

1V-02 グループ化可能な移動プログラムのための高階言語処理系の実現

高橋 正樹	神林 靖	滝本 宗宏	原田 賢一
慶應義塾大学大学院	トレド大学	東京理科大学	慶應義塾大学
理工学研究科	電気工学計算機科学科	理工学部	理工学部
計算機科学専攻		情報科学科	情報工学科

1 はじめに

近年、分散環境において、移動能力をもつ自律的プログラムの需要が増大している。移動プログラムは、プログラム自体が移動するので、遠隔地に存在する資源利用の際に有効であり、この移動プログラムのための様々な計算モデル [1, 2] が提案されている。それらには、移動プログラムを利用した分散システムを構築するに当って有用なアイデアが含まれている。一方、オブジェクトや関数自身を引数や返却値として受け渡すことができる高階言語 (higher order language) に対する研究も活発に行われている。この言語の特徴を利用することによって、移動プログラムの記述が簡潔になる。

本稿では、そのような移動プログラムのための高階言語処理系の実現について述べる。

2 概要

ここで提案する移動プログラムのための言語処理系は、代表的な高階関数型言語である Objective Caml [5] を基盤としており、本研究の O'Cam1 の拡張を DOCaml (Distributed Objective Cam1) と呼ぶことにする。

本言語のオブジェクト移動は遠隔サイトにおける複製によって表現し、分散言語 Obliq [3] において提案された分散字句有効範囲 (distributed lexical scope) を採用したことによって自由変数の束縛は静的に決定される。次に概要を示す。

オブジェクトの移動

オブジェクトの移動は、遠隔適用 (remote apply) によるオブジェクトの複製および再束縛によって表現する。その例を次に示す。

```
let o = new point () ;;  
let o = clone o at site2 ;;
```

ここでは、point クラスがすでに定義されていると仮定している。new によって point クラスからオブジェクトを生成し、それを o という名前に束縛することによって

¹ "Implementation of a Higher-Order Language for Mobile Objects with Dynamic Grouping," Masaki Takahashi, Keio University, Yasushi Kambayashi, University of Toledo, Munehiro Takimoto, Science University of Tokyo, Ken'ichi Harada, Keio University

識別可能にしている。下段の式が評価されると、site2 において clone o が遠隔適用される。そして、元のサイトへの返信結果は再度、o として束縛される。オブジェクト o の複製の実体は site2 に存在し、返信結果はそのプロキシに変更されたものである。したがって、オブジェクト o は、site2 へ移動したことになる。

メソッド呼出しは、オブジェクトが局所または遠隔に存在するかによって構文を区別する必要はなく、オブジェクトの実体が存在する環境において実行される。

オブジェクトのグループ階層化

オブジェクトのグループ階層化は、任意のオブジェクトを他のオブジェクトの中に移動させることによって実現する。そのために本言語では、関数適用を行う場所として、任意のオブジェクトを指定することができる。本言語では、サイト自身も特殊なオブジェクト型として取り扱う。簡単な例を次に示す。obj0, obj1 は任意のオブジェクトである。

```
let obj0 = clone obj0 at obj1 ;;  
let obj1 = clone obj1 at site2 ;;
```

特別形式 clone は、引数で受信したオブジェクトだけではなく、それと内包関係にあるすべてのオブジェクトも複製する。したがって、最後に示した式は、obj1 の site2 への移動 (複製) に伴い site2 も移動 (複製) されることを意味する。

また、移動オブジェクトは、自分自身を内包しているオブジェクトのメソッドを、自己参照を用いたメソッド呼出しによって利用することができる。

3 実現

DOCaml 言語処理系の実現は、次のとおりである。

- (1) 独自のバイトコードインタプリタを有する Scheme48 言語処理系の分散用拡張の 1 つである Kali Scheme [4] をさらにオブジェクト指向を支援するように、機能拡張を施した。Scheme48 は、並行環境における独自のマルチスレッドシステムを有し、拡張性の高い言語処理系である。それに、O'Cam1 ベースのオブジェクト指向のオブジェクト機構、分散 (局所) オブジェクトおよびそれらの通信を管理する機構、さらにオブジェクト移動のための機構などを導入した。
- (2) DOCaml コードを拡張

Scheme48 コードに翻訳する変換系を実現した。翻訳されたコードは、Scheme48 バイトコードインタプリタによって実行される。本研究では、この2つの機構の組合せによって、DOCaml 言語処理系を構成している。

DOCaml 言語処理系における移動オブジェクトシステムの構成は図1に示すとおりである。

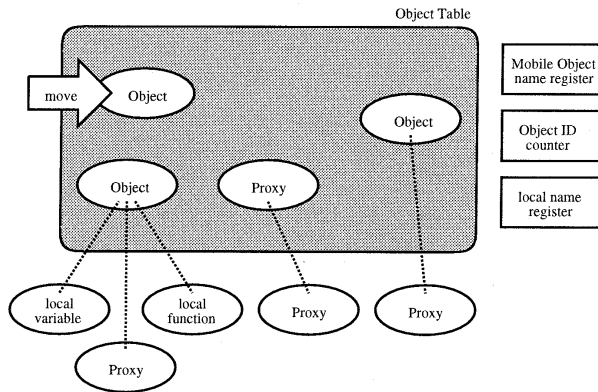


図1: DOCaml 移動オブジェクトシステムの構成

すべてのオブジェクトのデータ構造は、サイト毎に用意された Object Table に格納される。let 文による束縛の際には、Object ID counter からオブジェクト ID(OID) が与えられ、名前は local name register に自動的に登録される。オブジェクトは、OID とそれが生じたサイトの番号とのペアによって識別される。サイト利用者は、実行環境にあるプロキシを介してオブジェクトを使用する。local name register には、そのサイト固有の変数と関数の名前も登録され、分散字句有効範囲の実現に利用される。

他のサイトから移動してきたオブジェクトも、Object Table に格納されるが、それだけでは他のオブジェクトに影響を与えない。ただし、Mobile Object name register に登録されたオブジェクトのプロキシを介することによって、そのサイトのオブジェクト内部への移動が可能であり、そのメソッドを利用できる。そのために、受入れ側では、たとえば、次のように利用を許可できるオブジェクトを文字列と関連させて、(受入れ側サイトの利用者が)明示的に登録する必要がある。

```
regist_object "object0" site0obj ;;
```

この例では、オブジェクトsite0objを文字列"object0"として登録している。

次に、オブジェクトのグループ階層機構について述べる。本言語のオブジェクトは、埋込み型(embedding based)であり、インスタンス変数およびメソッドをコードとして実際にもっているので、ネットワークオブジェクトに適している。グループ階層化されたオブジェクトは、木構造として表現される。図2のように Object

Table に格納されたオブジェクトのデータ構造が、木構造に関係付けられる。

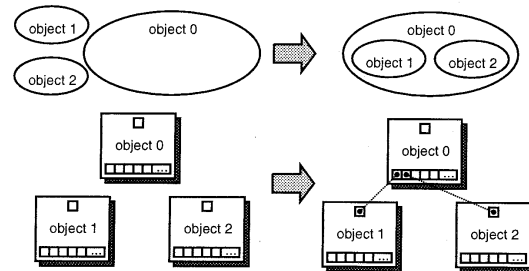


図2: オブジェクトのグループ階層化とデータ構造

外部オブジェクトのデータ構造は親になり、内部オブジェクトのデータ構造は子に表現される。前述のように、親のデータ構造が移動または複製される場合には、この関係に基づいて移動(複製)される。

また、移動オブジェクトが自分自身を内包しているオブジェクトのメソッドを利用する際は、対象となるデータ構造に格納されたメソッドを呼び出す。

4 おわりに

本言語は、高階関数型の特徴とオブジェクト指向の特徴を併せてもつ。さらに、移動オブジェクトのグループ階層化機能によって、オブジェクトを細かく部品として定義し、その集合体を1つのオブジェクトのように扱えるようになる。また、ソフトウェアシステムの動的な再構成やネットワークを通じての新機能の獲得といった機能を提供することが可能である。したがって、本言語は、より柔軟性に富んだ分散ソフトウェアシステムを簡潔に構築できると考える。

参考文献

- [1] L.Cardelli and A.D.Gordon, **Mobile Ambients**. *Proc. European Joint Conference on Theory and Practice of Software*, pp.140-155. 1998.
- [2] Cedric Fournet, Georges Gonthier, Jean-Jacques Lévy, Luc Maranget and Didier Rémy, **A calculus of mobile agents**. *CONCUR '96:Concurrency Theory*, pp.406-421, 1996.
- [3] Luca Cardelli, **A Language with Distributed Scope**. *22nd ACM Symposium on Principles of Programming Languages*, pp.286-297, 1995.
- [4] Henry Cejtin, Suresh Jagannathan, and Richard Kelsey, **Higher-Order Distributed Objects**. *ACM Transactions on Programming Languages and Systems, Vol.17, No.5*, pp.704-739. 1995.
- [5] Xavier Leroy, **The Objective Caml System Release 1.07**. *Documentation and User's Manual, Institut National de Recherche en Informatique et Automatique*, 1997.