Ambient Calculus を用いた移動エージェントの形式化*

2 P − 2

峯下 聡志† 渡部 卓雄†

北陸先端科学技術大学院大学

1 はじめに

分散型計算機環境における研究が色々となされる中、分散型ソフトウェアの開発に用いられる分散プログラミング言語に関する研究、特に型や意味論の研究が重要となっている。これらは処理系の設計に大きく関わってくる部分であり、分散プログラミング言語の意味の厳密な記述を行うことにより、安定した分散ソフトウェアへとつながる。

本研究では移動エージェント言語に注目し、切断時操作、フォールトトレランスなどの形式的モデル・仕様を与えることを目標とする。

2 移動エージェント

移動エージェントには、移動性、自律性、コミュニケーション、並列性、セキュリティなど、求められるものが多々あるが、それらを組み込んだ上での切断時操作、フォールトトレランスをどのように実現するか、また適当な calculus を構築できるかについて述べる。

2.1 移動エージェント言語

移動エージェント言語がさまざまな視野から開発されており、それぞれ異なる特徴を持つ。それらの移動エージェント言語を切断時操作、フォールトトレランスなどの側面から見る。

Kafka(富士通研究所 : Beta 1.3 97/06/24) では、クラス変更に伴うフォールト・トレランスは保証されるが、クラスの意味的な相違に基づく矛盾への対応はエージェントでは関知していない [2]。

また、Aglets (IBM 東京基礎研究所 : ASDK V1.0.2 98/07/05) では、常駐監視作業、非同期作業などにより、断続的な接続によるネットワークの利用が可能になっている [3]。

これらの様に、移動エージェント言語は実用に耐えるアプリケーションを提供するための研究が行なわれている。しかし、装置故障やトラブルが発生した場合の回復機能を備えたものはまだ難しく、これからの研究課題となっている。

2.2 形式的モデル : Ambient Calculus

ここで、プロセス・デバイスが ambient を持ったまま移動を行なう calculus を紹介している Mobile Ambients[1] について説明を行なう。

Mobile Ambients では、calculus に ambient の概念と特殊なアクション (ambient に対しての侵入、退出、解放) を用い、ファイアウォールへのアクセスや、自然数、チューリングマシンなどをモデル化できる表現力を持つ。

さらに、プロセス間の通信の概念を加えることにより、π-calculus のエンコードまでを可能にし、表現力が高いことを示している。

移動および通信の基本要素は次のように表される。

$P,Q ::=$	processes	$M ::=$	capabilities
$(\nu n)P$	restriction	x	variable
$\mathbf{0}$	inactivity	n	name
$P \mid Q$	composition	$in\ M$	can enter into M
$!P$	replication	$out\ M$	can exit out of M
$M[P]$	ambient	$open\ M$	can open M
$M.P$	capability action	ϵ	null
$(x).P$	input action	$M.M'$	pass
$\langle M \rangle$	async output action		

ambient は name によって表され、その ambient 内でプロセスが走るようになっている。

またリダクション規則、略記法は次の様なものがある。

$$n[in\ m.P \mid Q] \mid m[R] \rightarrow m[n[P \mid Q] \mid R]$$
$$m[n[out\ m.P \mid Q] \mid R] \rightarrow n[P \mid Q] \mid m[R]$$

*Formalization of Mobile Agent with Ambient Calculus

†Satoshi MINESHITA,Takuo WATANABE
School of Information Science,
Japan Advanced Institute of Science and Technology.
{mine,takuo}@jaist.ac.jp

$open\ n.P \mid n[Q] \rightarrow P \mid Q$
$!P \rightarrow P \mid !P$
$(x).P \mid \langle M \rangle \rightarrow P\{x \leftarrow M\}$
$n[] \triangleq n[0]$
$M \triangleq M.0$

これらを使って、ロックの機構を以下のように書くことができる。ロックの取得および、解放のモデルは次の通り。

acquire $n.P \triangleq open\ n.P$
release $n.P \triangleq n[] \mid P$

これらを使い、以下のように2つのエージェントを同期させることができるようになる。

acquire n.release $m.P \mid$ release n.acqire $m.Q$

3　モデル構築

ここで、実際にいくつかのモデルを提示する。なお、以下に出てくるモデルはAmbient Calculus[1]を用いている。

以下に、別作業プロセス、ambient 間ルートの確認のモデルを示す。

another work process $Q \triangleq q[(x).x \mid open\ p \mid open\ r.Q]$
verify route from b to $a \triangleq$
$p[out\ b.in\ a.out\ a.in\ b.in\ q.(out\ b.in\ a)] \mid in\ q.r[]$

これにより、ルート確認で b から a に行き、再び戻ってくるまで、ambient b で、ambient q 内の Q が作業を行なう切断時操作モデルが以下のように考えられる。

$a[P] \mid b[$verify route from b to $a \mid$
another work process $Q \mid R]$

また、切断されていることを他のエージェントなどに知らせる場合には、別目的のエージェントを作り、それに監視させるという方法もある。

例えば、ambient 間を往復しつづけるエージェントを作り、そのエージェントが動かなくなった場合、どちらかの ambient において切断されていることとなり、切断されていることを知らせることができるかもしれない。

以下に、移動のベースとなる ambient と、実際に往復するエージェントのモデルを示す。

base-set $a \triangleq a[!(x).l[in\ m.x.n[out\ m.\langle x \rangle]] \mid !open\ n]$
round-agent from a to $b \triangleq$
$m[in\ a.n[out\ m.(out\ a.in\ b.out\ b.in\ a)] \mid !open\ l]$

これによりルート確認をし続け、往復後に再び同じ状態に戻る不動点のような ambient 間往復エージェントのモデルが以下のように考えられる。

base-set $a \mid$ round-agent from a to $b \mid b[P]$

しかし、このエージェントが往復できなくなった、つまり、何か異常が起こった場合の対処のモデル化は難しく、現在の calculus に新たな要素を加える必要がありそうである。

このモデルはフォールトトレランスにも適用できるので、これから詳しく研究を進めていく予定である。

4　おわりに

本稿では、Ambient Calculus を用いた切断時操作、フォールトトレランスのモデル化を中心に試みた。現状ではまだ構想の段階であり、モデルとしてはまだ脆弱である。

今後、Ambient Calculus を用いたモデル化を行い、切断時操作やフォールトトレランスをモデル化するにあたっての長所や欠点を踏まえた上で、独自の calculus を用いたモデル化を行なっていく予定である。

参考文献

[1] L.Cardelli, A.D.Gordon. Moblile Ambients. ETAPS'98,1998

[2] T.Nishigaya. Design of Multi-Agent Programming Libraries for Java. http://blacky.fujitsu.co.jp/hypertext/free/kafka/paper/, 1997.

[3] Tokyo Research Laboratory, IBM Japan. IBM Aglets Software Development Kit White Paper. http://www.trl.ibm.co.jp/aglets/whitepaper/whitepaper-j.html, 1998.