

オブジェクト指向プロトタイピングのための視覚的支援環境

—基本構想と支援ツール—

1N-5

吉田正弘* 阿部倫之* 中川秀敏** 加久間勝***

金沢工業大学* 金沢工業高等専門学校** 金沢女子大学***

1 まえがき

ソフトウェア開発の初期段階では、対象としている製品の概念的構造とその概念的動作を決定し、制約等も含めて要求仕様としてまとめていく。この段階において、実行可能な仕様を試行プログラミングによって作成し、概念構造と動作の確認や検証を行い、修正をリアルタイムに実施していくプロトタイピングが注目されている。プロトタイピングには、拡張や修正が頻繁に行われることから、概念をオブジェクトで表現し、この作業をビジュアルに支援する環境が検討されている。ビジュアルプロトタイピングとしては、アイコン化された視覚部品の組み合わせを主操作とする支援環境が提案されている(1)(2)。これらアイコン操作を主体にした方法は、エンドユーザを含めて容易に操作、カスタマイズできるが、新機能を持つアイコンの設計など、組み合わせ操作の範囲内で実現できない場合には、何らかの(視覚)言語を用いたプログラミングを必要としている。

ここで、プログラミングの熟練者の立場からビジュアルプロトタイピングを考えた場合、プログラミング(テキストまたは図式記法による)を主にして、GUIなどの直感的に作業したい部分や動作検証をアニメ表示などを用いてビジュアルに支援するのみで十分な場合も多い(LISPのような記号処理向き言語を使用している場合など)。本稿で述べるビジュアルプロトタイピング環境は、まず、具体的な支援ツールの構成について述べる。次に、基本構想のオブジェクト間の事実的、推論的関連について述べる。

2 基本構想

実世界に存在する実体(entity)を、何らかの知覚できる表現(identity)と能力(ability)を持つものとして捉え、実体をこの2つのプリミティブで構成する。本稿では、特に視覚オブジェクトと機能オブジェクトについて扱い、視覚、機能オブジェクトが一体化(カプセル化)されたものをフレームと定義しており、その概念図を図1に示す。

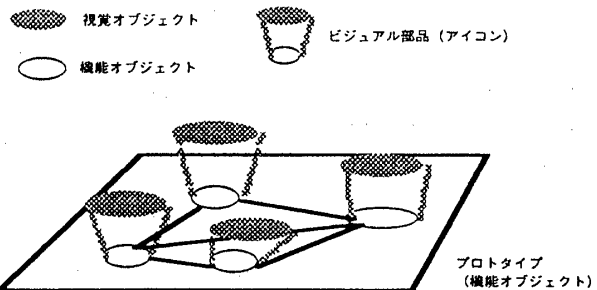


図1

また、具体例として、論理回路プロトタイプの例を図2に示す。図1のように、フレームは視覚オブジェクトと機能オブジェクトから構成されており、図2では、WIRE、NOT、ANDフレームが表現されており、それぞれに視覚オブジェクトと機能オブジェクトが貼り合わされている。

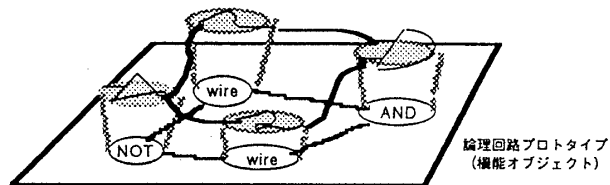
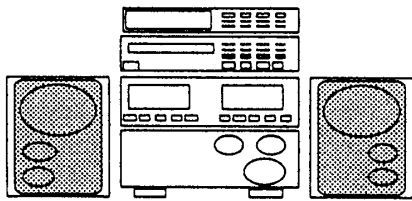


図2

先に述べたように、フレームは視覚オブジェクトと機能オブジェクトをカプセル化したものであり、そのフレーム定義には関係型言語のPROLOG言語を用いている。特に、視覚オブジェクトと機能オブジェクトのカプセル化については、組み込み述語”貼り合わせ”を用意している。また、機能オブジェクト間(外見上はフレーム間)の関連についてもPROLOG likeな関係の記述を行う。図3にステレオコンポの例を示す。このコンポは、チューナー、CDプレイヤー、カセットデッキ、アンプ、スピーカーから構成されており、それぞれが、フレームとして既に定義されているものと想定する。各々の機器は、接続されて初めてコンポとして成立ち、また、その接続方法にも、様々なバリエーションが存在し、それはユーザの自由意思に任されている。このように、確定していないオブジェクト間の関連を意識した場合、関係を記述するだけでよいPROLOGは有効であると考えられる。例えば、スピーカーから聴こえる音をヘッドフォンから聴きたい場合は、内部的な処理は別として、”接続(アンプ、ヘッドフォン)”という事実的な関係を追加すればよく、逆に、ヘッ

A Visual Environment for Object-Oriented Prototyping
Masahiro YOSHIDA
Kanazawa Institute of Technology
7-1, Ohigigaoka, Nonouchi-machi, Kanazawa-south area,
Ishikawa, 921, Japan

ドフォンを外す時は、この関係を削除すればよい。次に、PROLOGの特徴である規則を用いることにより、推論的に関係を導出することができ、冗長な事実的関係を記述する必要がなくなる。例えば、“接続(チューナー、アンプ)”と“接続(アンプ、チューナー)”は、関係“接続”では同意であるが、PROLOG処理系では、引数の順序により区別されるので、規則を追加することにより、推論的に関係を導出することができる。また、それぞれの動作について、フレーム間で成り立っていなければならないような制約事項も規則節を追加することにより、記述することができる。



(接続 ?X ?Y) :- (接続 ?Y ?X)
 (音が出る ?X ?Y) :- (接続 ?X アンプ), (接続 アンプ ?Y)
 (録音する ?X カセットデッキ(カセット2)) :- (接続 ?X カセットデッキ)
 (録音する ?X カセットデッキ(カセット2)) :- (接続 ?X アンプ), (接続 アンプ カセットデッキ)

図3

3 支援ツールの構成

支援ツールのシステム構成は、図4に示すように大きく5つのモジュールから構成されており、CLOS (Common Lisp Object System) を用いて、Macintosh上に実装中である。

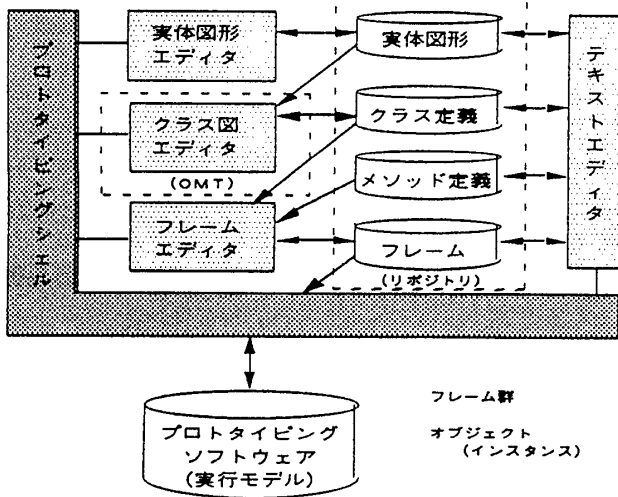


図4

・ 実体図形エディタ

視覚オブジェクトを表現する実体図形を描画するためのエディタであり、実体図形は基本図形(円、楕円、円弧、直線、長方形、多角形)の組み合わせで構成される。

・ クラス図エディタ

オブジェクトのクラス設計をOMTの図式記法で行うための支援ツールであり、このツールを用いて機能クラスと視覚クラスの設計を行う。

・ フレームエディタ

機能クラスと視覚クラスを貼り合わせ、機能合成を支援する。その際に、フレームを管理するためのクラス(フレームクラス)を定義する。

・ テキストエディタ

機能クラスの具体的な振舞いを定義するエディタである。また、クラス図エディタやフレームエディタのプロダクトを直接に、書き換えることを支援するエディタである。

・ プロトタイピングシェル

実体図形エディタ等の支援モジュール群を取りまとめているモジュールであり、各エディタのプロダクトを利用して、フレームの編集を行う。また、編集されたフレームをモデルウィンドウに複写することにより、実効可能なアイコン(フレームのインスタンス)が生成され、プロトタイプが作成される。

4 おわりに

本稿では、オブジェクト指向プロトタイピングのための視覚支援環境について、基本構想と支援ツールの構成について述べた。テキスト(図式記法)ベースのプログラミングから出発して、必要に応じてオブジェクト単位のビジュアル化を図る専門家向きではあるが、アイコン化されたフレームの組み合わせについては、非専門家でも容易に操作できる。今回の支援ツールでは、さらに、機能オブジェクトのプログラミングに図式記法を導入しているのと、関連と制約の宣言的記述、およびオブジェクトの貼り合わせを支援しており、プロトタイピング効率の向上が期待できる。

・ 謝辞

本研究の一部は、株式会社 日立製作所情報通信事業部ソフト技術センタ殿の委託研究の一環として行われたものである。

・ 参考文献

- (1) 湯浦克彦, 高橋久: ODETTE: オブジェクト指向CLOSをベースとした設計支援構築環境, 「オブジェクト指向ソフトウェア技術」シンポジウム, 1991
- (2) 阿部倫之, 吉田正弘, 中川秀俊: オブジェクト指向プロトタイピングのための視覚的支援環境, 情報処理学会ソフトウェア工学研究会, SE99-18, 1994
- (3) J.Rumbaugh: OBJECT-ORIENTED MODELING AND DESIGN, 羽生田栄一 訳: オブジェクト指向方法論, トッパン, 1992