

3F-3

オブジェクト間結合にデータ伝播機構を用いた
視覚的オブジェクト指向言語

中田 秀男 杉本 明 阿部 茂

三菱電機(株) 中央研究所

1 はじめに

オブジェクト間の結合にデータ伝播機構を適用した視覚的オブジェクト指向言語を提案する。オブジェクト部品をアイコンで表示し、グラフィカルエディタを用いて2次元的に配置する。データフローを表す線で部品間の結合を記述することによって視覚的にプログラミングを行う。オブジェクトの集合表現も可能である。

本稿では、本言語の設計方針、言語仕様を述べ、ユーザインターフェース構築への適用例を説明する。

2 設計方針

Pict、Show and Tell等の従来のグラフ・アイコン指向言語[1]は部品アイコンの機能が小さく、一般的なソフトウェアを組み立てるには不向きであった。多くの場合、高性能部品の生成には視覚的言語よりテキスト言語を使った方が効率的である。本言語では、C++等の既存のオブジェクト指向言語で部品を記述して視覚的に結合する仕組みにより、高い記述性を目的とする。

また、C++、Smalltalkのような既存のオブジェクト指向言語では、オブジェクト部品の規模が大きくなると、部品を熟知して組み立てる必要があった。本言語では、視覚的表現を用いて簡明な部品結合方法を実現することにより、理解しやすいプログラミングを目的とする。オブジェクト部品の規模の大小に係わらず結合方法が容易に解るように、部品間のインターフェースを決まった型に限定する。

3 言語仕様

3.1 オブジェクト

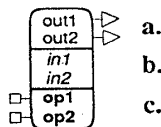


図1: オブジェクトの3層構造

a. 外部属性

他のオブジェクトへ影響を及ぼす属性名を列挙する。
外部属性が変更されると、データフローを表す線で接続された別の(あるいは自己)オブジェクトのメソッドが起動する。

b. 内部属性(省略可能)

オブジェクトの仕様を表す定数を列挙する。
内部属性の値は、実行前に設定する。

c. メソッド

オブジェクトに関する操作を表すメソッド名を列挙する。
メソッドの引数は1つとする。

Visual Object-Oriented Language with Data-Flow Mechanism
Hideo NAKATA, Akira SUGIMOTO, Shigeru ABE,
Central Research Laboratory, Mitsubishi Electric Corporation

外部属性を変更し、他のオブジェクトのメソッドの起動を行う。
ただし、メソッドの実行完了まで次のメソッドは起動されない。

3.2 オブジェクト間結合(基本形)

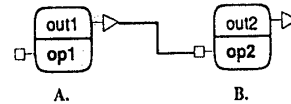


図2: オブジェクト間結合(基本形)

インスタンスの外部属性とメソッドをデータフローを表す線で接続する(図2)。外部属性とメソッドの引数のデータ型が異なる場合は接続不可能とする。ただし、外部属性のデータ型が、メソッドの引数のデータ型の導出クラスの場合は接続可能とする。

オブジェクトAの外部属性が変更されると、オブジェクトBのメソッドにその値が渡され、値を渡されたメソッドが起動する。

3.3 特殊なオブジェクト間結合

記述スペース削減、視認性向上のため、3.2の基本形とは異なるオブジェクト間結合の方式も用意してある。

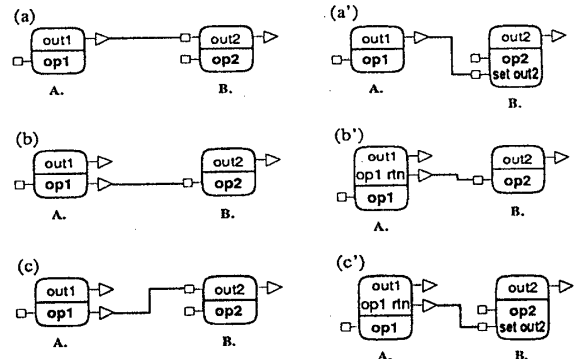


図3: オブジェクト間結合(特殊形)

3.3.1 外部属性 - 外部属性

図3(a)では、オブジェクトAの外部属性が変更されると、オブジェクトBの外部属性がオブジェクトAの外部属性と同じ値になる。オブジェクトBに外部属性を設定するメソッドを用意すると、図3(a')のように3.2のオブジェクト間結合で表現可能である。

3.3.2 メソッド - メソッド

図3(b)では、オブジェクトAのメソッドが実行されると、そのメソッドのリターン値がオブジェクトBのメソッドに渡され、値を渡されたメソッドが起動する。オブジェクトAにメソッドのリターン値を表す外部属性を用意すると、図3(b')のように3.2のオブジェクト間結合で表現可能である。

3.3.3 メソッド - 外部属性

図3(c)では、オブジェクトAのメソッドが実行されると、そのメソッドのリターン値がオブジェクトBの外部属性に代入される。同様に、図3(c')のように3.2のオブジェクト間結合で表現可能である。

3.4 オブジェクト集合

3.4.1 集合領域

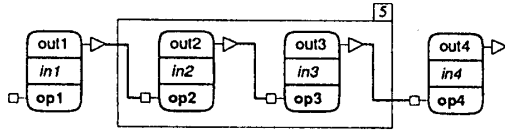


図4: 集合領域

図4で、2つのオブジェクトを含む長方形で囲まれた領域を、集合領域とする。集合領域中のオブジェクトはすべて、複数のオブジェクトを含むオブジェクト集合である。また、オブジェクト集合は集合領域のみに存在する。

1つの集合領域に存在するオブジェクト集合のオブジェクト数は、どのオブジェクト集合に関しても同じである。オブジェクト数は、集合領域の右上に記述する。

3.4.2 データフロー集合

集合領域中のデータフローはすべて、複数のデータフローを表すデータフロー集合である。そのデータフロー数は集合領域の右上に記述してある数である。

集合領域の境界では自動的に、集合領域の外から中に入るデータフローは単一のデータフローからデータフロー集合に変更され、集合領域の中から外に出るデータフローはデータフロー集合から単一のデータフローに変換される。図4では、外部属性out1の値が5つのメソッドop2のすべてに渡され、5つの外部属性out3の値のうち変更が生じた値がメソッドop4に渡される。

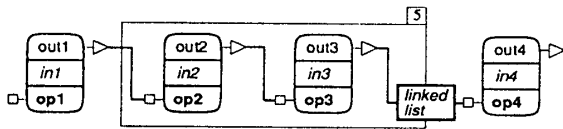


図5: データフローの変換方法指定

図5のように、単一のデータフローとデータフロー集合の変換方法(和・平均等の数値演算、リスト化etc)を指定する方法もある。変換方法を示したアイコンを集合領域境界に配置する。

3.4.3 オブジェクトの多次元集合

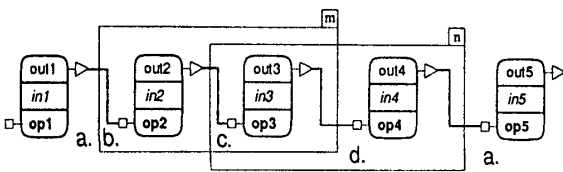


図6: オブジェクトの多次元集合

図6のようにオブジェクトの多次元の集合の記述も可能である。オブジェクトおよびデータフローの数は、領域aでは1個、領域bではm個、領域cではm×n個、領域dではn個である。

4 適用例

	a	b	c	d	total
1					
2			A		B
3					
total			C		D

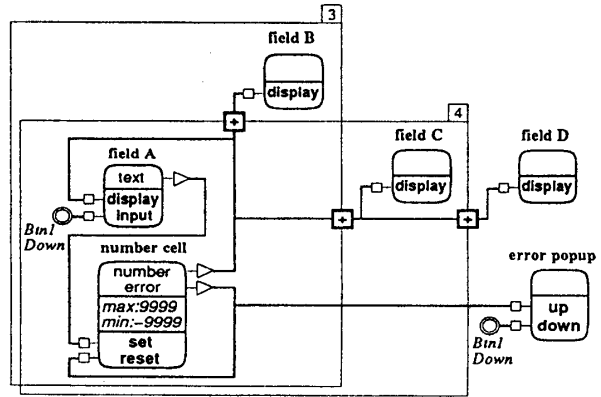


図7: テーブル操作への適用例

図7上図のような簡単なスプレッドシートについて考える。Aの部分に数字を入力すると、横の合計がBに、縦の合計がCに、全合計がDに表示される。また、Aは4桁までの数字が入力でき、5桁以上の数字を入力するとエラーの表示がポップアップする。

図7下図は本言語で記述した例である。

Aの部分のうち1つをマウスボタン押下すると、対応するオブジェクトfield Aのメソッドinputが起動し、入力待ち状態となる。入力が完了すると、入力文字が外部属性textに代入される。すると、オブジェクトnumber cellのメソッドsetが起動する。入力文字を数値に変換し、max、minの範囲内ならば外部属性numberに代入する。範囲外ならば外部属性errorにエラーメッセージを設定する。

外部属性numberが変更されるとその値がオブジェクトfield Aのメソッドdisplayに、横の合計がオブジェクトfield Bのメソッドdisplayに、縦の合計がオブジェクトfield Cのメソッドdisplayに、全合計がオブジェクトfield Dのメソッドdisplayに渡される。各メソッドは渡された値を画面表示する。

外部属性errorが設定されると、オブジェクトerror popupのメソッドupが起動しエラーがポップアップ表示され、マウスボタン押下すると、メソッドdownが起動し表示は消去される。また、オブジェクトnumber cellのメソッドresetが起動し、外部属性numberに0を代入し外部属性errorをクリアする。

5 おわりに

今後は、実システムで実現するとともに、実際のソフトウェア生産への適用を重ねて、より良い仕様に改善していく予定である。

参考文献

[1] Nan C.Shu, "Visual Programming", Van Nostrand Reinhold, 1988