時間拡張型LOTOSの実行環境

2T－2

安藤　勉　　中野　宣政　　渡辺　尚　　水野　忠則

静岡大学　工学部

1　はじめに

近年、マルチメディア通信に代表される情報通信の高度化に伴い、「ある要求された時間内での通信サービスの提供」即ち、タイムクリティカル性といった時間的な付加価値が求められている。またそれに伴って、システムを記述するための仕様記述言語に対しても時間を含めた記述を可能にするような取り組みが行われている[2, 3]。そこで本研究では、仕様記述言語の一つであるLOTOS[1]に対し、簡単な時間記述を含めた仕様記述に対する動作を可能にすることを目的としている。

2　時間拡張型LOTOS

今回、我々が定義する時間拡張型LOTOSでは、LOTOSに対して次のようなイベントアクションに対する時間制限の付与を行っている。

即ち、既存のLOTOSに対して以下のような時間記述を可能とした。

$$a\{t\};\ \mathrm{B}$$

LOTOSにおいて、a; B は動作式B にイベントa を前置した動作式である。ここで、イベントa に対して$a\{t\}$ という記述を可能にすることで、イベントa の生起に対して時間的な制限を付与することができる。$a\{t\}$ のセマンティックスは「直前のイベントの生起よりt 単位時間経過した瞬間に、イベントa は生起可能となる」である。t 単位時間経過した、まさにその瞬間のみ、$a\{t\}$ は生起可能となりうる。$a\{t\}$ が動作式の先頭であれば、それはプロセスもしくは仕様のインタンシェートされた時刻よりt 単位時間経過したその瞬間にのみ生起可能となることを意味する。また環境との相互作用の関係で、イベントa が生起しえない場合はstop してしまう。

図1に示すように、通常のLOTOSではイベントは生起可能となってから任意の時間で生起しうるのに対し、時間拡張型LOTOSではその内の時間t でのみ生起可能であるとする。

既存のLOTOSでは、提供可能となったイベントに対しその生起する時間を記述することはできなかったが、$a\{t\}$; B という記述を可能とすることで、イベントa の生起時間を直前のイベントからの相対時間として記述することが可能となる。

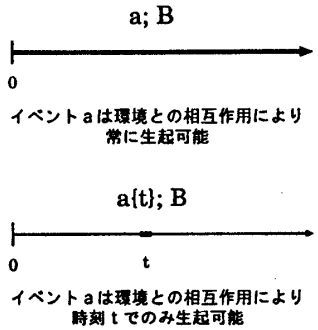

図 1: 時間拡張型LOTOSのセマンティックス

3　実現方法

今回の時間拡張型LOTOSの動作環境をつくるにあたって、(財) 情報処理相互運用技術協会 (INTAP) で開発された、LOTOSの動作環境であるLIpS (LOTOS Interpretation Server)[5] を使用する。時間仕様の付与の方法としては、時間拡張型LOTOSで追加されたセマンティックスを損なわないように、時間拡張型LOTOSで書かれた仕様を、プリプロセッサによりのLOTOSの標準仕様へと変換し、これをLIpS上で動作させる。システム全体の構成は図2のようになる。

次にどのように変換を行うかであるが、時間の概念を取り入れた以上、時間を測定するプロセスは不可欠である。時刻を刻むプロセスとしてclock を以下のように定義する。ここで、時間は単純増加する離散的な数とする。

```
process clock[tick](t:Nat):exit :=
    tick!t; clock[tick](succ(t)) [] i; exit
endproc
```

また、$a\{t\}$ はそのセマンティックスより、直前のイベントよりt だけ経った瞬間に生起するため、a が生起するまでの時間t だけ待たせるタイマープロセスとしてプロセスtimer を以下のように定義する。ただし、全てのタイマープロセスtimer は、唯一のタ

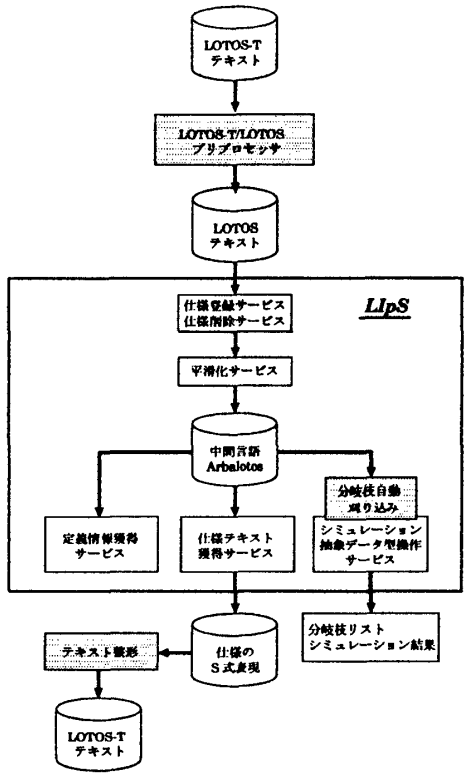

図 2: システムの全体構成

イミングプロセス clock とゲート tick で同期しているものとする。

```
process timer[tick](t:Nat):exit :=
    [t eq 0] -> exit []
    [t gt 0] -> tick?cur:Nat;
    timer[tick](t - 1)
endproc
```

例えば、$a\{t\}$; stop は以下のように置換される。

```
a{t}; stop
    ↓
( timer[tick](t) >> (a; stop) [] i; stop )
```

上式で、i; stop という動作式は、t 単位時間待った後にイベント a が生起し得なかった場合の振る舞いであるが、ここで問題となるのは内部イベント i の生起タイミングの解釈である。LOTOS においては、内部イベント i は生起可能となってから任意の時間で生起するため、「$a\{t\}$ において、t 単位時間経ったのちイベント a が生起できない場合には stop として振る舞う」という時間拡張型 LOTOS のセマンティックスを表現できない。そこで先程述べた変換を行ううえで、内部イベント i は ASAP(As Soon As Possible) で生起するという制限を設ける必要がある。

図 3に、時間拡張型 LOTOS で書かれた仕様を LOTOS の仕様へと変換した例を示す。

```
specification test[a,b](v:Nat):exit
library NaturalNumber endlib
behaviour
    a{1}; b{2}; exit
endspec
            ↓
specification test[tick,a,b](v:Nat):exit
library NaturalNumber endlib
behaviour
    clock[tick](0) |[tick]|(
    timer[tick](t + 1) >> (a; timer[tick](t + 2) >>\
(b; exit
    ) [] i; stop ) [] i; stop )
where
    process clock[tick](t:Nat):exit :=
        tick!t; clock[tick](t + 1) [] i; exit
    endproc
    process timer[tick](t:Nat):exit :=
        [t eq 0] -> exit []
        [t gt 0] -> tick?cur:Nat;
        timer[tick](t - 1)
    endproc
endspec
```

図 3: 時間拡張型 LOTOS 仕様記述の変換例

4 今後の展開

今回は LOTOS の動作環境である LIpS をそのまま利用し、時間拡張型 LOTOS の仕様を変換を行ったうえで、LOTOS 仕様として LIpS 上で動作させた。しかしながら、グローバルな時間の取得や時間軸上での ASAP ルールの実現、今後のシンタックス・セマンティックスの追加に対し、現 LOTOS 仕様への変換では限界があるため、今後は同様の処理を LIpS のシステム内部の処理として実現できるよう LIpS 本体の改造を検討する。

参考文献

[1] ISO: Information processing systems – Open System Interconnection – A formal description technique based on the temporal ordering of observational behaviour, ISO 8807, (1989.2)

[2] ISO/IEC JTC1/SC21/WG1/Q48.6: Revised Draft on Enhancement to LOTOS, 1994

[3] 中野、渡辺、水野：LOTOS の時間拡張に関する研究とその標準化動向について, 情報処理学会研究報告, 情報研報 vol.94,No.39, pp.163-168, (1994.5)

[4] 中野、太田、渡辺、水野：TCCA 仕様記述対応 LOTOS の時間拡張案とその実行環境について, 情報処理学会研究報告, 情報研報 vol.94,No.56, pp.43-48 (1994.7)

[5] 辻、田中、水野：LOTOS Server とモジュール仕様記述, 情報処理学会第 41 回全国大会, (1990.9)