A-005

混合型時間アンビアント計算の記述性

Timed Mobile Ambient との比較

Hybrid Timed Ambient Calculus

Comparison with Timed Mobile Ambient

藤坂 吉秀 †　　**樋口 昌宏** †
Yoshihide Fujisaka　　Masahiro Higuchi

1 はじめに

アンビアント計算 [1] は動的に変化するオブジェクトの階層構造の記述に適した形式モデルであり，物流監視システムへの適用が可能であると考えられる．しかし，アンビアント計算は実時間を記述するための仕組を備えていないため，荷物の積み込みや移動に一定の時間を要する等の時間制約を含む物流の記述には不十分である．そこで我々は，アンビアント計算のケーパビリティと，それを時間拡張したケーパビリティを混在させた混合型時間アンビアント計算 [2](以下，HTAC とする) を提案している．本発表では，同様にアンビアント計算を時間拡張した Timed Mobile Ambient[3](以下，tMA とする) と比較する．

2 混合型時間アンビアント計算 (HTAC)

2.1 アンビアント計算

アンビアントとは階層構造を持ち，互いに出入りが可能な「境界を持った場」であり，アンビアント計算 はその振る舞いを記述するために導入されたプロセス代数である．アンビアント計算の構文規則は文献 [1] にて以下のように定義されている。

定義 2.1 (構文規則)

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

アンビアントの振る舞いは以下の遷移規則により定められる．

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

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

$open\ n.P \mid n[Q] \to P \mid Q$

それぞれ，$n[\,]$ が $in\ m$ を消費して $m[\,]$ の中に入る遷移，$n[\,]$ が $out\ m$ を消費して $m[\,]$ から出る遷移，$open\ n$ を消費して $n[\,]$ を消滅させる遷移を表している．

2.2 時間拡張

以下では $t \in \{1, 2, \ldots\}$, $t' \in \{1, 2, \ldots\} \cup \{\infty\}$ とする．時間拡張のために，in, out, $open$ に加え有効期限付き ($in(t')$, $out(t')$, $open(t')$) および待機 ($wait(t)$) ケーパビリティを導入する．有効期限，及び待機時間の長さを表現するために正の整数値を用い，無限の有効期限を表すため ∞ を用いる．HTAC では，定義 2.1 の構文規則に加えて以下の構文規則を持つ．

定義 2.2 (構文規則)

$t ::=$	time	$M, N ::=$	capabilities
$1, 2, \ldots$	positive integer	$in(t')M$	enter into M within t'
$t' ::=$	time with ∞	$out(t')M$	exit out of M within t'
t	time	$open(t')M$	open M within t'
∞	infinity	$wait(t)$	wait t

通常のケーパビリティにおける遷移関係を「$\to$」で表す．また，有効期限付きケーパビリティに関する遷移関係を「$\xrightarrow{P}$」，時間経過による遷移関係を「$\xrightarrow{T}$」で表し，以下のように定める．in についての遷移のみ示すが，out, $open$ についても同様の遷移を持つ．

$n[in(t')\ m.P \mid Q] \mid m[R] \xrightarrow{P} m[n[P \mid Q] \mid R]$

$in\ m.P \xrightarrow{T} in\ m.P$

$in(t)\ m.P \xrightarrow{T} in(t-1)\ m.P \quad (t \geq 2)$

$in(1)\ m.P \xrightarrow{T} 0$

$in(\infty)\ m.P \xrightarrow{T} in(\infty)\ m.P$

$wait(t).P \xrightarrow{T} wait(t-1).P \quad (t \geq 2)$

$wait(1).P \xrightarrow{T} P \quad (t \geq 2)$

$P \xrightarrow{T} P'$ かつ $Q \xrightarrow{T} Q'$ ならば $P \mid Q \xrightarrow{T} P' \mid Q'$

$P \xrightarrow{T} P'$ ならば $n[P] \xrightarrow{T} n[P']$

4 番目の遷移規則の「0」は inactivity プロセスであり，有効期限の切れたケーパビリティ とそれに引き続く部分プロセス式 P が無効化されることを表している．以上の定義により，例えば $n[wait(5).in(3)\ m]$ により，アンビアント n が 5〜8 単位時間の間に m に入ることが可能であることを表現できる．また，特別な役割をもつアンビアントとして $alarm$ アンビアントを導入し，これが活性化することはシステムが異常な状態に到達したとみなすことにする．

2.3 ケーパビリティの使い分け

HTAC では物流記述を想定しており，物流記述ではトラックやコンテナなどのモノや，港や倉庫などの場所をアンビアントで表現し，それらのアンビアントの移動に関する同期をとるための制御情報のやりとりにもアンビアントを用いる．移動するモノや場所を表すアンビアントを物理アンビアント，制御情報のやりとりに用いるアンビアントを制御用アンビアントと呼ぶ．実際の物流システムでは，モノの移動には時間を要する．そこでモノの移動，つまり物理アンビアントの移動には有効期限付きケーパビリティを用いる．一方，瞬時的に実行されるべき制御情報のやりとり，つまり制御用アンビアントの移動には通常のケーパビリティを用いる．

3 tMA 固有の構文と HTAC での模倣

tMA は通信プロトコルのタイムアウト処理の記述のためにアンビアント計算を時間拡張したものである．tMA 固有の構文を列挙し，HTAC でそれらの動作の模倣について述べる．

† 近畿大学 大学院総合理工学研究科, Kindai University

3.1 セーフティプロセス

tMA ではケーパビリティに有効期限をつけることができ，$C^{\Delta t}.(P,Q)$ と記述する．ケーパビリティ C を有効期限 Δt 時間以内に消費すればプロセス P が活性化するが，期限が切れるとセーフティプロセス Q が活性化する．これによりタイムアウト処理を記述できる．

HTAC ではそのための構文は用意していないが，以下のような形でセーフティプロセスが表現できる．

$(\nu\ x)(x[\ ] \mid C(\Delta t).open\ x.P \mid wait(\Delta t).open\ x.Q)$

またマクロ記法の導入により，以下のように記述可能である [4]．

$\{C(\Delta t).P\ timeout\ Q\}$

3.2 アンビアントの有効期限

tMA では，アンビアントにも有効期限をつけることができ，$N^{\Delta t}[P]$ と記述する．Δt 時間経過による期限切れによりアンビアントとその内部プロセスが消滅する．

HTAC では，特別な役割を持つアンビアントとして $trashbox$ アンビアントを導入し，期限の切れたアンビアントを $trashbox$ に格納し，他のプロセスとの相互作用を不能にさせることでこれを模倣する．

$N[P \mid wait(\Delta t).(trashbox[out\ N] \mid in\ trashbox)]$

これをマクロ化すると以下のように記述可能である．

$N[P \mid wait(\Delta t).trash(N)]$

3.3 アンビアントのタグ

tMA ではアンビアントに active と passive の 2 値のタグをつけ，アンビアントの状態を示す．active はアンビアントが in, out , go といった移動ができ，passive はアンビアントがそれらの移動ができないと定めている．状態が active であるとき，移動により状態は passive になる．状態が passive であるときは時間経過により active になる．これにより，個々のアンビアントの動作が離散的に発生することを保証している．

HTAC では，以下のように $wait$ を用いることで，強制的に時間経過させることが可能である．

$N[in(\Delta t)M.wait(1).P]$

3.4 Location と go

tMA では，プロセスは Location の内部でのみ実行される．Location 間の移動には go のみを用い，目的地に到達するまでは他のプロセスが動作することはない．go は時間パラメータが 1 になると動作可能となる．

HTAC では，階層構造の一番上のアンビアントを Location と見立て，in，out，$wait$ を組み合わせる．これにより同様の機能を模倣したものが以下である．

$LocA[N[out\ LocA.wait(t-1).$
$(in(1)LocB.P \mid wait(1).alarm[\ ])]] \mid LocB[\ ]$

これをマクロ化すると以下のように記述可能である．

$LocA[N[go(\Delta t)LocB]] \mid LocB[]$

4 HTAC の記述性

3 節では tMA の構文を HTAC で模倣することで tMA と同等の表現力があることを示した．ここでは，HTAC と tMA の記述性について比較する．

4.1 物流システムの記述

3.3 節で tMA では，アンビアントの連続した移動は離散的に発生しなければならないことを述べた．物流システムでは，制御情報のやりとりとして，1 つの制御用アンビアントが連続的に移動することも必要である．例えば，同期では二つのアンビアント A, B が時間経過なしでお互いの存在を確認した後に，続く動作 P, Q として振る舞うというシステムを考える．これは HTAC 式で以下のように記述できる．

$N[A[in\ B.out\ B.P] \mid B[in\ A.out\ A.Q]]$

この記述では P と Q が同じタイミングで活性化することができるが，tMA では連続した移動に時間経過を伴ってしまうので同様の記述では上述のシステムを表現できない．

4.2 通信プロトコルの記述

文献 [3] では，tMA により，TCP/IP における 3 ウェイハンドシェイクのコネクション確立フェーズの仕様を Location 数 2，アンビアント数 5，ケーパビリティ数 11 で記述している．これを 3 節で示した模倣法により HTAC 式に直接変換すると，アンビアント数は 11 となり，ケーパビリティは 29 となった．一方，3 節で示したマクロを HTAC 式に適用すると以下のようになる．

$l1[Client[Send] \mid SendAck]] \mid l2[Server[Receive]]$	
$Send =$	$SYN[out\ Client.go(\Delta t)l2.in\ Server \mid$
	$wait(\Delta t_1).trash(SYN)]$
$SendAck =$	$\{open(\Delta t_4)SYNACK.ACK[out\ Client.$
	$go(\Delta t)l2.in\ Server \mid wait(\Delta t_1).$
	$trash(ACK)]\ timeout\ Send \mid SendAck\}$
$Receive =$	$open(\Delta t_8)SYN.$
	$(SYNACK[out\ Server.go(\Delta t)l1.in\ Client \mid$
	$wait(\Delta t_1).trash(SYNACK)] \mid Receive)$

この記述では直接表示されるアンビアント数は 7，マクロを含めたケーパビリティ数は 18 となった．これによりマクロの利用により，HTAC は tMA とほぼ同程度の簡潔さで，この仕様を記述できたと言える．

5 おわりに

本発表では，HTAC が物流システムの記述に向いており，さらに，通信プロトコル仕様の記述もできることを示した．また，HTAC はマクロ化をすすめることで，さらに簡潔な式の記述が可能になると思われる．

謝辞 本研究は科研費 (25330095) の助成を受けたものである．

参考文献

[1] Gardelli, L. and Gordon, A.D.: Mobile Ambients, The oretical Computer Science, Vol. 240, pp. 177-213 (2000).

[2] 樋口 昌宏: 時間付き Ambient Calculus, 情報処理学会, プログラミング研究会, (2013 − 01).

[3] Bogdan Aman. and Gabriel Ciobann: Timed Mobile Ambient for Network Protocols, LNCS, Vol. 5048, pp. 234-250 (2008).

[4] 濱本 明伸: 時間アンビアント計算による物流記述支援システムにおけるマクロ機能の導入, 近畿大学卒業研究報告 (2015 − 02).