計支援環境を構築するための基礎となるオブジェクト指向データベースの試作について述べる。

2. 統合設計支援環境の要求機能

計算機を設計支援に用いるCADシステムは、計算機技術の発達にともなってその適用分野を拡大してきた。しかしながら、機械設計などの分野においては、一部の製品分野を除けばCADシステムの利用は不十分となっている。この基本的な理由として、設計作業の多様性や複雑性によってその体系的理解がなされていないことや、CADシステムを構築するための汎用性のあるプログラミング概念が未成熟であることが指摘されている[1]。この文献ではCADシステム構築の課題を以下のように整理している。

- (1) 設計論の展開：設計者の行為を統一的に理解し、CADシステム開発の基礎付けをすること（知識工学的手法を用いてCADシステムを構築するには、設計過程のモデル化が最も基本的な課題となる）
- (2) 設計過程のモデル化：設計者の設計行為そのものを計算機向きに適切にモデル化すること
- (3) 設計対象のモデル化：設計者が設計過程で使用するモデルは一般に多面的であり、工業製品を例にとれば、機能、強度、形状、組立構造、加工条件などを統一的に表現すること
- (4) 設計システムのソフトウェア支援環境：十分構造

化されていない知識を試行錯誤しながら整理し、体系化していくのに適したプログラミング概念の開発およびそのソフトウェア支援環境として実現すること

これらのCADシステム構築の課題は、実現すべき統合設計支援環境の要求機能として、つぎの3つの機能にまとめることができるものと考えられる。

- (1) 設計過程のモデル化機能：問題解決過程の支援、設計ノウハウのデータベース化
（Know-How型の知識表現に関する機能）
- (2) 設計対象のモデル化機能：設計対象知識のデータベース化
（Know-What型の知識表現に関する機能）
- (3) マンマシンインタフェース機能：概念の実体化、視覚化機能

現在までに提案されている概念としては、これらの統合設計支援環境の要求機能を実現するプログラミング・パラダイムとして、オブジェクト指向と制約指向の2つのパラダイムが基礎になるものと考えられる。

3. オブジェクト指向データベースの導入

設計対象のモデリングにおいて、オブジェクト指向概念は、オブジェクトが内部状態を持ち、外からのメッセージによって動作する計算実体であるという特徴のために設計対象を計算機上に実体として表現するのに好都合である。しかしながら、CADにおいてはオブジェクト指向概念は以下の理由によって不十分である。

設計過程はある制約（目標）を満たす構造（設計仕様）を決定する過程として一般化でき、最終的な設計解を得るまでの過程では制約の変更や構造の変更が動的になされると考えられる。このような設計過程においては、設計案が複数同時に存在することが通常であり、一度棄却された設計案が復活することも十分ありうることである。また、設計過程を問題解決過程と考えたとき、計算機に

よる自動化の度合は設計知識の完成度によって大きく異なり、完全な自動設計から全ての意思決定が人間によるものまで考えられる。したがって、少なからず人間の意思決定に頼る必要がある段階においては、人間の思考と並行的に計算機上に複数の設計案がオブジェクトとして永続的に存在する必要がある。

以上の考察からオブジェクト概念には、「オブジェクトは削除しないかぎり永続的に存在する」という新しい性質が必要であり、この機能を実現するデータベースとしてオブジェクト指向データベース [2] の導入が必要である。

4. データベース言語と処理系の試作

以下の設計方針に基づいてオブジェクト指向データベース言語 Q & A III を設計し、その処理系のプロトタイプを Lisp で試作した。

[設計方針]

(1) 対話型言語であること

(コマンドを読み、その結果を表示する)

(2) オブジェクト指向言語であること

(メッセージによってデータ操作を行う)

(3) 関数型言語であること

(返す値はやはりオブジェクトであること)

(4) 知的検索機能、選択・計算機能をもつこと

(設計支援のための必要機能をもつこと)

(5) クラス定義マクロ機能をもつこと

(設計言語による記述ができること)

Q & A III のオブジェクトはクラスとインスタンスとからなり、クラスはメタクラス CLASS のインスタンスであって、そのクラスのインスタンスの集まりを保持することによってオブジェクトの集合を表現している。

対象の構造は、クラス変数 SUPER によってクラス階層を表現し、インスタンス変数の値を他のクラスのインスタンスとすることによって部分全体関係を表現できるようにした。属性継承はクラス変数とメソッドについてのみ行う。また、インスタンス変数のデフォルト値として式を書くことでデータ導出を可能とした。

Q & A III のデータベースとしての特徴は、

(1) 変数の値をオブジェクトだけでなく、評価によ

ってオブジェクトを得る式に拡張したこと

(2) 値の評価をメソッドで定義することによって、

機能拡張ができるようにしたこと

である。

5. 例題による評価

クラス定義マクロによる部品データベースの記述例を図 1. に示す。

部品

SUPER= nil

@部品名

@下位部品=下位部品

@コスト=SUM ((@下位部品@コスト) : コスト)

下位部品

SUPER= nil

@部品名

@単価=部品: コスト; 部品名=@部品名

@数量

@コスト=@単価* @数量

図 1. クラス定義の記述例

ここで、@ は変数評価演算子で以下のように値を求める。

① 変数の値が nil でなければ、その値を返す

② 変数の値が nil で変数名と同じメソッドが定義されていれば、そのメソッドを実行して変数の値を変更した後、その値を返す

③ さもなくば、nil を返す

クラス定義マクロによって変数とメソッドを同時に定義できるので、クラス定義の記述はきわめて簡潔である。

6. おわりに

オブジェクト指向データベースの利点は、設計対象モデリングと問題解決の記述を統一した形式で簡潔に記述できることである。しかし、試作したデータベースはテキストだけのインタフェースしかないため、今後、設計者の直感に訴えるグラフィックインタフェースを作成する必要がある。また、実使用には処理スピードを向上するために、コンパイル形言語による実装とメソッドのコンパイラが必要である。

参考文献

[1] 長澤勳: 設計エキスパートシステム、情報処理、Vol.28, No.2, PP.187-196(1987)

[2] 田中克巳: オブジェクト指向データベース・システム、bit、Vol.20, No.6, PP.687-693(1988)