5D-6

並列オブジェクト指向言語ＤＵＥ

福井　眞吾

日本電気（株）Ｃ＆Ｃシステム研究所

１．はじめに

ＤＵＥは、①オブジェクトを単位とする並列実行、②返事の到着を待たないメッセージパッシング機構、③返事を受信する変数を指定する構文、④accept文による選択的受信、を特徴とするオブジェクト指向言語である。ＤＵＥはFranz Lisp上に構築されており、現在、並列処理アルゴリズムの記述および分散型オフィスシステムのプロトタイピングに使用されている。本稿ではＤＵＥの設計方針と言語仕様の概要について述べる。

２．設計方針

ＤＵＥの設計にあたって、次の３点を基本方針とした。

①メッセージ送信以外の特殊な機構を導入しない。

②①を逸脱しない範囲で、できるだけ従来からある手続き型言語の記述方式に近づける。

③Lispのデータ構造をオブジェクトとして扱う。

Smalltalkに代表されるオブジェクト指向言語では、メッセージ送信という枠組の他に、手続き呼び出しとそのリターンバリュー，プロセスといった概念が導入されている。ＤＵＥではオブジェクト間の通信手段としてメッセージ送信だけを仮定し、メッセージ送信の評価値は仮定しない（①）。

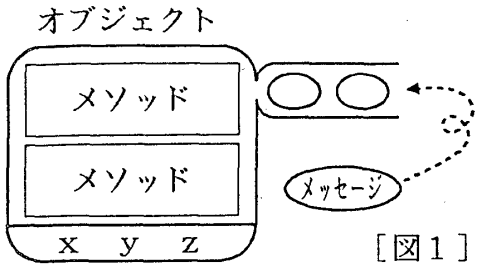
しかし、従来の手続き型言語から極端に離れたシンタックスにするのではなく、メッセージ送受信に還元できる範囲で式と評価値を用いた記述を可能にする（②）。

オブジェクト指向言語を構成するには、数，文字，配列等の基本オブジェクトが必要である。ＤＵＥではすべてLispのデータ構造を利用する。ただし、Lispのデータ構造もユーザが定義するオブジェクトと同様にＤＵＥのオブジェクトとして取り扱えるようにする（③）。

３．ＤＵＥの概要

３－１．オブジェクトのモデル

オブジェクトは、インスタンス変数，メソッド，メッセージキューから構成される。（図１）



［図１］

オブジェクトにはメッセージ待ち状態（非活性状態）とメソッドを実行している状態（活性状態）がある。非活性状態のときに到着したメッセージはすぐに受信され対応するメソッドが起動される。活性状態の時に到着したメッセージはメッセージキューで待たされる。メッセージの受信は通常到着順で行われるが、accept文[2]が実行されると、指定されたメッセージのみが受信される。

メソッドは常に１つだけが実行される。ＤＵＥでは、オブジェクト間の並列性によって並列処理を実現する。

３－２．メッセージ送信

メッセージは、起動するメソッドを指定するキー部とメソッドに渡される引数部から構成される。

<message> ::= <key> {<arg>}...

メッセージ送信文には次の２種類がある。

［継続なし］（<target> <message>）

［継続付き］（<target> <message> :c <arg>）

いずれの場合にも<target>で指定されたオブジェクトに<message>が送られる。継続付きのメッセージを受信したオブジェクトでは'continuation'という疑似インスタンス変数に:c部の引数が束縛される。継続なしの場合にはnilが束縛される。メッセージを送信したオブジェクトは

Concurrent Object Oriented Language ＤＵＥ

Ｓｈｉｎｇｏ　ＦＵＫＵＩ

ＮＥＣ　Ｃｏｒｐ．

<target>がメッセージを受信するのを待つことなく処理を続行する。

３－３．返事の受信

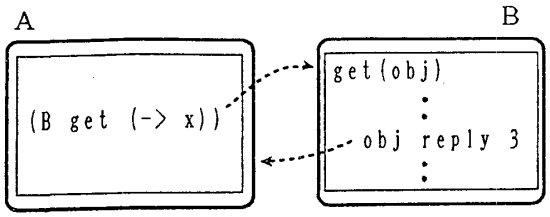
メッセージに対する返事の受け取りには、futureオブジェクトを利用する。たとえば、インスタンス変数ｘで返事を受け取る場合、future生成子'(->)'を用いてｘのfutureオブジェクトを生成する。

```
(-> x)
```

このfutureオブジェクトを引数として相手に送信する。相手のオブジェクトが、futureオブジェクトに

```
reply <ans>
```

を送信すると、ｘに返事<ans>が束縛される。（図２）



［図２］

この束縛は、accept文を用いることによって通常のメッセージ送信で実現される。(-> x)によってfutureオブジェクトを生成するとｘは束縛待ち状態になる。値を参照しようとすると、値が束縛されるまで実行が中断される。

３－４．選択的受信

有限バッファ等のメッセージ受信順序に制約のあるオブジェクトの場合到着順だけではうまく記述することができない。ＤＵＥではメッセージキューの中から指定したパターンにマッチするメッセージを受信するaccept文を導入している[2]。accept文は次のような形をしている。

(accept {<pattern>}... {:default (<message>)})

<pattern>には受信しようとするメッセージパターンを指定し、:default部には一致するメッセージがなかった場合に受信するメッセージを記述する。:default部のみを指定したaccept文は自己呼び出しになる。

３－５．継続の利用

メソッドには、返事を１つ返す事を目的として利用されるものが多い。３－３．の方式を用いれば簡単に実現できるが記述が煩雑になる。そこでＤＵＥでは返事を受信するfutureを引数として渡す標準的な方法として継続を用意し、さらに次の略記法を用意している。

① (x <- <target> <message>)
　　=> (<target> <message> :c (-> x))

② (^ <ans>)
　　=> (continuation reply <ans>)

３－６．メッセージ送信文の式としての利用

従来の手続き型言語では関数呼び出し文を式として利用できるが、メッセージ送信文は文であって評価値は存在しない。そこで式を書くべき所に継続なしのメッセージ送信文がある場合には、暗黙のインスタンス変数を用いて展開する。

```
(A put (B get)) => (B get :c (-> tmp))
                       (A put tmp)
```

これによってメッセージ送信文を式として扱える。

３－７．Lispのデータ構造の取り扱い

<target>がLispのデータ構造である場合には、<key>と<target>を入れ換えて、

(<key> <target> <arg>...)

をLisp関数として評価する。また、一般のオブジェクトを要素とするLispのデータ構造を生成できるように'lisp'というダミーのオブジェクトを用意している。

(lisp cons A B) => (cons A B)

４．まとめ

メッセージ送信と選択的受信機構を基盤とする並列オブジェクト指向言語ＤＵＥについて報告した。ＤＵＥではこの２つの機構によって返事の受信を実現している。また、継続付きと継続なしのメッセージ送信文，略記法，メッセージ送信文の式としての利用によって、メッセージのみによる通信という枠組から外れる事なく、従来の関数呼び出しと同様な記述が可能になっている。

<参考文献>

1.福井 眞吾：オブジェクト指向並列処理言語によるプログラミング，第２８回情処全国大会，昭和５９年３月
2.福井 眞吾：並列オブジェクト指向言語における同期制御方式，第３２回情処全国大会，昭和６１年３月