

Coloured Petri Nets に基づいた通信サービス要求仕様化支援

4H-6

菊地奈穂美 上田佳寛 長谷川晴朗

沖電気工業株式会社

1. はじめに

ペトリネット[1]はグラフィカル表現で、データフローなどと同様にシステムの構造を記述できるだけでなく、動的な動作も表現することができる。また、ペトリネット理論に基づいて理論的な解析・検証を可能としている。通信システムにおいては、仕様のモデル化・分析にペトリネット理論の応用が検討・報告されている。[2]

実際のアプリケーションにおいては、仕様のモデル化時、的確なデータ表現方法が必要とされる。(ピュアな)ペトリネットは、多くの実用的なシステムに対して十分なモデル化能力をもつが、データをモデル上で表現するための洗練された手段が提供されていない。そのため、データの追加により修飾されたネットでは、それまで使われていた解析が不可能になることが多い。

Coloured Petri Nets(CP-Nets)[3]は、データのモデル化を可能にするべく、開発されてきたハイレベルネットであり、かつネットの解析に使われてきた能力を保っている。

今後、通信システムのサービス要求は、知的なサービスの実現が必要とされる。そのため、さまざまなシステム資源を表現するための機構が必要である。さらに、作成される要求仕様は、解析可能性と実行可能性を兼ね備えていなければならない。本稿では、CP-Netsが通信システムの要求仕様であるサービス仕様の表記法として有効であることを示し、要求仕様化への適用性を論じる。

2. 要求仕様化支援

通信システムに対するサービス要求の仕様化の支援方法を述べる。

通信システムは、状態遷移マシンであり、状態とイベントによりその動作が決まる。サービスは状態とイベントからなる状態遷移の組である。

この状態遷移モデルからなる個々のサービス仕様をCP-Netsによりモデル化し、システム全体のサービスをそれらのサービスの統合体として構築する。サービスはネットの構造に基づき、論理的検証と意味的検証が可能である。

この方法により、サービスの変更・追加によって変化するシステム全体のサービスの確かさを分析し検証することができる。仕様は、CP-Netsのモデルに基づいて実行可能である。

さらに、ペトリネットと形式的仕様記述言語

(FDT)間の変換の研究が各所で行われている[4,5]ことからわかるように、CP-Netsに基づいたサービス仕様は、フォーマルな設計仕様への展開が可能と見られる。よって、ソフトウェア開発の全工程から考えても、この方法は意義があるといえる。

3. CP-Netsの特性

CP-Netsの詳しい定義は[3]にゆずる。ここでは、CP-Netsのもつ特性をピュアなペトリネット、例えばPlace/Transition Nets(PT-Nets)[1]との違いに重点をおいて述べる。

(1) トークンはデータアイテム、アークとトランジションはシンボリックな式として記述する。プレース・トランジションはトークンの定義域・値域をもち、アークには生起可能トークンがカラーで表される。これにより仕様は明示的に表される。

(2) カラードトークンにより仕様を抽象化できる。プレース、トランジションからなるピュアなPT-Netsに基づくモデルと比べ、ネットの要素数の爆発を押さえ、仕様が抽象化される。

(3) CP-Netsでは、データも含めて仕様はフォーマルに定義される。これは、ピュアなペトリネット等で可能な解析検証方法がCP-Netsの構造に基づいて行えることを意味する。

以上のように、サービスの要求仕様を明示的にかつフォーマルに定義できる。

4. CP-Netsによるサービスのモデル化

カラーの使用により、複数のリソースをフォーマルに定義することができる。サービスモデルを構成するトランジションはリソースの動作、トークンはリソースを表す。プレースは、リソースの状態を表すプレース、リソースの関係を表すプレースから構成される。

通信システムの性質として、リソースは初期状態(アイドル状態)から遷移し初期状態に戻ることから、ネットの初期マーキングでは、各リソースを表すトークンはアイドル状態を表すプレースに存在する。

図1に構内電話交換のサービス表現の例を示す。

5. 検証

CP-Netsの構造に基づき、論理的な検証と意味的な検証が可能である。

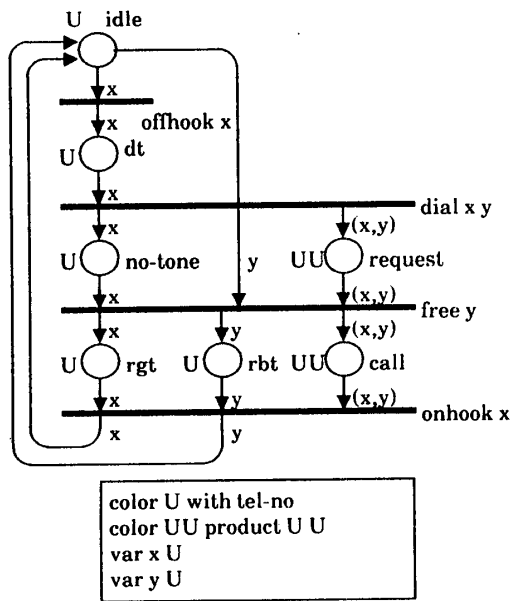


図1 CP-Netsによるサービス表記例

5.1 論理検証

サービスはトランジションの発火系列に対応する。その中のすべてのトランジションは活性(live)でなければならない。つまり、ネットが有界であると仮定した時、T-完全でなければならない。そこで、ネットの構造に基づき、有界性、T-完全性を調べる。さらに、システムの全てのノードはリソースを表すトークンが初期状態から初期状態に到達するようなトランジションの発火系列によって被覆されなければならない。デッドロックの解析はネットの構造に基づいて行う。

またPT-Netsに対する可達木と同様、CP-Netsに対してもCP-tree(CP-Netsに対する可達木)が得られる。[3] CP-treeのノードはあらゆる可達マーキングの集合であり、これを調べることによって望ましくないシステムの状態の有無を知ることができる。一般に可達木は有限でないが、被覆マーキングと同値なマーキングの使用によってリダクションが可能なので、有限なCP-treeが得られる。

5.2 意味検証

● 意味検証用知識の表現

意味検証のためには通信システムに関する知識が必要である。知識には、個々のリソースの状態の変化の規則と複数のリソース間の従属した関係がある。知識表現は、サービスを構成するCP-Netsの構造の部分定義として定義する。

個々のリソースの知識は、プレース、トランジション、アークの組み合わせ(カラーを含む)として表す。さらに、リソース間の関係の知識は、プレースの組み表せで表す。

従来のシステムの多くは、ルールの集合としてシステム独自の形で知識を表すものが多い。ここでは、意味検証用知識をCP-Netsの構造の部分定義

として定義することにより、意味検証をもネットの構造に基づき行うことを可能にする。また、この方式により定義した知識自体の論理的な確かさを調べる事が可能になる。

次に意味検証用知識の定義例を示す。

● P-invariantによる知識の表現

P-invariantはある重みづけを行ったトークン数の総和が一定のプレースの集合を表す。従って、関係するリソースの状態の組み合わせ、リソースの背反する状態をP-invariantにより定義できる。これらの条件をP-invariantとして定義すると、サービスの追加・変更によりネットが変更したとき、P-invariantを確認することにより、元の条件が成立するかどうかを確認できる。原則として、新サービスの定義やサービスの変更によってP-invariantは不変でなければならない。従ってサービスの変更に伴うネットの変化後、新たなP-invariantを求めることにより、その状態が正しいかどうかを判断することができる。次に知識表現例を示す。

例: 図1の呼び出し中の電話(rgt)は、呼び出し音を聞いている電話(rbt)とつながっている。2つの電話の関係はcallプレースにより表現される。次式はこの関係を表すP-invariantの例である。

$$M(rgt) = P1 \cdot M(call)$$

$$M(rbt) = P2 \cdot M(call)$$

注) P1, P2はincidence-function[3]と呼ばれる。

$$\text{ここで、 } P1(x,y) = x, P2(x,y) = y$$

6. まとめ

通信システムのサービス仕様をCP-Netsに基づいて明示的にかつフォーマルにモデル化し、検証する方式を示した。CP-Netsは、ビュアなベトリネットや修飾されたネットと比べるとデータのモデル化能力に富み、かつ検証可能で仕様の実行も可能であることから、要求仕様の表現として適していると考えられる。

この方式に基づく支援システムは現在インプリメント中である。今後、ユーザ/設計者が入力する要求サービスの記述形式/入力形式の考慮が必要である。

参考文献

[1] W. Reisig: "Petri Nets, an Introduction," Springer-Verlag, 1982.
 [2] G. Comparin, et al.: "Analysis of SDL Specifications Using Petri Nets Techniques," 2nd SDL-Users' Forum, PTL, 1986.
 [3] K. Jensen: "Coloured Petri Nets," Lecture Notes in Computer Science, Vol.254, Springer-Verlag, 1986.
 [4] V. K. Garg: "Modeling of Distributed Systems by Concurrent Regular Expressions," FORTE'89, 1989
 [5] E. Kettunen, M. Lindqvist: "Toward Practicality of Predicate/Transition Net Reachability Analysis of SDL," 3rd SDL Forum, North-Holland, 1987.