

4.2 サービスの階層的定義

設計仕様におけるサービスには、サービス(service)、素サービス(prime service)、隠れたサービス(hidden service)がある。

①サービス(service)

サービスとは、各リソースがアイドル状態である初期マーキングから最終的に初期マーキングに戻る発火可能ベクトルで示される発火列である。

②素サービス(prime service)

ベトリネット行列における発火可能ベクトルの数は、一次従属なものまで考えると、無限である。そのためサービスの数も無限になると考えられる。このことは、多くのサービスが、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ていることを意味する。素サービスとは、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ことであり、但し、サービスは発火列、発火可能ベクトルはT-invariantで表されるために一次従属な発火可能ベクトルの中にも素サービスに対応するものがある。その例を5章に示す。4.3項に素サービスを求めるための条件を示す。

③隠れたサービス(hidden service)

曖昧なユーザ要求から設計仕様を作成するためには、抜けや矛盾に対する処理を行う必要がある。ユーザ要求からPSG(素サービスに対応する設計仕様)を作成する過程において、抜けや矛盾が生じた場合は、システム内の知識ベースを参照し、ユーザとインタラクションを取ることで、抜けや矛盾を解消している。しかし、ある一つのユーザ要求がすべて抜けている場合、システムによりその抜けたユーザ要求に対する素サービスを求めることは不可能に近い。複数のユーザ要求を統合する本システムでは、ユーザの気付きがなかった新しいルートが作成される場合がある。一例を図4に述べる。またその中には、抜けたユーザ要求に対する素サービスが存在することがある。本システムでは、その性質を利用することでサービスの抜けを解消する一手法にしている。このような、ユーザの気付きがなかったルートに対する素サービスのことを隠れたサービス(hidden service)と呼ぶ。4.4項に、その基本概念について述べる。またそれを求めた例を5章に述べる。

4.3 素サービスを求める条件

素サービスを求めるための条件として次の二つがある。

(1)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越えてはいけな。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、入力アーク数で有界である。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、ベトリネットグラフのモデルで例を示す。

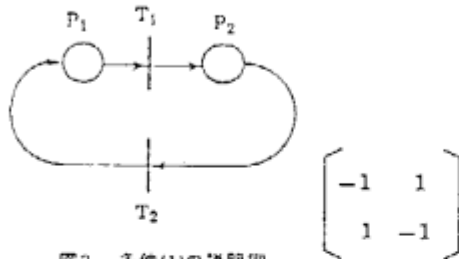


図3. 条件(1)の説明図

この行列のT-invariantは(n,n)となる。初期マーキングを $\mu=(2,0)$ とすると、 $T_1 \rightarrow T_2$ 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、 $T_1 \rightarrow T_1 \rightarrow T_2 \rightarrow T_2$ と発火できるが、これは、 $T_1 \rightarrow T_2$ 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。

(2)トークン間の関係として次の二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。

②①のトークンの遷移により関係付けられたトークンで①の関係を持たない場合。例えば、三個のトークンa,b,cが存在し、aとb,bとcが①の関係をもち、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。

4.4 隠れたサービスの基本概念

図4は、二つの異なるユーザ要求から作成したPSGの一部とそれを統合して作成したTSGである。

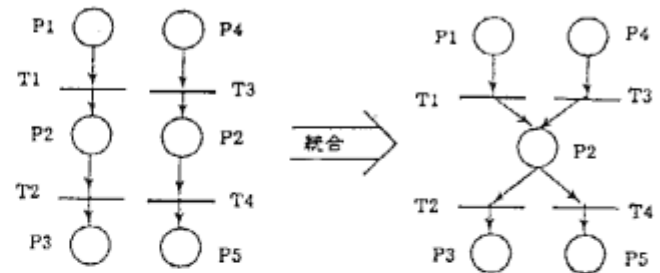


図4. 新たなルートの発見例

図4によると、ユーザの要求したサービスは $T_1 \rightarrow T_2$ 及び $T_3 \rightarrow T_4$ である。しかし二つのPSGを統合したTSGより $T_1 \rightarrow T_4$ 及び $T_3 \rightarrow T_2$ についても求めることができる。この時の $T_1 \rightarrow T_4$ 及び $T_3 \rightarrow T_2$ が、ユーザ要求に抜けがあるとシステムが判断するサービスである。但し最終的にそのサービスを認めるかどうかは、ユーザの判断によらなければならない。本システムではユーザとインタラクションを取ることでそのサービスを仕様を含むかどうかの最終的な判断を行う。

5.統合後のTSGから隠れたサービスが求まる例

実際に、図2のベトリネットグラフにより隠れたサービスを求めてみる。図2で統合する前の3つのサービスのベクトルは、以下の3ベクトルである。

$$\begin{aligned} p &= (1, 0, 1, 0, 0, 1, 1, 1) \\ q &= (0, 0, 0, 1, 1, 0, 0, 0) \\ r &= (1, 1, 0, 0, 0, 0, 0, 0) \end{aligned}$$

この3つのベクトルは一次独立なベクトルである。また、その他の一次独立なカーネルベクトルは、以下のものである。

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

それについて発火順序を求めると、 $T_1 \rightarrow T_3 \rightarrow T_4$ のルートが成り立つことがわかる。またベクトルsは、4.3項の条件(1)、(2)も満たす。よってベクトルsは発火可能ベクトルとなり、隠れたサービスといえる。次にベクトルp,q,r,sによる一次従属なベクトルの中に、隠れたサービスがあるかを調べる。任意の発火可能ベクトルは次のようになる。

$$f(a) = ap + bq + rr + ds$$

$$a, b, r, d \text{ は整数, ただし } f(a) \text{ の要素は非負}$$

このf(a)の中で、4.1項の項目(3)を満たし、かつ4.3項の条件(1),(2)を満たすベクトルは、p,q,r,s及び

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

の5つである。よって図2のベトリネットによる行列から、ベクトルp,q,r以外の素サービスによる発火可能ベクトルs,tを求めることができる。このことは、p,q,rを統合することにより、隠れたサービスs,tが求まることを意味する。

6 おわりに

ベトリネットを用いて表現した設計仕様に、関数行列を使用した解析を行って隠れたサービスの発見に適用した。今後は、色つきトークンの導入されたシステムにも関数行列を使用した解析の可能性を、調べてみるつもりである。

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, 桑田, 田中, 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] 坂井, 野: "ベトリネットの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)

4.2 サービスの階層的定義

設計仕様におけるサービスには、サービス(service)、素サービス(prime service)、隠れたサービス(hidden service)がある。

①サービス(service)

サービスとは、各リソースがアイドル状態である初期マーキングから最終的に初期マーキングに戻る発火可能ベクトルで示される発火列である。

②素サービス(prime service)

ペトリネット行列における発火可能ベクトルの数は、一次従属なものまで考えると、無限である。そのためサービスの数も無限になると考えられる。このことは、多くのサービスが、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ていることを意味する。素サービスとは、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ことである。但し、サービスは発火列、発火可能ベクトルはT-invariantで表されるために一次従属な発火可能ベクトルの中にも素サービスに対応するものがある。その例を5章に示す。4.3項に素サービスを求めるための条件を示す。

③隠れたサービス(hidden service)

曖昧なユーザ要求から設計仕様を作成するためには、抜けや矛盾に対する処理を行う必要がある。ユーザ要求からPSG(素サービスに対応する設計仕様)を作成する過程において、抜けや矛盾が生じた場合は、システム内の知識ベースを参照し、ユーザとインタラクションを取ることで、抜けや矛盾を解消している。しかし、ある一つのユーザ要求がすべて抜けている場合、システムによりその抜けたユーザ要求に対する素サービスを求めることは不可能に近い。複数のユーザ要求を統合する本システムでは、ユーザの気付かなかった新しいルートが作成される場合がある。一例を図4に述べる。またその中には、抜けたユーザ要求に対する素サービスが存在することがある。本システムでは、その性質を利用することでサービスの抜けを解消する一手法にしている。このような、ユーザの気付かなかったルートに対する素サービスのことを隠れたサービス(hidden service)と呼ぶ。4.4項に、その基本概念について述べる。またそれを求めた例を5章に述べる。

4.3 素サービスを求める条件

素サービスを求めるための条件として次の二つがある。

- (1)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越えてはいけな。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、入力アーク数で有界である。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。

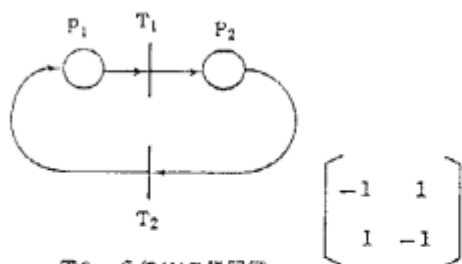


図3. 条件(1)の説明図

この行列のT-invariantは (n,n) となる。初期マーキングを $\mu=(2,0)$ とすると、 $T_1 \rightarrow T_2$ 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、 $T_1 \rightarrow T_1 \rightarrow T_2 \rightarrow T_2$ と発火できるが、これは、 $T_1 \rightarrow T_2$ 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。

- (2)トークン間の関係として次の二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。

②①のトークンの遷移により関係付けられたトークンで①の関係を持たない場合。例えば、三個のトークン a, b, c が存在し、 a と b, b と c が①の関係をもち、 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。

4.4 隠れたサービスの基本概念

図4は、二つの異なるユーザ要求から作成したPSGの一部とそれを統合して作成したTSGである。

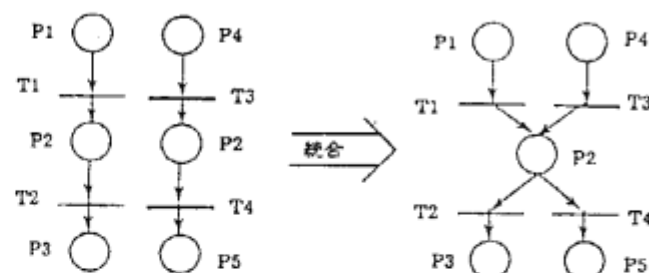


図4. 新たなルートの発見例

図4によると、ユーザの要求したサービスは $T_1 \rightarrow T_2$ 及び $T_3 \rightarrow T_4$ である。しかし二つのPSGを統合したTSGより $T_1 \rightarrow T_4$ 及び $T_3 \rightarrow T_2$ についても求めることができる。この時の $T_1 \rightarrow T_4$ 及び $T_3 \rightarrow T_2$ が、ユーザ要求に抜けがあるとシステムが判断するサービスである。但し最終的にそのサービスを認めるかどうかは、ユーザの判断によらなければならない。本システムではユーザとインタラクションを取ることでそのサービスを仕様を含むかどうかの最終的な判断を行う。

5.統合後のTSGから隠れたサービスが求まる例

実際に、図2のペトリネットグラフにより隠れたサービスを求めてみる。図2で統合する前の3つのサービスのベクトルは、以下の3ベクトルである。

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

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

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

この3つのベクトルは一次独立なベクトルである。また、その他の一次独立なカーネルベクトルは、以下のものである。

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

それについて発火順序を求めると、 $T_1 \rightarrow T_3 \rightarrow T_4$ のルートが成り立つことがわかる。またベクトル s は、4.3項の条件(1)、(2)も満たす。よってベクトル s は発火可能ベクトルとなり、隠れたサービスといえる。次にベクトル p, q, r, s による一次従属なベクトルの中に、隠れたサービスがあるかを調べる。任意の発火可能ベクトルは次のようになる。

$$f(a)=\alpha p+\beta q+\gamma r+\delta s$$

$\alpha, \beta, \gamma, \delta$ は整数、ただし $f(a)$ の要素は非負

この $f(a)$ の中で、4.1項の項目(3)を満たし、かつ4.3項の条件(1)、(2)を満たすベクトルは、 p, q, r, s 及び

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

の5つである。よって図2のペトリネットによる行列から、ベクトル p, q, r 以外の素サービスによる発火可能ベクトル s, t を求めることができる。このことは、 p, q, r を統合することにより、隠れたサービス s, t が求まることを意味する。

6 おわりに

ペトリネットを用いて表現した設計仕様に、関数行列を使用した解析を行って隠れたサービスの発見に適用した。今後は、色つきトークンの導入されたシステムにも関数行列を使用した解析の可能性を、調べてみるつもりである。

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, 桑田, 田中, 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] 斎藤, 西: "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)