

2J-1

ビジュアル・プログラミングのための
制御構造のオブジェクト化と視覚化

秋山憲一 中村裕治 上田賀一

茨城大学

1 はじめに

近年では、エンドユーザ・コンピューティングが主流になりつつあるが、現在のコンピューティング・スタイルは誰でも熟考することなくコンピュータを扱えるという状況には程遠い。そこで新しいプログラミング・スタイルの必要性が叫ばれ、その一つとしてビジュアル・プログラミングを挙げることができる。

プログラミング言語の進展は、低級、高級、超高級、超高位の四つの水準に分けられるが[1]、水準が上がってもプログラムの構造は一次元的かつ文章的であるのは変わらない。これに対し、ビジュアル・プログラミング言語は絵の持つ特性により、これまでのプログラミング言語の伝統から離脱している。

さらに、ビジュアル・プログラミング言語は設計原理に基づき、以下の三つに大きく分けられる。

1. アイコンがプログラミングの中心的な役割を担う言語
2. 紙の上で利用されているフローチャートやダイアグラムを取り入れた言語
3. 視覚的表現が言語の不可欠な部分を担う言語

ここでは、プログラミング言語として比較的に利用しやすいと思われる1.の方法を用いて言語を構成することとする。

さらに、プログラムを視覚的に記述するためにオブジェクト指向の考え方を導入する。プログラミングにとって必要である制御構造は、既存のオブジェクト指向言語では未だにオブジェクト化がなされていない。ここではその点に目をつけ、制御構造のオブジェクト化と視覚化について述べる。

2 制御構造のオブジェクト化

アイコンをプログラミングの中心的な役割とすることは、言語の視覚化に向いている。しかし、ここで問題とされるのは、アイコンが命令(ステートメント)を、アイコンを結ぶ線が制御構造を表してしまい、視覚的表現を活かした言語を構成したつもりがフローチャート形式の言語を構成しただけになってしまう可能性があるということである。

そこで、オブジェクト指向の考え方を取り入れ、制御構造の視覚化を考える。

現在のオブジェクト指向プログラミング言語はみなテキスト記述形式である。しかし、テキストでは線形的な描写しかできないので、逐次処理のプログラムには適しているが、オブジェクト指向のように処理が並列に実行されるようなプログラムの記述には最適であるとはいえない。

ビジュアル・プログラミング言語であれば、プログラムを平面的な広がりを持って記述できるので、並列処理のプログラムの記述に適している。また、アイコンでオブジェクトを、アイコンを結ぶ線でメッセージの流れを表すことにすれば、テキスト言語のチャートになってしまうという可能性がかなり少なくなる。さらに、このような形のオブジェクト指向言語ならば、今までのテキスト形式のオブジェクト指向言語よりも容易にオブジェクト指向プログラミングというものを理解できるもの考える。

3 制御構造の視覚化

前節で述べたように、プログラムの制御構造にオブジェクト指向の概念を取り入れ、制御構造の視覚的表現法を考える。制御構造のオブジェクト化についてはオブジェクト指向プログラミング環境 Smalltalk のもの

を参考にした [2]。また、複数のオブジェクトをまとめて一つのオブジェクトのように扱えるようにする。これをブロックオブジェクトと呼ぶ。ただし、ブロックオブジェクトは見かけ上一つのオブジェクトとして存在するだけで、実際には複数のオブジェクトとして動作する。

プログラムの制御構造には、接続、選択、反復の三つの要素がある。しかし、オブジェクト指向の概念を取り入れたため、プロセスの流れを示す接続についてはオブジェクト間のメッセージのやりとりで既に記述されているので新たに考える必要はない。

選択、反復の実現図をそれぞれ図1、図2に示す。どちらも、選択（反復）、条件、ブロックの各オブジェクトがまとまってブロックオブジェクトを形作っている。

選択オブジェクトは条件オブジェクトの判断により、ブロックオブジェクトを実行する。選択オブジェクトが外からのメッセージを受けると、条件判断のため条件オブジェクトへメッセージを渡す。条件オブジェクトからリターンメッセージが返ってくると、リターンメッセージの引数の値が真ならば、ブロックオブジェクトを実行させるためブロックオブジェクトへメッセージを送る。ブロックオブジェクトからリターンメッセージが戻った時点処理を終了し、リターンメッセージを返す。条件オブジェクトからのリターンメッセージの引数が偽ならば、その時点で処理を終了し、リターンメッセージを返す。

反復オブジェクトはブロックオブジェクトを繰り返して実行させる。メッセージの流れは選択オブジェクトとほぼ同じであるが、ブロックオブジェクトからリターンメッセージが返ってきても処理を終了せず、再び条件オブジェクトへメッセージを渡し、動作を繰り返す。条件オブジェクトからのリターンメッセージの引数の値が偽になった時点で処理を終了し、リターンメッセージを返す。

条件オブジェクトはメッセージを受けると、その引数リストにより条件を判断し、引数真 (True) または偽 (False) を伴ったリターンメッセージを返す。

4 まとめ

本研究では新しいプログラミング・スタイルとしてビジュアル・プログラミングを取り上げ、プログラムを視覚的に記述するための制御構造のオブジェクト化と視覚化について報告した。今までのオブジェクト指向プログラミング言語では実現されていなかった制御構造のオブジェクト化によって、本来のオブジェクト指向により近い形でプログラムの記述ができる言語を

作り出せると考える。

今後の課題として、今回オブジェクト化および視覚化した制御構造を用いたビジュアル・プログラミング言語を実現することがあげられる。

参考文献

- [1] Nan C. Shu, 西川博昭 訳：ビジュアル・プログラミング、日経BP社 (1991)
- [2] 加藤木和夫：Smalltalk/Vによるオブジェクト指向プログラミング、日刊工業新聞社 (1990)

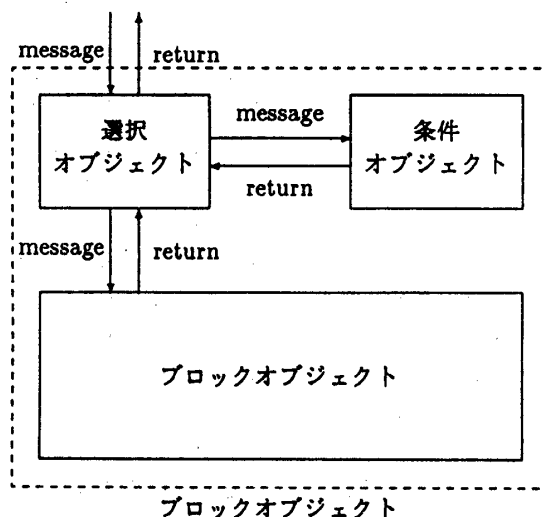


図1: 選択構造のオブジェクト化の実現

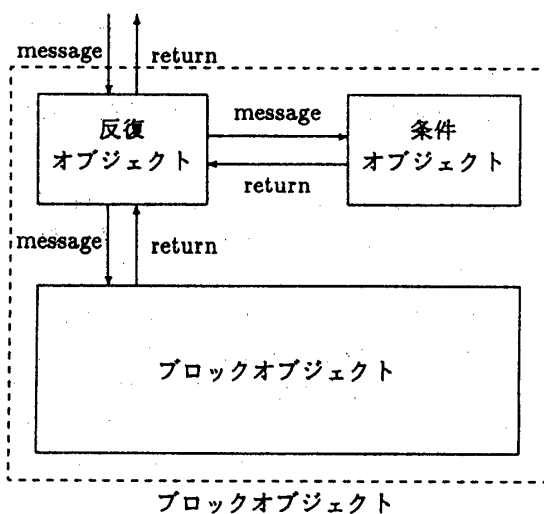


図2: 反復構造のオブジェクト化の実現