TM-0407

通信システム用設計仕様の論理検証方式

柴田健次，上田佳寛，湯山さつき，田中　亘
長谷川晴朗

November, 1987

ICOT
Mita Kokusai Bldg. 21F
4-28 Mita 1-Chome
Minato-ku Tokyo 108 Japan
(03) 456-3191～5
Telex ICOT J32964

Institute for New Generation Computer Technology

通信システム用設計仕様の論理検証方式

A Logical Method of Verifying Design Specification in a Communication System

柴田 健次　上田 佳寛　湯山 さつき　田中 亘　長谷川 晴朗

Kenji SHIBATA　Yoshihiro UEDA　Satsuki YUYAMA　Wataru TANAKA　Haruo HASEGAWA

沖電気工業株式会社

Oki Electric Industry Co., Ltd.

Abstract　We are now developing a software support system -EXPRESS (EXPeRt system for ESS). EXPRESS designs automatically the specification for a communication system from users' requirements. Each requirement is converted into PSG(Partial Service Graph) and all PSGs are integrated into TSG(Total Service Graph). TSG is the final design specification. PSG and TSG are described by using Petri Nets. The specification is verified syntactically by utilizing an algebraic method of Petri Nets. This paper gives an outline of EXPRESS and describes the feature of Petri Nets in EXPRESS. Then it shows some examples of analysis of the design specification.

1　はじめに

近年、通信システムにおいてユーザの要求は多様化、高度化してきている。そのため、通信システム用のソフトウェア開発の負担が大きくなってきている。特に、設計工程の上流工程である、要求仕様化段階において、高級技術者にかかる工数が多大である。要求仕様化段階は、後の工程の基本となるため、最も重要な段階である。ユーザの要求から正確な仕様が作成されなければならない。

我々は、通信システム開発における仕様化段階のサポートシステムであるEXPRESS(EXPeRt system for ESS)を開発中である。EXPRESSは、自然言語で記述されたユーザの要求から、全体として整合のとれた設計仕様を自動作成し、それを検証するシステムである。EXPRESSにおいて、仕様はペトリネットで記述されている。そして、ペトリネットの代数的解析法を用いることにより、仕様の検証を行っている。

本稿では、はじめに、EXPRESSの概要および検証の概要について述べる。次に、EXPRESSで使用されるペトリネットについてその性質等を明らかにし、設計仕様の論理検証について述べる。

2　EXPRESS概要

2.1　システム構成

図1にEXPRESSのシステム構成概念図を示す。本システムは、要求理解サブシステム、仕様統合サブシステム、知識ベースおよび仕様検証サブシステムより構成される。

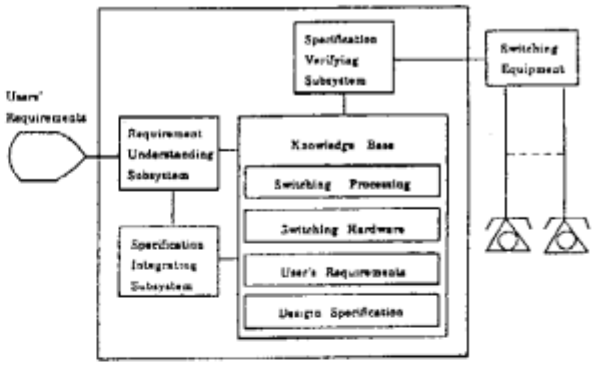


図1.　システム構成概念図

(1)要求理解サブシステム

サービス要求定義者が、自然言語により交換サービスに関する曖昧な要求を入力すると、知識ベースの中の知識を用いて要求理解を行う。また、知識ベースと矛盾するサービスが入力された時は、ユーザとインタラクションをとることにより矛盾を解消する。

(2)仕様統合サブシステム

本サブシステムにおいて、個々の仕様を一つの最終的な設計仕様にする。入力される要求は、一般に個々のサービス単位で断片的である。従って相互に矛

盾する可能性を有しているが、本サブシステムはこれから誤りのない設計仕様を作成する。

(3)知識ベース

設計仕様および設計仕様を作成するために必要となる知識が蓄積される。この知識は、交換処理、交換ハードウェア、ユーザ要求等に関するものである。

(4)仕様検証サブシステム

知識ベースに蓄積された設計仕様に従って、交換ハードウェア装置を制御する。ユーザの要求が正確に理解されているか否かの意味検証を行う。

2.2　仕様表現

図2にEXPRESSにおいて、矛盾･曖昧性のない設計仕様を作成する過程を示す。

専門家がユーザの要求から設計仕様を決定する過程は、ユーザの要求を表すサービス表現から設計仕様を表すサービス表現への変換として捉えることができる。以下にそれぞれのサービス表現について述べる。

(1)SE(サービス要素)

最初にユーザの要求はSEに変換される。SEは、サービスの開始から終了までの、端末の動作とその動作によって変化する端末の状態とを記述したものである。図3にSEの表現例を示す。

```
se(knt(内線相互接続, 発呼者先掛),時間(1)),
  ([tr( オフフック,[agent(発呼者)])],
   [st(聞く,[goal(発呼者),object(ダイヤルトーン)])])).

                  :
```

図3.　SEの表現例

(3)PSG(部分サービスグラフ)

PSGはSEを時間の経過に従ってペトリネットに変換したものである。PSGにおいて動作をトランジション、前状態を入力プレースの集合、後状態を出力プレースの集合として表す。入力プレース、出力プレースは関与する個々の端末の状態を表す。また、個々の端末は、トークンとして扱う。

図4にPSGの表現例を示す。また、図6の(a)、(b)、(c)にそれぞれ内線相互接続サービスの発呼者先掛、呼び出し中切断、ホットライン呼び出し中切断のペトリネット表現を示す。

```
psg(knt(kn( 内線相互接続, 発呼者先掛),時間(1)),
 spp([place([rel(token(発呼者,ext),idle,nil)]),
       place([rel(token(ダイヤルトーン,dt),idle,nil)])]),
 tr(token( 発呼者,ext),offhook,nil),
 snp([place([rel(token(発呼者,ext),receive,
                   token(ダイヤルトーン,dt))])])) .
```

図4.　PSGの表現例

(3)TSG(統合サービスグラフ)

TSGは複数個の断片的なPSGを統合し、一般的なサービスグラフにしたものであり、最終の設計仕様である。単一の状態一つに一つのプレースを、動作にトランジションを対応させている。そして、TSGはこれらのプレースとトランジションの集合によって表され、お互いの接続関係が明確にされる。図5にTSGの内部表現を示す。また、図6の(d)に(a)、(b)、(c)、を統合したTSGのペトリネット表現を示す。

```
place(Pt1,gpl([rel(token(Ttk1,ext),idle)]),
          spt([Tt2,Tt4,Tt7]),
          snt([Tt1,Tt5]))
transition(Tt1,Arc1,offhook,transit([Arc1,Arc2]),
           spp([Pt1,Pt6]),
           snp([Pt2]))
arc(Arc1,in(Ttk1),out(Ttk3),atr([]))
```

図5.　TSGの表現例

上記のようにPSG、TSGはペトリネット表現に対応させている。通信システムの仕様記述用言語としては、CCITTによって勧告されたSDLが良く知られているが、EXPRESSではペトリネットを採用している。ペトリネットは動的なシステムをモデル化し解析をすることを目的としたもので、実働性の検証、解

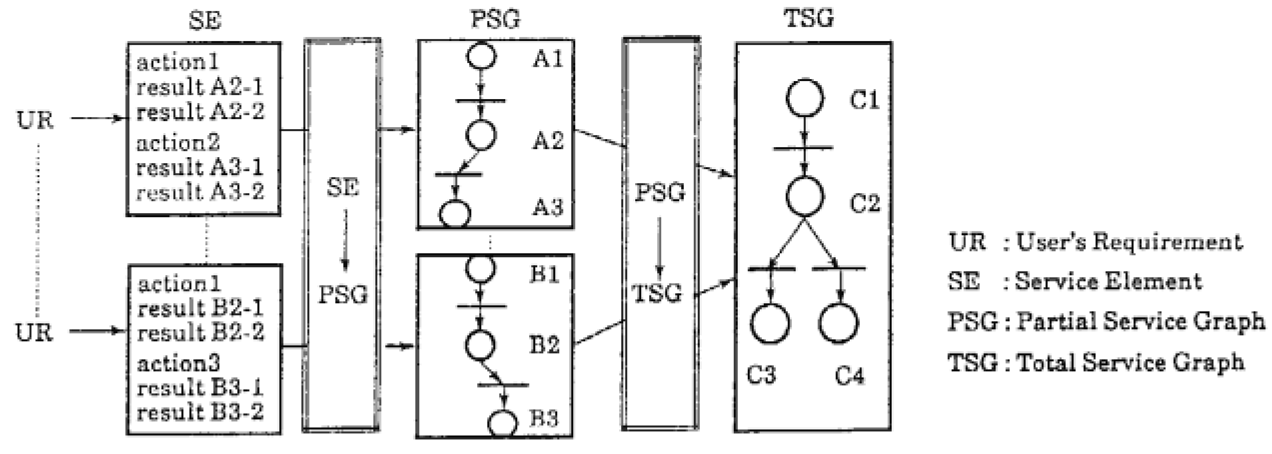
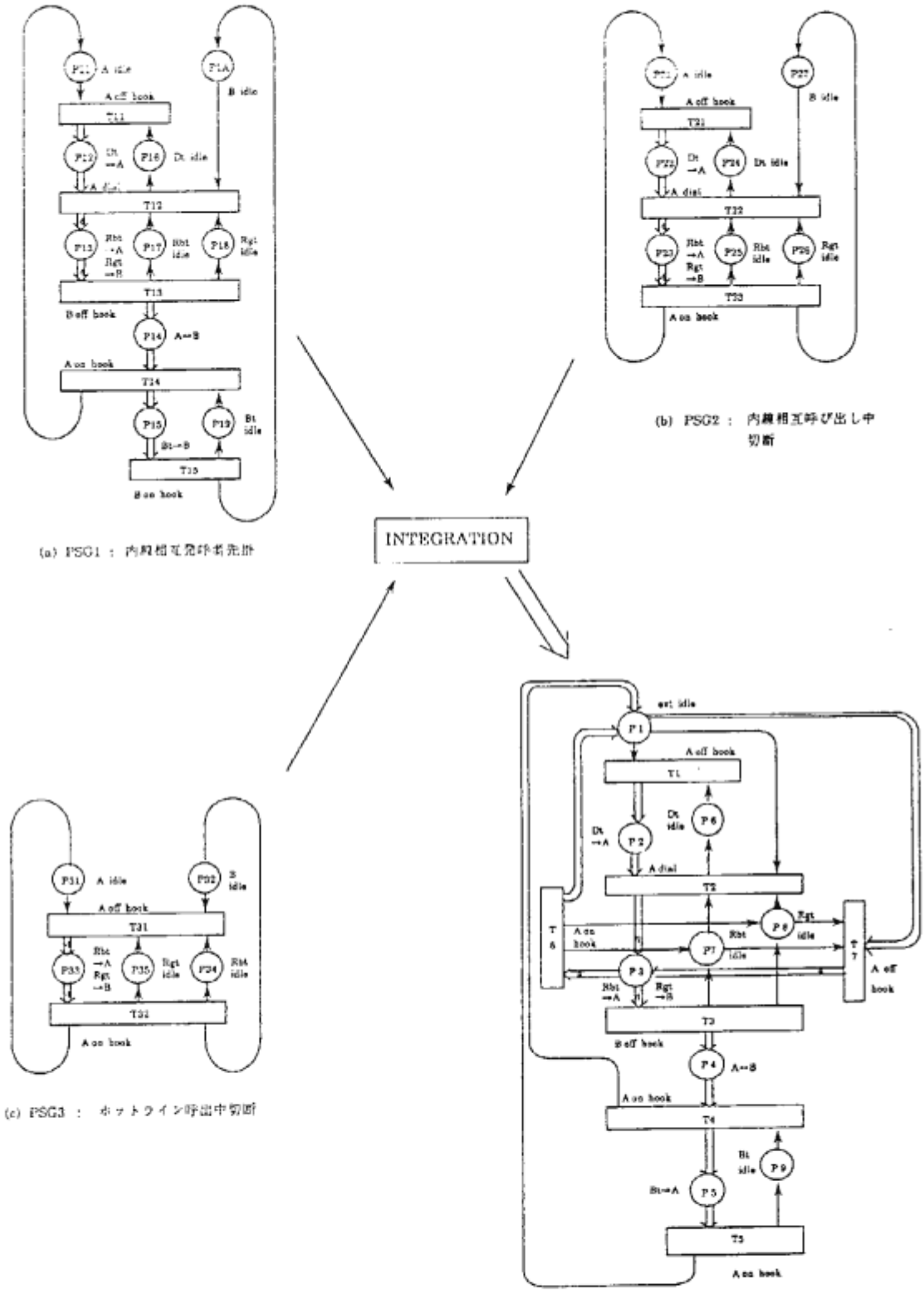


図2.　設計仕様の作成過程



(a) PSG1 : 内線相互及び外線発信

(b) PSG2 : 内線相互呼び出し中切断

(c) PSG3 : ホットライン呼出中切断

(d) TSG

図6. PSG及びTSGの表現例

析が行いやすいという特徴を有する。またEXPRESSでは得られた仕様そのものを知識として利用する。この時、仕様をペトリネットで表現しておくことにより、追加、変更に柔軟に対応できるので知識表現に有効であると考えられる。

3 EXPRESSにおける検証

交換システムにおいては、要求の新規作成、変更、追加ということが頻繁に起りうる。このとき、新しい要求そのものに誤りが含まれていたり、追加することによって誤り、矛盾が生じることが多々あると考えられるが、専門家はサービス追加時に発生するエラーを設計仕様にまで持ちこまないようにしている。

本システムは、ユーザの要求を正確に仕様化することを目的としている。このためには、入力に矛盾が含まれていたり、作成する過程で矛盾が作りこまれるようなことがあってはならない。よって、設計仕様の作成段階で矛盾を検出し除去しておくことが必要である。また、設計仕様が要求を満足するかどうかの確認も重要である。

本システムでは、次に示す検証を行う。
(1)仕様が論理的に正しいかどうかを調べる論理検証
(2)仕様が意味的に正しいかどうかを調べる意味検証

3.1 論理検証

論理的に正しいとは、ペトリネットで表現されている各PSG、TSGに対してそのペトリネットが論理的に正しいことを言う。論理的な矛盾は個々のPSGに含まれるものと、PSGをTSGに統合する際に作りこまれるものがある。PSG自体の矛盾の一例としては次のようなものがある。全てのあるいはいくつかのトランジションを経由した後、初期状態に戻らない場合である。また、TSGに統合する際、入力プレースとトランジションは一致するが、出力プレースが異なる場合がある。以上のような論理的な矛盾は、行列を用いた解析的手法により検出することができる。

3.2 意味検証

意味的に正しいとは、次の性質を持つものを言う。
(1)一意性 ： 曖昧さがなく意味が一通りに定まる
(2)無矛盾性 ： 出来上がった仕様に内部矛盾がない
(3)要求充足性 ： ユーザの要求を完全に満たしている

ユーザは通信システムにおける正確な知識を持っているわけではない。従って、曖昧な言葉を使用したり、ハード的に不可能な動作を要求したりする。これらの曖昧さや矛盾を取り除くために、設計仕様の作成段階で交換に関する知識を用いる。また、本システムでは、要求充足性の検証を行うために、実際に交換ハードウェアを動作させる。これによりユーザが実際に端末を操作することで要求の確認ができる。このようなプロトタイピング技法により、通信システムに関する知識のないユーザも容易に検証をすることができると考えられる。

4 ペトリネットの関数行列

本章で、ペトリネットの関数行列について紹介する。トランジションの入力関数及び出力関数を表す行列D^-、D^+を定義する。各行列は、トランジションに対応するm個の行とプレースに対応するn個の列から成る。D^-のj行i列の要素はプレースP_iからトランジションT_jに入るアークの多重度を、また、D^+のj行i列の要素は、T_jからP_iに出ていくアークの多重度を示す。各プレースに存在するトークンの数をn次元ベクトルμ(マーキングと呼ぶ)で表すと、(1)式が成立するとき、T_jが発火する。

$$\mu \geqq e[j]\cdot D^- \qquad (1)$$

但し、e[j]は第j成分が1で、それ以外の成分が0のm次元ベクトルである。

また、マーキングμにおいて、T_jが発火すると、新しいマーキングは次のようになる。

$$\mu = \mu + e[j]\cdot D \qquad (2)$$
$$\text{但し、} D = D^+ - D^-$$

従って、マーキングμにおいて発火系列

$$\sigma = T_{j1}, T_{j2}, T_{j3}, \cdots T_{jn}$$

が発火すると、新しいマーキングは次のようになる。

$$\mu = \mu + (e[j1]+e[j2]+\cdots+e[jn])\cdot D$$
$$= \mu + f(\sigma)\cdot D$$

5 設計仕様の性質

EXPRESSにおいては、設計仕様はペトリネットに対応づけて表現している。すなわち、リソースをトークンに、動作をトランジションに、状態をプレースに対応させている。また、設計仕様は、ユーザの要求した複数のサービスの集合とみなすことができる。ここでは、このサービスのペトリネットにおける定義を行い、ペトリネットで表現する設計仕様がどのような性質を持っているかについて明らかにする。

5.1 サービスの定義

$f(\sigma)$が次の項目を充たすとき、発火可能ベクトルと定義する。

(a) $f(\sigma)$はT-invariantである。つまり、$f(\sigma)$のすべての成分は非負整数であり、$f(\sigma)\cdot D = 0$ を充たす。

(b) $f(\sigma)$の示すトランジションの発火系列において、少なくとも一つは(1)式を充たす発火系列を持つ。($f(\sigma)$が擬似解(トランジションの可能な発火系列に対応しない解)ではないことの保証)。

このとき、$f(\sigma)$は、マーキングμがμ自身から可逆となるときの解になる。μにおいて、idle状態を示すプレースのみにトークンがあれば、$f(\sigma)$が一つのサービスとなる。つまり、空き状態から空き状態にいたるループが、サービスである。また、発火可能ベクトルで表されるサービスは、他のどのようなサービスの和でも表せない独立したサービスの和で構成される場合が多い。このような独立したサービスをここでは素サービスと呼ぶことにする。

次の2条件は、素サービスを求めるための条件である。

(I)初期マーキングにトークンが存在しているプレース以外に含まれるトークンが、常に入力アーク数をこえてはいけない。つまり、初期マーキングでトークンが存在していないプレースは有界である。これは、プレースに存在するトークン数を制限することにより、発火可能ベクトルの一次従属なベクトルを素サービスとするものである。以下に、ペトリネットグラフのモデルで例を示す。

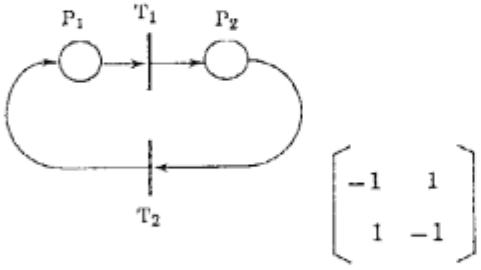


図7. 条件(I)の説明図

この行列のT-invariantは(n,n)となる。初期マーキングをμ=(2,0)とおくと、T1→T2の発火によりマーキングは元にもどり、ベクトル(1,1)は発火可能ベクトルとなる。ベクトル(2,2)を考えると、T1→T1→T2→T2と発火できるが、これは、T1→T2の発火を2度繰り返しているにすぎない。このような場合、プレースに存在するトークンの最大数を入力アーク数とすることにより素サービスを一意に決定することができる。
(II)トークン間の関係として次の2つの場合のいずれかを満足しなければならない。

(1) 複数のトークンがあるトランジションの発火により同時に遷移し、同じプレースに同時に存在する。

(2) (1)の関係のトークンの遷移により関係づけられたトークンで、(1)の関係を持たない場合。たとえば、aとb、bとcが(1)の関係を持ち、aとcが(1)の関係を持たない場合のaとcの関係を言う。

5.2 設計仕様を表すペトリネットの性質

以下にあげることが、ペトリネットで表される設計仕様に必要となることである。

s1.仕様に記述された動作が可能であり、かつ、仕様化された全ての遷移は論理的に起りうる。
s2.リソースは必ず初期状態に戻る。
s3.少なくとも一つのサービスが存在する。
s4.同一のサービスが複数は存在しない。
s5.一つのサービス内において、全く関係のないリソースは存在しない。

このような性質からペトリネットとして充たすべき性質は次のようになる。

① 動作が可能になるということは、可達性が成り立つということであり、全ての遷移が起りうるということは活性を充たすということである。

② リソースが初期状態に戻るためには、リソースを表すトークンの遷移がなくなってはならず、保存性を充たすことが必要である。また、トークンの消滅がないため、入力プレースの集合と、出力プレースの集合は同じであり、ループを形成している。

③ 少なくとも一つのサービスが存在するには、必ず1個以上のT-invariantを持たなければいけない。つまり、

$$\mathrm{rank}(D) \leqq m-1$$

である。
ここで、mは行列Dの行数である。

④ 上記のs4、s5は、設計仕様を表すサービスが素サービスとなることを意味している。

6 設計仕様の検証

本章では、前章で明らかにした設計仕様の性質を用いて、仕様の検証を行う。

6.1 サービス全体の検証

図6(d)は、発呼者先掛サービス、呼出中切断サービスおよびホットライン呼出し中切断サービスとを統合して作られるTSGのペトリネットグラフである。このTSGをあらわすペトリネットの関数行列は、次のようになる。

$$D=\begin{pmatrix} -1 & 2 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ -1 & -2 & 4 & 0 & 0 & 1 & -1 & -1 & 0 \\ 1 & 0 & 0 & -2 & 2 & 0 & 0 & 0 & -1 \\ 1 & 0 & 0 & 0 & -2 & 0 & 0 & 0 & 1 \\ 2 & 0 & -4 & 0 & 0 & 0 & 1 & 1 & 0 \\ -2 & 0 & 4 & 0 & 0 & 0 & -1 & -1 & 0 \end{pmatrix}$$

このTSGにおいて、一次独立なT-invariantの個数は、

$$m - \mathrm{rank}(D) = 3$$

mは行列Dの行数

となる。
統合する前の3つのサービスを表すベクトルは、次の3ベクトルである。

$$p = (1,1,1,1,1,0,0)$$
$$q = (1,1,0,0,0,1,0)$$
$$r = (0,0,0,0,0,1,1)$$

これら3つのベクトルは一次独立であり、素サービスを表している。
また、ベクトルp、q、rによる一次従属なベクトルの中にも、素サービスを表すベクトルが存在する可能性がある。任意の発火可能ベクトルは、

$$f(\sigma) = \alpha p + \beta q + \gamma r \quad (\alpha, \beta, \gamma \text{は整数})$$

のようになり、素サービスを表すベクトルは、p、q、rおよび

$$s = (0,0,1,1,1,0,1)$$

の4つである。このsで表される素サービスは、ホットライン発呼者先掛けサービスに対応している。

以上から、TSGは、ベクトルp、q、rおよびsで表される4つの素サービスからなっていることが導かれる。

6.2 部分的サービスの検証

サービスは初期状態を表す初期マーキングから初期マーキングまでと定義され、サービス全体の検証については上に述べた。さらに、初期状態から初期状態、でない場合について記述したサービスの一部が正しいかどうかを検証することができれば、既に確認されている動作について冗長な記述をすることなく必要とする部分だけの検証をすることが可能となる。

これについて、以下に述べる。

1. TSGからすべての素サービスを求める。

$$L^* = \{L1, L2, \cdots\}$$

2. 追加したペトリネットについて、L^*に含まれないトランジションが存在するならば、このペトリネットは異常とみなされる。これは、通信システムにおいては、リソースを表すトークンの遷移がなくなることはないからである。

3. 追加したペトリネットについて、トランジションがすべてL^*に含まれる場合。追加したペトリネットの開始状態のマーキングをμ_s、終了状態のマーキングをμ_eとする。μ_s、μ_eが、一つの素サービスの中に含まれているとすると、μ_sからμ_eまでの発火系列と、μ_eからμ_sまでの発火系列との和が、μ_sからμ_sへの発火系列となり、追加したペトリネットもその一つの素サービスに含まれていることになる。このとき、追加したサービスの一部分が正しいということができる。

一例として、発呼者先掛サービスに呼出中からidleへ戻るサービスの部分を付加する場合を考える。

図8のTSGから全ての素サービスを表すベクトルを求めると6.1と同様にしてt、uの2つのベクトルが求まる。

$$t = (1,1,1,1,1,0)$$
$$u = (1,1,0,0,0,1)$$

追加したペトリネットのμ_s、μ_eはtおよびuにより遷移するマーキングの中にともに存在し、トランジション6がuのベクトルにより発火するので、呼出中からidleへ戻るサービスの部分は呼出中切断サービスの一部として正しいとみなすことができる。

7 おわりに

本稿では、通信システムの開発における仕様化段階のサポートシステムであるEXPRESSの概要、および、ペトリネットによって表した設計仕様について述べた。EXPRESSにおけるサービスの定義を明確にするとともに、設計仕様についての性質とそれを表すペトリネットの性質をも明らかにした。そして、これをもとにして、仕様の論理検証を行った。ペトリネットの関数行列を解析することによって、仕様の正当性の検証、仕様から全てのサービスの導出が可能であることを示した。本システムにおいては、リソースの識別を行わせるために色つきトークンの導入も検討しているが、このようなシステムにも関数行列の解析が適応できるかどうか今後の検討課題である。

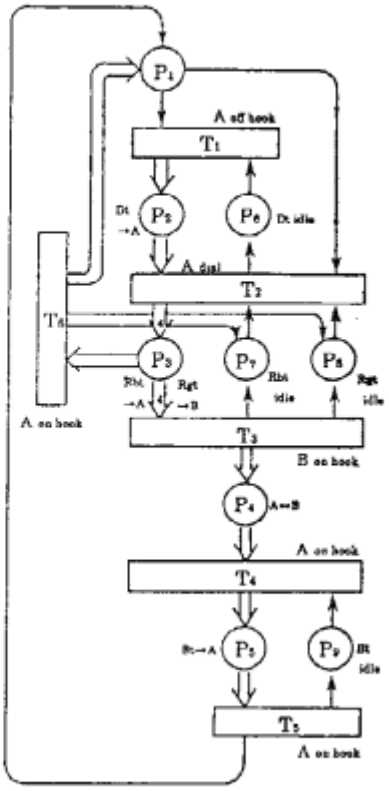


図8. TSG

なお、本研究は第5世代コンピュータプロジェクトの一環として行っているものである。

参考文献

[1] J.L.Peterson:"Petri Net Theory and Modeling of Systems" Prentice-Hall,(1981)

[2] 上田,柴田,出中,長谷川:"通信システム用仕様設計における追加仕様の検証方式",第35回情報処理学会全国大会,3W-10,(1987)

[3] H.Hasegawa,W.Tanaka,K.Shibata:"Analysis of Design Specification in a Communication System by means of Petri Nets", Proceeding of 11th Computer Software and Application Conference, pp.701~706, (1987)

[4] 翁長,若:"ペトリネットのT-invarianceの構造的解析", 電子情報通信学会論文誌,Vol.J70-A,No.2, pp.185~194,(1987)