4O－7

動作実行時刻に制約のある分散システムの全体仕様から各ノードの動作記述の自動導出

山口 弘純　岡野 浩三　東野 輝夫　谷口 健一

大阪大学 大学院基礎工学研究科 情報数理系専攻

E-mail : { h-yamagu, okano, higashino, taniguchi }@ics.es.osaka-u.ac.jp

1. まえがき

近年のリアルタイム分散システムの急速な普及により，動作の生起時刻に制約がある分散システムの全体仕様から，通信に要する時間を考慮して，その制約を満たしながら全体仕様で指定された通りに動作する各ノードの動作記述を自動導出する研究が注目を集めている [1, 2]. 文献 [1], [2] の手法では全体仕様を記述するモデルとしてそれぞれ時間オートマトンの組及び時間制約付き LOTOS (LOTOS/T+) を用いている. しかし，これらのモデルでは，外部入力によるシステムの内部リソース値の更新を記述できず，さらに並列動作の同期を取り扱っていない.

我々は動作の実行時刻間の制約が指定できるマーリンの時間ペトリネット (Time Petri Net) を，リソース値の更新が記述できるように拡張したレジスタ付き時間ペトリネットモデル (Time Petri Net model with Registers, 以下 TPNR モデル) を記述モデルとして用いる. 以下では，(a) TPNR モデルで記述された分散システムの全体仕様 $Sspec$, (b) 分散システムの各ノードへのリソースの配置指定, (c) ノード間の通信路の最大遅延時間, が与えられたとする. ここで，ある $Sspec'$ が $Sspec$ の各動作の時間制約を短くした仕様であることを $Sspec' \subseteq Sspec$ で表すとする. このとき，(1) ある実行方針のもとで，分散システム上でのノード間の遅延を考慮しても，その時間制約を満たしながら同様の動作が実行可能であるようなある $Sspec'$ ($\subseteq Sspec$) が存在するか否かを判定し，(2) 存在する場合には，(1) の判定で得られる解を用いて，同実行方針のもとで $Sspec'$ の時間制約を満たしながら同様の動作を実行する，(TPNR モデルで記述された) 各ノードの動作記述を自動導出する手法の概略を述べる. なお，分散システムは各ノードが正確に時刻を更新するクロックを保持するとする.

実行方針は各ノードがどのタイミングでどの動作を行うかの方針を与える. 与えられた実行方針のもとで (1) の条件を満たす $Sspec'$ は一意ではないが，なるべくその動作の実行可能時刻の幅が大きい $Sspec'$ を見つける工夫をしている.

本手法により，設計者自身が通信動作に要する遅延時間を詳細に検討せずに，動作の実行可能時間幅の広い動作記述を生成できる，などの効果が期待できる.

2. 導出例とアルゴリズムの概略

TPNR モデルでは，入出力が実行される有限個のゲート及び大域変数を表す有限個のレジスタに対し，ゲートへの入出力イベント (I/O event) 及びレジスタ値を表す変数 (レジスタ変数) の値更新 (change of register values) をトランジションに記述できる. また，トークンによる発火制御，入力変数及びレジスタ変数による条件文 (guard) での発火制御，及び上記の 2 条件が揃ってから発火するまでの時間による発火制御が可能である.

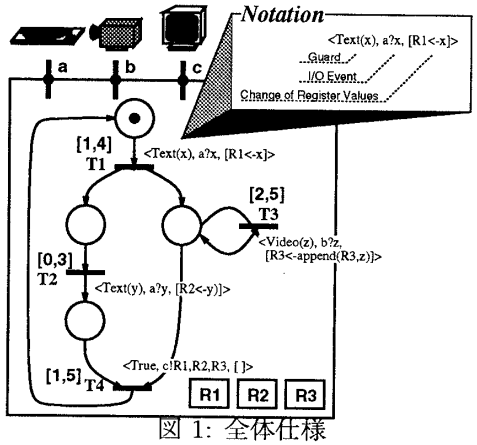

図 1: 全体仕様

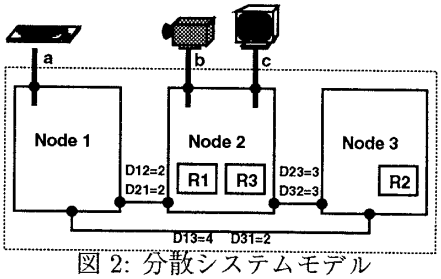

図 2: 分散システムモデル

• **全体仕様**　図 1は TPNR モデルによる全体仕様記述の例である. 例えばトランジション $T1$ はこのマーキングにおいて発火に必要なトークンが揃っているので，ガード Text(x) が真であればその時刻から 1 ～ 4 単位時間経過の間に必ず発火し，発火すると実際にゲート a から入力 x が読み込まれ，その値がレジスタ $R1$ に代入される. $T3$ は $T1$ の発火後，2 単位時間から 5 単位時間の間で周期的に発火を繰り返す. $T4$ は $T2$ の 1 回の発火と $T3$ の 0 回以上の発火を経た後発火し，ゲート c に $R1$, $R2$, $R3$ の値を出力して初期マーキングに戻る.

• **分散システムモデル**　各ノードはそれぞれ正確に時刻を更新するクロックを持ち，いくつかのレジスタ及びゲート (これらをリソースと呼ぶ) を保持する. 各ノード間には十分信頼できる全二重通信路が存在し，ノード i から j へのメッセージ送信の最大遅延時間は定数 D(i, j) で抑えられる (図 2).

• **各ノードの動作記述**　図 1 で与えられた全体仕様記述と，図 2 で与えられる分散システムモデル (ノード数，リソース配置，各通信路の最大遅延時間は自由に指定で

An Algorithm for Deriving Protocol Specifications from Service Specifications with Time Constraints
Hirozumi YAMAGUCHI, Kozo OKANO,
Teruo HIGASHINO and Kenichi TANIGUCHI
Department of Information and Computer Sciences,
Faculty of Engineering Science, Osaka University
Toyonaka-shi, Osaka 560 Japan

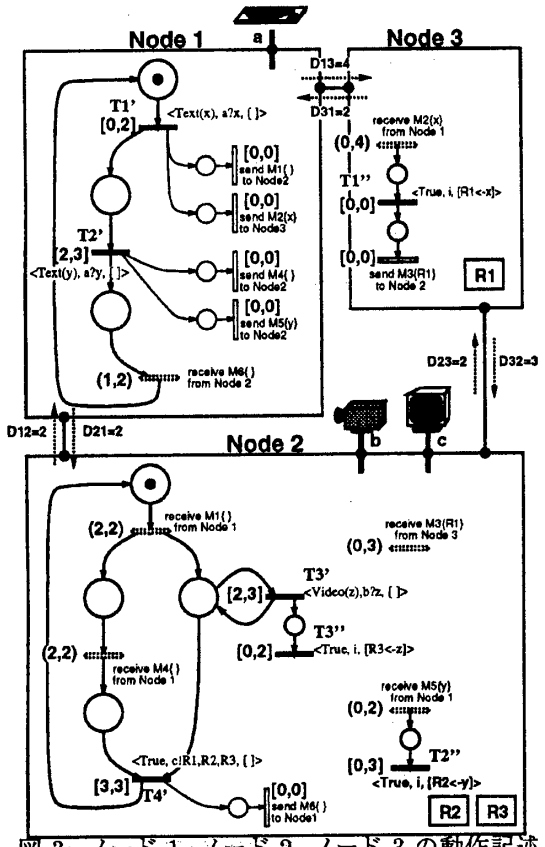

図 3: ノード 1, ノード 2, ノード 3 の動作記述

きる) に対し, 我々の導出アルゴリズムを用いて導出した各ノードの動作記述を図 3に示す*.

動作記述は例えば以下のように動作する. (I) 全体仕様のトランジション $T2$ に対し, その入力イベント, レジスタ値更新はゲート a を保持するノード 1, レジスタ $R2$ を保持するノード 2 がそれぞれが実行する ($T2'$, $T2''$). ノード 1 は入力 y の値を渡すためのメッセージ $M5$ をノード 2 に送信する. (II) $T2$, $T4$ について, $T4$ の入力イベントを実行するノード 2 に対し, ノード 1 は $T2$ の入力イベントの実行 ($T2'$) の終了を知らせるメッセージ $M4$ を送信する. (III) ノード 2 は $T4$ の出力イベントの実行 ($T4'$) に必要な $R1$ の値を知らない. この値はそれを知っているノード 3 が $T1''$ で更新した値をノード 2 に知らせるためのメッセージ $M3$ を送信する.

ここで, 上述の (III) より, メッセージ $M3$ は $T4'$ が実行される前に到着する必要がある. $T1$ のイベントが実行された時刻 ($T1'$ が発火した時刻) を t とすると, 時刻 t からメッセージ $M3$ がノード 2 に到着するまでの時間は ($M2$, $M3$ の遅延時間により) 最大 7 単位時間が必要であるため, t から $T4'$ が実行されるまでの時間を最小でも 7 単位時間以上となるよう, $T2'$ の実行時間, $M4$ の遅延時間, 及び $T4'$ の実行時間の和の最小を調整し (ここでは $2+2+3=7$), $M3$ を受け取った後に $T4'$ が実行されるようにする (この結果, 動作記述では $T2'$ は $T1'$ の実行後 [2,3] (全体仕様では [0,3]), $T4'$ は $T2'$ の実行後 [5,5] (同 [1,5]) の時刻範囲でそれぞれ実行することとなり, 動作の実行可能時間幅が全体仕様より一般には短くなるが, それぞれ全体仕様での対応する動作の実行可能時刻の範囲に収まっていることがわかる).

• **導出アルゴリズムの概略**　導出アルゴリズムは (I), (II), (III) のような実行方針に基づき, 全体仕様で用いられるゲート (ここでは a, b, c) での入出力動作 (すなわちシステム外部とのやりとり) のみに着目した場合†, 分散システム上でのノード間の遅延を考慮しても, その時間制約を満たしながら同様の動作が実行可能であるようなある $Sspec'$ ($\subseteq Sspec$) と同様の動作を行う動作記述を導出する (紙面の都合上, 詳細は [3] 参照).

アルゴリズムのステップ (1) では, そのような $Sspec'$ が存在するか否かを判定する. そのためには, 上述の例のように, メッセージ $M2$, $M3$ による遅延を各動作の実行可能時間幅を短くすることで吸収可能であるための条件を, 各動作の実行時刻を表す変数を導入し, それらの変数上の線形不等式として表す. 例えば先の例では,

$$\mathrm{Emin}(T2) + \mathrm{Dmin}(M4) + \mathrm{Emin}(T4) \geq \mathrm{Dmax}(M2) + \mathrm{Rmax}(T1, R1) + \mathrm{Dmax}(M3)\ ^{\ddagger}$$

などとなる (左辺は時刻 t から $T4$ のイベントが実行される時刻までの最小時間, 右辺は時刻 t からレジスタ R1 の値がノード 2 に到着する時刻までの最大時間). このような線形不等式は与えられた全体仕様と分散システムモデルから機械的に得ることができる. 得られた不等式をすべて満たす解が存在するか否かを整数線形計画問題の解法を用いて調べ, 解が存在すれば, そのような $Sspec'$ が存在すると判定する. このとき, 適当な目的関数を与えることにより, その動作の実行可能時刻の幅が $Sspec$ に近い $Sspec'$ を見つけることができる [3].

ステップ (2) では, TPNR モデルで記述された各ノードの動作記述を, 与えられた全体仕様を一定のルールに基づき変形し, ステップ (1) で得られた解をトランジションの時間制約に用いることで得ることができる.

3. あとがき

本稿では, TPNR モデルで記述された分散システムの全体仕様と分散システムのパラメータが与えられたときに各ノードの動作記述を導出する手法の概略を述べた. 今後の課題は, 実用的な例題に提案した手法を適用することで本手法の有効性を確認することなどである.

参考文献

[1] Khoumsi, A., Bochmann, G.v. and Dssouli, R.: "On specifying services and synthesizing protocols for real-time applications," *PSTV-XIV*, pp.185-200 (1994).

[2] Nakata, A., Higashino, T. and Kenichi, T.: "Protocol Synthesis from Timed and Structured Specifications," *ICNP'95*, pp. 74-81 (1995).

[3] 山口弘純, 岡野浩三, 東野輝夫, 谷口健一: "レジスタ付き時間ペトリネットで記述された分散システムの時間制約付き全体仕様からその時間制約を満たす各ノードの動作記述の自動導出," 信学技報 Vol. 95, No. 609 (SS95-51), pp. 55-60 (1996).

*本稿では送受信を行うトランジションを例えば "Send M2{x} to Node 3" のように表し, 入力変数 x の値を含むメッセージ $M2$ をノード 3 に送信することを意味する. また, メッセージを受信するトランジションの発火時間制約を (0,3) のように表し, メッセージが送信されてから 0 ～ 3 単位時間経過までに受信することを意味する. いずれも TPNR モデルで表現できる (後者はクロックによる時刻印を必要とする) が, 見やすさのためそのような表記を用いる.

†レジスタ値更新は (外部からは見えない) 内部動作であるため, ゲートでの入出力値さえ正しいことが保証できる範囲でその実行時間をずらすことができる.

‡Emin($T2$) は $T1$ のイベント実行発火から $T2$ のイベント実行までの最小時間, Emin($T4$) は $M4$ の受信から $T4$ のイベント実行までの最大時間, Dmin($M4$) は $M4$ による遅延の最小時間, Rmax($T1$, $R1$) は $M2$ を受信してから $T1$ のレジスタ更新実行までの最大時間をそれぞれ表す変数.