

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

中野 宣政、太田 賢、渡辺 尚、水野 忠則

静岡大学工学部

TCCA、およびTCCを実現するメカニズムに関する仕様記述の観点から、ISO/IECにおけるLOTOS拡張案の時間拡張仕様LOTOS-Tを検討してきた。LOTOS-Tは、最少のシンタクス、セマンティックスの拡張を標準LOTOSに施すことにより、主にタイマーなどの記述、時間ベースの性能記述を可能とするとともに、標準LOTOS仕様との互換性を極力上方互換で保とうとしている。またこの拡張仕様により、我々の意図するTCCAおよびそのメカニズムの仕様記述に、最小限必要な機能はサポートできるであろうと判断している。

我々はTCCAおよびそのメカニズムの仕様記述とその検証を行うための開発ツールとして、LOTOS-Tのシミュレータを、(財)情報処理相互運用技術協会(INTAP)で開発されたLipS (LOTOS Interpretation Server)を利用し構築中であり、本稿では、LipSに時間軸を擬似的に追加して構築すべく、その実装について検討を行った内容、およびLOTOS-T/LOTOSトランスレータ等について述べる。

Time Enhancement of LOTOS (LOTOS-T) for the Description of Time Critical Communication Architecture and Mechanisms and it's development Environment

Nobumasa Nakano, Ken Ohta, Takashi Watanabe, Tadanori Mizuno

Shizuoka University

We have studied LOTOS-T, the Time Enhancement of LOTOS definition ;the part of the initial draft on Enhancements to LOTOS (ISO/ IEC JTC1/ SC21 N8023) as the candidate for describing Time Critical Communications Architecture and Mechanisms.

LOTOS-T is designed with minimal syntactic and semantic additions (upward compatible) to existing LOTOS to enable to describe time related functional specifications such as performance related and real time specifications. We assume, using the LOTOS -T definition, we can at least describe the specifications of TCCA and it's mechanisms to realize it.

This paper describes our study to construct LOTOS-T Development Environments (Simulator) using LipS; (LOTOS Interpretation Server) developed by INTEROPERABILITY TECHNOLOGY ASSOCIATION FOR INFORMATION PROCESSING, JAPAN (INTAP) by adding pseudo Time faculties and LOTOS-T/LOTOS Translator to it.

1. まえがき

TCCA[1]、およびそのメカニズムに関する仕様記述の観点から、ISO/IECにおけるLOTOS拡張案の時間拡張仕様LOTOS-T[2]を検討してきた[3]。LOTOS-Tは、最少の拡張を、標準LOTOS (IS8807)に施すことにより、主にタイマー等の時間記述、時間ベースの性能記述を可能とするとともに、標準LOTOS仕様との互換性を、極力その上方互換で保とうとしている。

我々は、TCCAのメカニズムを実現するシステム記述に、LOTOS-Tを使用してみることとし、その開発支援環境として、(財)情報処理相互運用協会(INTAP)で開発された、LipS (LOTOS Interpretation Server) [4]を時間拡張使用する、LOTOS-T開発環境を構築中である。LipSを対象とした理由として、特に全体の部分仕様をモジュールとして記述でき、部分的なシミュレーション実行をも可能にするなど、本格的な実用システムへの応用を考慮した特長があげられる。

以下本稿では、第2章に、LipSの開発環境の時間拡張の方針、および時間軸の実装に関するシステム・グローバル/プロセス・ローカル時間の導入の検討について記し、第3章に、LipS応用実装としての時間制御方式、すなわちイベントアクション制御方式、時間プロセスの実装を記し、第4章にLOTOS-T/LOTOSテキストへのトランスレータの内容、第5章に時間/ASAPルールの適用による、不要インターリーピング岐の刈り込み問題について述べ、第6章に、まとめと今後の研究課題について述べる。

2. LOTOS-T開発環境の構築に関する基本方針の検討

2.1 拡張基本方式の検討

LipSでは、LOTOSテキストの字句解析・構文解析を行ったうえで、ISO8807で規定する平滑化関数(flattening function) $\#$ 、 $\#$ を適用して、LOTOSテキストをCLS (Canonical LOTOS Specification)に相当する意味を持った中間言語に変換する。LOTOS-Tに拡張するためには、LOTOSのCLS、および写像関数[CLS]も時間拡張し、

$$\#, \#' : \text{LOTOS-Tテキスト} \rightarrow \text{CLS}' = \langle \text{AS}, \text{TIME}, \text{BS}' \rangle \quad (1)$$

$$[\text{CLS}'] : G^N X Q \{s_1\} x \cdots Q \{s_m\} \rightarrow \{TTS_{CLS}(B')\} \quad (2)$$

とする必要が有る。但し、 $TTS_{CLS}(B')$ はLOTOS-Tで規定する導出システム D_{CLS} によるラベル付き時間拡張遷移システム、振舞記述 B' においては、各遷移は時間制約付きで記述される。

我々は、LOTOS-T自体がISO/IECにおけるワーキングドラフトであり、流動的であることより、本格的な拡張は行わず、LipSをそのまま使用し、トランスレータにより、時間制限を含む振舞記述 B' を、時間記述を陽に含まず一般変数、常数として扱う通常の B 仕様に変換し、時間プロセスを追加することによりシミュレーションを行う方式を採用することとした。

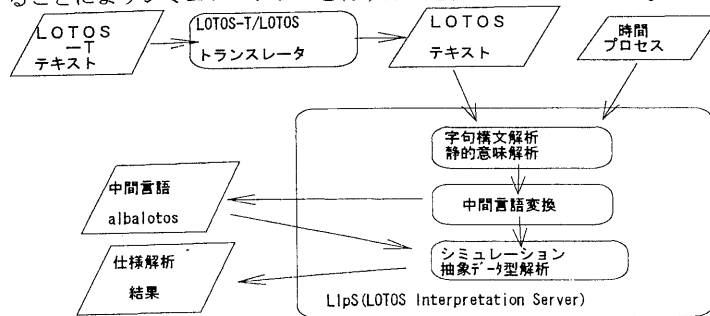


図1. LOTOS-T開発支援環境

2. 2 時間軸の実装

TCCAにおいては、複数のネットワークプロバイダー・プロセスがノード内、ノード間に並列に存在し、その実装に於いては、先ずシステム全体、およびこの並列実行プロセスの時間表現を過不足無く行われなければならない。LOTOS-Tにおいては、そのシンタックス、セマンティックスに関する拡張と共に、その時間ドメインを抽象データ型として定めている。具体的には、時間ドメインとしてはシステム全域に関する、初期値 t_0 、システム起動後常に増大しつづける絶対時間、 $Q(\text{time})$ がグローバル定義されている。同時に、同仕様書において個々のプロセス内の実際の時間記述では、各プロセスの生起、または、先行するイベントの生起からの所定の時間増分（すなわち、当該プロセス生起時間、または先行イベント生起時間を $t_0 (= 0)$ とする）でもって指定される。我々はこれに対し、理論的にも、時間ドメインのシグネチャーの各（サブ）プロセス対応のローカルな時間ドメインを再定義し、更には各ゲート対応のインクレメンタル時間をローカル時間より導出する必要があると考える。すなわち、

(1) システムグローバル時刻

システムに唯一の、システム起動時よりの経過時間を自然数、または有理数で t_{glo} として持つが、そのデータタイプを以下に規定する。

<pre> type global_time_type is nat renamedby sortnames global_time for nat opnnames + for + 0 for 0 > for gt endtype </pre>	<pre> type global_time_type is p_rational renamedby sortnames global_time_type for p_rational opnnames + for + 00 for 00 > for gt endtype </pre>
--	---

図2. グローバルタイム型

(2) 総てのプロセスはそのインスタンス化された時刻を0とし、そのプロセスの存在する間計時しつづけるローカルタイム t_{loc} を、総てのプロセス対応に有するものとしそのタイプを以下とする。

```

type local_time_type is global_time_type
    sort local_time
    opns - : global_time, global_time -> local_time
endtype

```

図3. ローカルタイム型

図3のシグニチャーをベースとする時間のセマンティックスは以下となる。

$$t_{\text{loc}}:\text{local_time} = Q(\text{time}):\text{global_time} - t_0(\text{process xyz}):\text{global_time} \quad (3)$$

「あるプロセス (x y z) のローカルタイム t_{loc} は、グローバルタイムの現在時刻 $Q(\text{time})$ から、当該プロセスの生起したグローバル時刻 t_0 を引いた値である」

(3) 各プロセスにおいて、それに続くイベントアクションは、先行するイベントを起点とするインクレメンタル時刻 t_{inc} を有し、そのデータタイプは以下とする。

```

type incremental_time_type is local_time_type
    sort incremental_time
    opns - : local_time, local_time -> incremental_time
endtype

```

図4. インクレメンタルタイム型

図4のシグニチャーをベースとする時間のセマンティクスは以下となる。

$$t_{inc}:incremental_time(gate\ i) = t_{loc}:local_time - t_i(gate\ i):local_time \quad (4)$$

「あるプロセスのあるゲート (i) に後続する任意のイベント対応のインクリメンタルタイムは、そのプロセスのローカルタイム t_{loc} から、先行ゲートイベントの生起時間 (ローカルタイムで現される) t_i を引いた値である」

グローバル/ローカル/インクリメンタルタイムの相関を図5に示す。

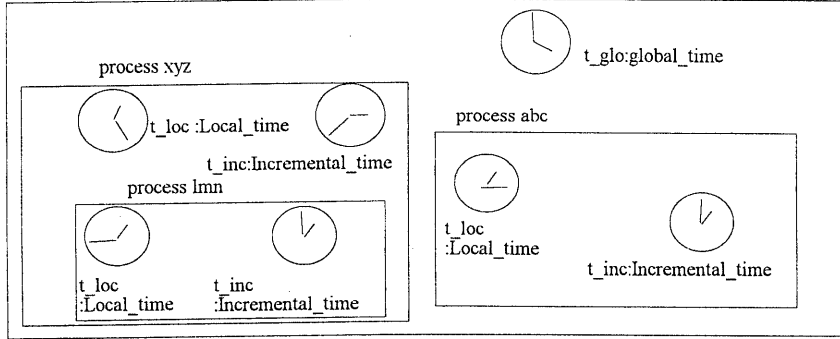


図5. グローバル/ローカル・タイムドメインの相関

3. 時間制御実現方式

3.1 イベントアクション時間制御方式

一般的に、ローカル時間ドメインの上での時間制限付きイベントアクションの実現手法としては、カノニカルフォームに展開して行うやり方と、実行時にダイナミックに制御する2種類がある。

(1) カノニカル展開によるスタティック決定方式

イベントアクションの生起時間を平滑化することにより、グローバルタイム (または任意のプロセスのローカルタイム) として表わし、その時刻となったら当該イベントアクションを活性化させる。

本来的にはグローバルタイム一本で平滑化を行うべきであるが、特にシミュレーションにより部分的に検証を行う場合などは、任意のプロセスのローカルタイムを起点とすることがおこる。

(2) 実行時ダイナミック制御方式

システム外部よりの実行時イベントを付与され動作するシステムに於いては、実行前にイベントアクション木が決まらない場合があるし、またイベントアクション木が決定したとしても、その木内の時間値が実行時に決定されるものも考えられる。従ってこの場合は実行時に値を評価するダイナミック制御方式となる。

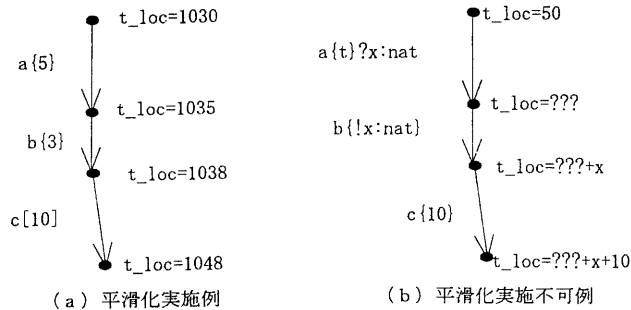


図6. イベントアクション・タイムの平滑化についての2例

3. 2 グローバルタイムプロセス

このプロセスは、基底プロセスが起動してすぐ起動し、シミュレータとしてのグローバルタイムとして内部イベント i に同期して作動する。シミュレータとしては i の発生を操作することにより、時刻を起動停止できるものとする。また、シミュレータの操作関数を使用して、任意の時刻まで進めることもできるものとする。

```
process global_time[g](t_glo:nat):noexit:=
    i:g!t_glo:global_time[g](succ(t_glo))
endproc
```

図7. グローバルタイムプロセス

4. トランスレータ

以下、LOTOS-Tテキストを、本支援環境対応のLOTOSテキストへ変換するトランスレータについて述べる。

LOTOS-TはLOTOSに対し、二つだけのシンタックス、セマンティクスの追加を行っている。すなわち、イベントアクション、および $exit$ アクションへの時間制限付与である。基本的には、この二つの構文対応のISO/IEC標準LOTOSテキストへの変換を行えばよい。

(1) ローカルタイムの設定

各アプリケーションプロセスにおけるローカルタイムを確立するため、各プロセスの起動直後に以下の行を挿入する（但しBをプロセス動作式とする）。これは前述(3)式を具体的に記載したものであり、プロセスが起動されたとき t_0 の値をそのときのグローバル時刻にインスタンスエートし、グローバルタイムとの差分をそのプロセスのローカルタイムとする。

```
let t_0:global_time=t_glo:global_time,
    t_loc:local_time=t_glo:global_time-t_0:local_time in B (5)
```

(2) 時間制限付きイベントアクション $a\{t\};B$ の書き換え

以下と書き換える。これは前述(4)式の t_{inc} を導出し、これと時間仕様 t (インクリメンタル時間インスタンス) を比べ、一致したところで a を付勢している。

```
let t_i:local_time=t_loc:local_time,
    t_inc:incremental_time=t_loc-t_i [t_inc>=t]->a;B (6)
```

(3) 時間制限付き $exit$ 、 $exit\{t\}>>B$ の書き換え

以下と書き換える。

```
let t_i:local_time=t_loc:local_time,
    t_inc:incremental_time=t_loc-t_i;[t_inc>=t]->exit>>B (7)
```

5. 時間 (ASAP) 制限 (may タイミング) による分岐木の刈り込み

LIP Sは、トランスレータより出力された時間記述を、単に変数またはそのインスタンスとしてのLOTOS標準テキストとして扱い、それをカノニカルフォームに展開し仕様獲得を行う。このとき、LIP Sには時間の概念が無いから、インターリーブを行う際に不必要な実行木を形成する可能性が有る。本格的には、2.1項で述べた写像関数[CLS]を時間拡張版[CLS']へ変更する必要が有り、具体的にはLOTOS-Tの規定する導出システム D_{CLS} をベースとするラベル付き時間拡張遷移システム TTS_C を構築する必要が有るが、ここではシミュレータの操作上で、オペレータの判断により不必要枝の削除を行うものとする。

例えば、図8に示す如く、時間概念を容れた平行プロセスのインターリーピングは、時間概念無しの場合の6分岐にたいし、可能なのはその内の一つのみである。

a{3};b{5};stop ||| c{2};d{10};stop

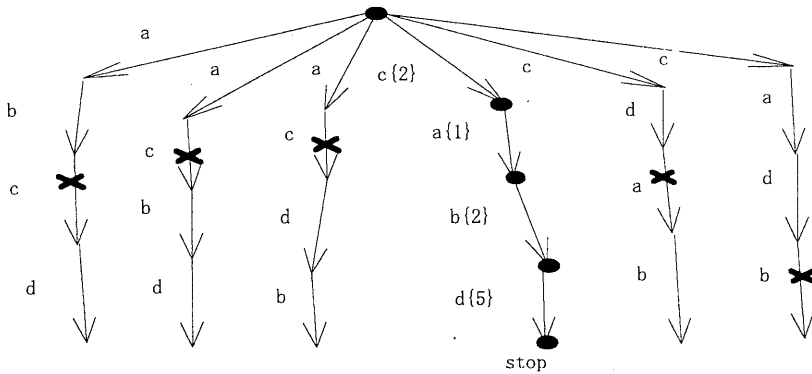


図8. インターリーブ分岐の刈り込み

6. むすび (今後の課題)

LIP Sを応用したLOTOS-T開発環境の構築について述べた。今後、本検討にもとづくLOTOS-Tの開発環境を完成させると共に、我々の研究の目的であるTCCAとそのメカニズムの研究を、本開発環境を利用して行っていく。問題のカテゴリーとしては、タイムクリティカル通信プロバイダーとしてのアーキテクチャーとサポートメカニズム、および関連する資源共用問題であり、またそれに伴うQoSの検討である。

具体的な問題としては、TCCAとそのプロバイダー関連サービス/プロトコル、ユーザ・プロバイダー間QoS/プロバイダー内QoS・性能マッピングの検討、クライアント・サーバにおける応答時間管理問題、ネットワーク帯域モニター/管理問題、共用メモリー管理問題、時空コヒーレンシー維持メカニズム、等があげられる。

謝辞 LOTOS拡張に関して、標準化活動に尽力されている、情報規格調査会FDT-SWG小委員会の諸氏に感謝いたします。また、LIP Sシミュレータおよび貴重な関連資料をご提供頂き、適切なご助言を頂いた、電子技術総合研究所 大蒔和仁氏、および三菱電機(株)情報システム研究所 辻宏郷氏、LOTOS全般に関しご助言頂いた、三菱電機(株)情報システム研究所、佐藤文明氏に深く感謝いたします。

参考文献

- [1] 中野宣政, 渡辺尚, 水野忠則: FA/CIMにおける分散制御監視システム構築の技術要件とその具現化についての考察, マルチメディア通信と分散処理ワークショップ 論文集, 情報処理学会, pp169-178, oct. 1993.
- [2] ISO/IEC JTC1/SC21 N 8023: Initial Draft on Enhancements to LOTOS, pp157-179
- [3] 中野宣政, 渡辺尚, 水野忠則: LOTOSの時間拡張に関する研究とその標準化動向について, 第64回システムソフトウェアとオペレーティングシステム, 第65回マルチメディア通信と分散処理合同研究会予稿集, 情報処理学会, pp163-168, April 1994.
- [4] Kazuhito Ohmaki, Hirosato Tsuji, Kenjiroh Yamanaka, Yoshikazu Sato, Yoshinori Itabashi, and Toshihiko Shimizu: Design and Implementation of an Application Interface for LOTOS Processors, Fourth International Conference on FORMAL DESCRIPTION TECHNIQUES (FORTE'91) Proceedings, pp351-366, November, 1991.