2 P − 3

アクタモデルのπ計算に基づく意味づけ
−エージェントの形式化に向けて−

河辺 義信　真野 健　小暮 潔
NTTコミュニケーション科学研究所

1 はじめに

エージェント技術は，複雑な分散コンピューティングを強力に支援できる技術として期待を集めている．本稿では，エージェント処理のための言語処理系構築を目指す．言語処理系構築のベースとしてアクタモデルに焦点を当て，Gasparらによる形式モデル[1](アクタ代数)を利用する．アクタ代数は詳細には調べられていないが，プロセス代数π-計算への解釈を導入することで，π-計算の知見をアクタ代数に対して適用できる．そこで本稿では，アクタ代数に基づく言語からπ-計算に基づく分散言語 Nepi^2[2] への翻訳機構を設計する．これによって，Nepi^2 処理系上でのアクタ言語処理系の実装が可能となる．実際に，我々はそのような方法で実装を行った．

2 エージェント記述のための理論体系

エージェントモデルは分散オブジェクトモデルに制限を加えたものとみなせる[3]．制限の中でも，

- エージェントは固有の識別子を持ち，他のエージェントとメッセージのやりとりを行う．
- 自己の行動や内部状態を制御する仕組みを持つ．

などの特性が重要視されている．分散オブジェクトのモデルであるアクタモデルは，これらの特性を満足する．そこで，アクタモデルをエージェント記述体系構築のベースとして用いる．

Gasparらは分散オブジェクトの条件として，

(1) 各オブジェクトが固有の識別子を持つこと(識別子を使って通信をするため)
(2) 非明示的なメッセージ受信が行われること(メッセージ受信によってアクタを起動するため)
(3) メッセージのやりとりに非同期通信が行われること(メッセージバッファリングの利用のため)
(4) 内部状態(記憶領域)を持つこと

の4点を指摘し，アクタモデルのためのプロセス代数(アクタ代数)を与えた(図1)．アクタ代数では，各アクタ($^a[P]_s$)は固有の識別子(a)と内部状態(s)を持ち，並行動作を行う．また，非明示的メッセージ受信によって入力待ち状態(^aP_s)にあるアクタが起動状態($^a[P]_s$)に変化し，処理が開始される．さらに，各アクタ間のメッセージ通信では非同期通信が用いられる．これらの特徴は上記の条件((1)〜(4))に対応している．

A	::=	0	(終了動作)
		$\mid\ ^aP_s$	(入力待ち状態)
		$\mid\ ^a[P]_s$	(起動状態)
		$\mid\ < a, v >$	(送信)
		$\mid\ A \mid A$	(並行動作)
		$\mid\ A \backslash a$	(名前制限)
P	::=	$\surd$	(終了動作)
		$\mid\ send(e_1, e_2).P$	(送信準備)
		$\mid\ create(b, P, e).P$	(アクタ生成)
		$\mid\ become(P, e).P$	(内部状態変更)
		$\mid\ e_1 : P_1 + \cdots + e_n : P_n$	(並列分岐)
		$\mid\ D(x)$	(動作定義)

図1 アクタ項

3 Nepi^2 : π-計算に基づく分散言語処理系

Nepi^2 はプロセス代数π-計算に基づいた分散言語処理系で，プロセスの並行合成やチャネルを介したデータ通信，チャネル生成によるプロセスの動的変化を言語レベルでサポートする(図2)．Nepi^2 を用いる主な利点として，以下を挙げる．

- π-計算の理論に対する拡張として，実際のプログラミングの場面を考慮した複合的なデータや通信チャネルの扱いができる．
- Pict などの他のπ-計算ベース言語に対する利点として，ガード演算子(出力，入力)を内部にとることのできる非決定和演算子のサポートがある．

これらの特徴によって，動的に変化する分散環境を簡潔に記述できる．

Pi-calculus Based Operational Semantics of An Actor-based Language.
Yoshinobu Kawabe, Ken Mano and Kiyoshi Kogure
NTT Communication Science Laboratories
2-4, Hikaridai, Seikacho, Soraku-gun, Kyoto 619-0237, Japan

P	::=	δ	(終了プロセス)
	|	$(\|\| P_1\ P_2)$	(並行合成)
	|	$(\nu\,\xi\,P)$	(チャネル生成)
	|	G	(ガード)
	|	$(+\,G_1 \cdots G_n)$	(非決定和)
	|	$(if\ n\ P_1\ P_2)$	(条件分岐)
	|	$(A\ v_1 \cdots v_n)$	(プロセス呼出し)
G	::=	$(!\,c\,v\,P)$	(出力)
	|	$(?\,c\,(x)\,P)$	(入力)

図2 Nepi^2 の構文 (一部)

4 アクタモデル言語の Nepi^2 による解釈

アクタ代数の操作の Nepi^2 による記述を行うことで，π-計算の理論で得られている知見をアクタ代数に導入し，諸性質の検証に役立てられる．その上，Nepi^2 処理系を用いたアクタ言語の実装が可能となる．そこで，アクタ代数に対して，Nepi^2 の言語への解釈を与える．

変換を行う際，解釈が第2節で述べた分散オブジェクトの持つべき条件 ((1) ～ (4)) を満たす必要がある．以下による解釈を行えば，各条件は満たされる．

(1) 各オブジェクト識別子に一対一に対応するようなチャネルを生成して用いる．これによって，Nepi^2 上での識別子の解釈を行う．

(2) アクタの非明示的なメッセージ受信動作の Nepi^2 における解釈は，アクタの動作系列の解釈の先頭にメッセージ受信のための Nepi^2 の入力演算を付加することによって行う．

(3) アクタ間の非同期通信の解釈は，以下のようにして行う．アクタの動作系列 $send(e_1, e_2).P$ が与えられているとする．このとき，$send(e_1, e_2)$ の解釈と P の解釈を Nepi^2 の並行合成演算子を用いて結合したものを解釈とする．

(4) Nepi^2 では，通信チャネルに複合的なデータを扱うことのできるような拡張が行われている．これを用いて，各アクタの持つ内部状態 (記憶領域) の解釈を行う．

また，アクタの内部状態変更操作，アクタ生成操作および並列分岐操作に関しては，以下のようにして解釈を与える．

- アクタの内部状態変更 $(become(P, e).P')$ の解釈は，P(内部状態 e) の解釈と P' の解釈の並行合成とする．アクタ生成 $(create(b, P, e).P')$ の解釈も，これと同様に行う (P は識別子 b と内部状態 e を持つ)．ただし，アクタ生成時にチャネル生成演算子を用いて新たな識別子を生成する必要がある．
- 並列分岐の解釈には条件分岐演算子を用いず，非決定和演算子とガード演算子 $(!, ?)$ を組み合わせて行う．なぜなら，並列分岐 $(e_1 : P_1 + \cdots + e_n : P_n)$ では各条件部 $(e_i (1 \leq i \leq n))$ が並行に評価され，真となるもののどれか一つだけが選択されるためである．

以上に基づいて，アクタ言語のプログラムからそれに対応する Nepi^2 プログラムへの変換を設計した．さらに，Common Lisp 上でこの変換を実装した．

5 おわりに

本稿では，分散オブジェクトモデルのアクタモデルに焦点を当て，Gaspari らの代数モデル (アクタ代数) をベースとした言語処理系を実装した．その手段として，アクタ代数から π-計算に基づく分散言語処理系 Nepi^2 の言語への解釈を定義した．Nepi^2 の言語に変換されたアクタプログラムは Nepi^2 処理系によって実行できる．

アクタモデルをエージェント記述モデルにまで拡張するために，さらに解決すべき問題がある．ひとつは外部環境や実時間の扱いの導入である．さらに，知識処理の組み込みもモデル拡張の鍵となる．知識処理機構は KQML などのエージェント通信言語を用いた通信を導入する際に必要である [4]．送信および受信エージェントの知識や心的状態を通信条件として利用するからである．知識処理の導入のため，様相論理の処理系などを Nepi^2 と組み合わせる必要がある．今後はアクタモデルに外部環境・実時間・知識処理の扱いを導入し，エージェント記述体系として拡張を行う予定である．

参考文献

[1] M.Gaspari and G.Zavattaro, An Actor Algebra for Specifying Distributed Systems: the Hurried Philosophers Case Study, In Advances in Petri Nets on Object-Orientation, to appear of LNCS (or `http://www.cs.unibo.it/~zavattar/`).

[2] E.Horita and K.Mano, Nepi^2: A Two-Level Calculus for Network Programming Based on the π-Calculus, LNCS 1345, pp. 377-378,1997.

[3] 金 淵培, エージェント技術の現状と実用化. 人工知能学会誌, Vol.12, No.6, pp 42-52, 1997.

[4] Y.Labrou, Semantics for An Agent Communication Language, PhD Thesis, University of Maryland, 1997.