〔公　開〕

ＴＲ－Ｃ－０１５２

異常な状態への遷移の検出精度向上手法の検討

原田　英二
Eiji HARADA

１９９６　３．１５

ＡＴＲ通信システム研究所

異常な状態への遷移の検出精度向上手法の検討

平成８年３月

原田　英二

目次

1　まえがき

通信ネットワークの高度化にともない多様なサービスの早期開発がもとめられており、そのために仕様の早期確定が必要である。仕様確定における重要な課題にサービス合成検証がある。これは、新規サービスと既存サービスを合成したときに発生する不具合すなわち競合を検出し、解消するものである。

サービス合成検証については、競合の一種である意味的矛盾に着目し、異常な状態への遷移を定義して、その検証手法を検討してきた[1]。異常な状態への遷移とは、合成サービスによって生成する状態遷移に被合成サービス（合成する前の各々のサービス）単独では生成されない状態があらわれることである。この検証手法は2つのサービスを機械的に合成し、2つの被合成サービスと合成サービスの状態遷移を比較判定しながら、異常な状態を検出して設計者に提示するものである。

この検証手法によって2つのサービス仕様を合成したときに生じる意味的矛盾を機械的に検出することが可能となり、また、設計者が検証しなければならない合成サービスの状態遷移の数を大幅にしぼりこむことができた。

ただし、この手法によって検出される異常な状態には実際の競合に比べて冗長な状態、つまり、競合ではないものがあった。競合が必ず検出されないと設計支援手法としては意味がなくなるため、検出もれを生じさせずに、その競合でない状態の検出を機械的にできるだけ削除すること、すなわち検出精度の向上が課題であった。

本稿では、まず、合成検証の異常な状態への遷移について、従来の検出手法について説明する。次に、問題点である冗長な検出について発生原因の考察と原因に応じた対策を検討する。そして、従来の検出手法の改善手法を提案してその評価を行う。最後にまとめと今後の課題について述べる。

2　従来の異常な状態への遷移の検出法

2・1　対象とするサービス仕様

（1）サービス仕様の定義

サービス仕様ＳＰＥＣは、通信システムをブラックボックスとし、端末の状態と端末からの入力に基づく状態遷移モデルとして以下のように定義される。

ＳＰＥＣ＝＜Ｔ，Ｓ，Ｅ，Φ，s0，f0＞

Ｔ：端末の集合

Ｓ：サービスの状態の集合

Ｓの要素Ｓは端末Ｔの状態τ (T) の集合として定義される。τ (T) はより細分化され

た状態記述要素pの集合で定義される。

E：端末から入力されるイベントの集合

Φ：サービス状態の遷移関数 Sj＝Φ（Si，ei）　　（Si,Sj∈Sかつei∈E）

s0：初期状態　　（s0∈S）

f0：最終状態　　（f0∈S）

（2）仕様記述言語

通信サービス仕様から異常な状態への遷移を機械的に検出するための仕様記述言語は、仕様の形式的な記述が可能であり、単独サービス仕様の機械的な足し合わせ（和集合をとる）が可能であることが要求される。これらの機能は、2・1・(1）項のサービス仕様の遷移関数Φを単一の状態遷移を規定したルールの集合とそれらのルールの適用関数として規定することにより実現している。

Φ＝＜R，Σ＞

R：ルールの集合。要素をrで示す。ｉｆ－ｔｈｅｎ型であり、条件部と動作部をコロン（：）で区切り以下のように記述する。

r＝cs e：ns

cs：現状態。端末の状態記述要素pを用いて記述する。

ns：次状態。端末の状態記述要素pを用いて記述する。

e：遷移の起因となるイベント。e∈E

Σ：ルールの適用関数であり、2つの関数M，Wから構成される。

M：ルールのマッピング関数。イベントeが選択された時にサービス状態Skに対して、ルールrの条件部csがSk⊇csを満足するrを抽出する。　　r＝M（Sk,e）

W：ルールrが適用された結果、Skをそのルールの動作部nsに記述されている内容に従って書き換えて次状態Slを生成する関数。　　Sl＝W（r,Sk）

2・2　異常な状態への遷移の定義

2つの被合成サービス仕様、Xサービス仕様とYサービス仕様からXとYサービスの合成サービス仕様を生成する場合を考える。各被合成サービス仕様は共に正常な状態遷移をすることが設計者により確認されていることを前提条件とする。

XとYのサービスの状態SxiとSyiが重なりあった状態Si＝{Sxi∪Syi}があったとする。

SiにおいてXサービスの仕様が実行されてSjが生成されたとする。Xサービスの次状

態SxjはXサービスの仕様からも求められる。Yサービスの属する端末の状態Syjは以下の式で求められる。

Syj＝ {{Syi － {Sxi ∩ Syi}} ∪ κ j}

κ j：状態Sjにおいて、XとYサービスが重なりあっている状態k個の端末の状態

このSyjがSyj∈Ｓｙ（Yサービスの状態の集合）を満足しないことを異常な状態への遷移と定義する。

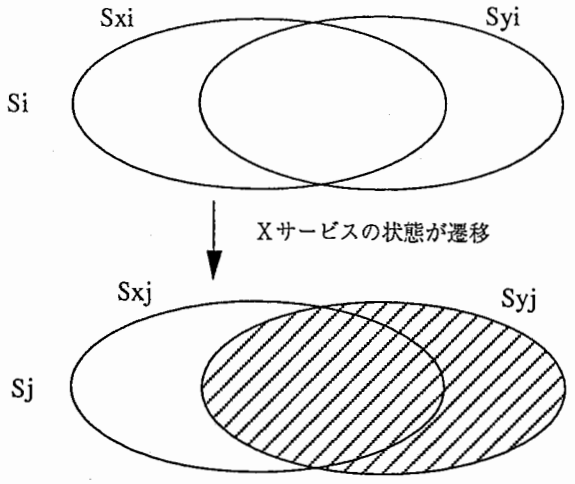

図1　異常な状態への遷移

2・3　異常な状態への遷移の検出アルゴリズム

2・1・(2）項で示した仕様記述言語を用いて記述したサービスを対象として、異常な状態への遷移を機械的に検出する手法を示す。

入力：X，Yサービス仕様を規定したルールの集合Ｒｘ，Ｒｙ

手順：

（1）合成した状態の生成

Ｒｘ，Ｒｙから1つずつルールを選択して、それらのルールの現状態の和集合Siを取る。

（2）Siが実際に出現するか判定

初期状態からＲｘ，Ｒｙに属するルールを順次適用してSiが存在すれば、出現すると判定する。

（3）ルールの実行後の合成サービスの状態の導出

状態Siにおいてrxを適用し、遷移後の状態Sjを導出する。

Sj＝W（rx，Si）（rx∈Ｒｘ）

（4）単独サービスの状態の抽出

SjからYサービスを構成する状態Syjを抽出する。

Syj＝terminal-state（Sj，terminal（Syj））∪ relation-state（Sj,relation（Syj））

terminal-state：サービスのある状態についてその状態に含まれる端末集合の端末自体の状態（他端末とは関係のない端末の状態）の集合だけを求める関数

terminal：サービスのある状態においてその状態を構成する端末の集合を求める関数

relation-state：サービスのある状態についてその状態に含まれる端末集合の他端末とは関係している端末の状態の集合だけを求める関数

relation：サービスのある状態においてその状態を構成する端末のそれらの間の関係の

集合を求める関数

（5）異常な状態への遷移の判定

SyjがYサービスの状態Sy（∈Ｓｙ）として出現するか判定する。Yサービスの状態の集合ＳｙはＲｙを初期状態から順次適用しすべてのルールを適用して生成する。

3　検出結果の分析

従来の検出アルゴリズムによる冗長な検出の内容を分析するため、検出実験を行った。実験は、検出アルゴリズムを実現したプロトタイプシステムで実際のサービス仕様を用いて行った。システムで用いた仕様記述言語はＳＴＲである。ＳＴＲは2・1・(2)項の機能を実現した仕様記述言語である［2］。

実験では8つの実際のサービスについて相互に合成し、異常な状態の検出を行った。使用したサービスはキャンプオンビジー（ＣＣＢＳ)、着信転送（ＣＦＶ)、発番号表示（ＣＮＤ)、キャッチホン（CW)、ダイレクトコール（ＤＣ)、着信拒否（ＤＴ)、発信拒否（ＤＯ)、三者通話（３ＷＣ）である。

検出された状態は、異常として判断した理由を調べるとともに、競合／非競合に分類した。分類結果を図2に示す。競合の判定はベルコアのサービス仕様書［3］を参考にして、通信サービス設計の専門家が行った。

検出された競合でない状態は次の2種に特徴づけることができた。

（1）XサービスとYサービスの合成状態のYサービスを構成する状態に属する端末Tの状態τ (T) が、Xサービスだけで定義している状態記述要素だけで構成されているために異常として検出される。ただし、τ (T) の意味を考えればYサービス仕様と矛盾しない場合、非競合と判定できる。

プロトタイプによる検出数（総数１１２６）

	CCBS	CFV	CND	CW	DC	DT	DO	3WC
CCBS		236	72	91	192	58	52	201
CFV	1		6	6	3	6	0	37
CND	2	1		12	3	0	0	30
CW	0	1	2		6	8	4	37
DC	3	1	1	0		0	0	28
DT	2	1	0	0	0		0	28
DO	0	0	0	0	0	0		10
3WC	2	1	2	1	0	2	1	

競合である状態の数（総数２４）

図2　異常な状態への遷移検出結果

（2）XサービスとYサービスの合成状態のYサービスを構成する状態に属する端末Tの状態τ (T) が原因で異常として検出されるが、端末TがYサービス仕様を検証するために不要な端末である場合、非競合と判定できる。

（1）にあたるのは異常な状態総数の２０％程度であり、（2）にあたるのは８０％以上を占めていた。ただし、両方の特徴を有するものも多かった。

上記のほかに、同じルールを適用して同じ異常な状態へ遷移したものが複数あるため、検出数が１０％程度増加した場合があった。これは本来、1件として検出すべきであり、検出アルゴリズムをプロトタイプシステムで実現した時のインプリメントの誤りとして分析の対象からはずした。

4　未定義な状態の意味を獲得することによる競合判定法の検討

２項で定義した異常な状態への遷移（図３）を基に説明する。

Xサービスだけで定義され、Yサービスでは未定義の状態記述要素pは、Yサービスの状態の集合Ｓｙに属するSyには絶対あらわれない。したがって、Syjに属する端末Tnの状態τ (Tn) が、Xサービスだけで定義されている状態記述要素だけで構成されている場合、Syj=Syとなる Syは存在しないため、必ず異常な状態として検出される。

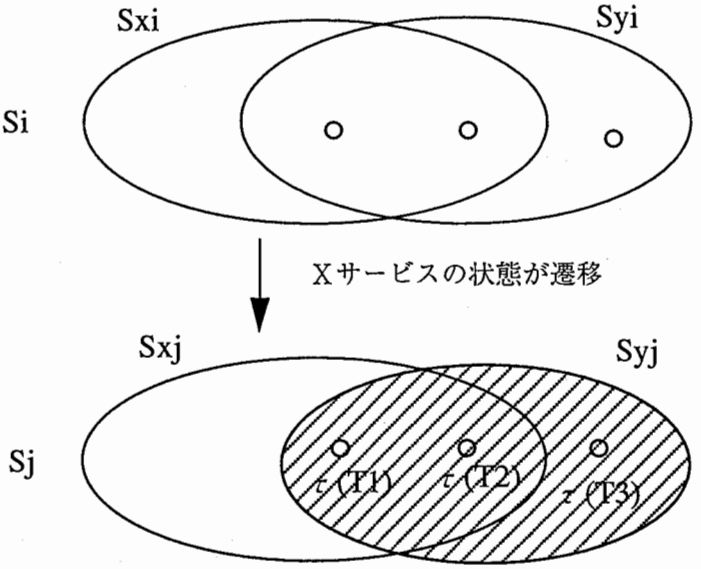

図３　異常な状態への遷移

しかし、τ (Tn) の意味を検討すると、Yサービスにとって非競合と判定できるものもある。例として、Xサービス＝３ＷＣサービス、τ (T1) ＝リコールダイヤルトーン（Xサービスだけで定義された状態)、Yサービス＝ＤＴ（着信拒否)，ＤＣ（ダイレクトコール）サービスの場合を示す。未定義状態と各サービスの意味を図４の通信機能の整理から獲得する。

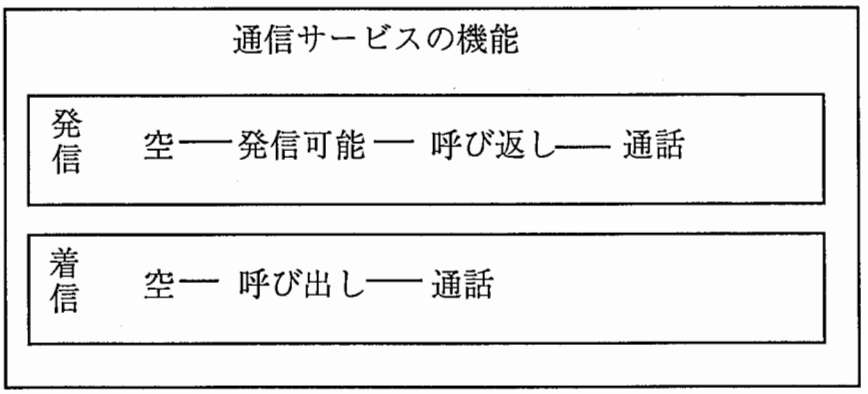

図４　通信機能の整理

τ (Tn) の意味＝発信機能の発信可能を意味する

Ｙサービスの規定する機能と意味

ＤＣ　　＝　発信機能については、呼び出し相手が特定のため発信可能にならないことが規定されている

ＤＴ　　＝　発信機能について規定なし

未定義な状態の機能と単独サービスの規定する機能が異なる場合は、意味的矛盾＝競合は起こらない。したがって、競合／非競合は以下のとおり判定できる。

ＤＣサービス＝競合　　ＤＴサービス＝非競合

上の例を含めたいくつかの例から、τ (Tn) の意味およびＹサービスの仕様の規定する機能を形式的な通信サービス仕様から獲得する方法を検討したが、現在のところ確立できていない。課題となっているのは、τ (Tn) の意味とＹサービスの意味の比較を可能とするための通信機能の整理方法である。

5　異常判定基準の変更による検出精度向上

２項のサービス仕様の定義、異常な状態への遷移の定義、検出アルゴリズムに基づき、検出精度向上手法を提案する。

5・1　サービスのルールの集合に要求される条件の考察

被合成サービスの設計を行う際はサービスの特徴的な動作(基本サービスとの差分)を規定することが検討されている［4］。これは、被合成サービス（名前をＹサービスとする）を設計する前の端末の動作を規定するルールの集合をＲｏとすると、Ｙサービスの設計を、Ｙサービスを端末に登録することで実現する、Ｒｏに属するルール以外のルールを設計することで行うものである。このルールの集合をＲｙｄとする。

Ｙサービスのルールの集合Ｒｙは、Ｒｏに属するルールを含んでいる。Ｙサービスの状態の集合ＳｙはＹサービスの初期状態S0yにＲｙに属するルールを順次適用することで生成するのであり、Ｒｏのルールがないと、生成できない状態がある恐れがあるからである。ＲｙはＲｙｄに属するルールも含んでおり、Ｒｙ＝Ｒｙｄ＋Ｒｏである。

５・２　サービスの状態を構成する端末に要求される条件の考察

端末T1へのサービス登録を状態記述要素m-Y (T1)、初期状態の記述要素をp0y (T1) とすると、S0y ＝ m-Y (T1) ,p0y (T1) である。

Ｓｙに属する状態を構成する端末はルールの適用によって決定される。図５に示す通り、Ｙサービスの状態Sykに対してルールry (τ 3 (A) , τ 4 (B) ｅ : τ 5 (A) , τ 6 (B)) (ry ∈ Ｒｙ) を適用するとは、ルールryの条件部と状態Sykとの一致する部分を、ルールのry動作部に記述する内容にしたがって書き換えて次状態Slyを生成することであり、Slyの端末は適用したルールryの動作部により決定される。Ｙサービスの初期状態S0yを構成する端末はサービスを登録したT1だけであるが、Ｙサービスのルールの適用によってＹサービスの状態Syを構成する端末は変化する。

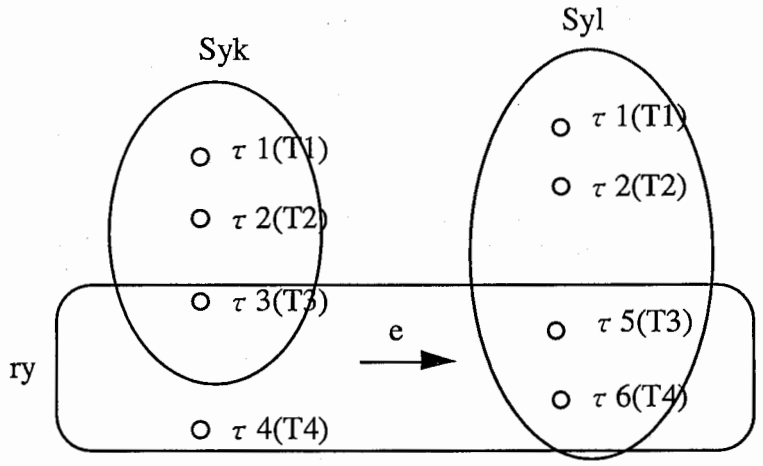

図５　ルールの適用

５・３　サービス仕様の競合検証に必要な端末の定義

５・１項で述べたように、Ｙサービスの状態の集合Ｓｙを生成するためにはＲｙに属するルールがすべて必要である。しかし、ＸサービスとＹサービスの合成状態のＹサービスへの意味的矛盾をもとめるためには、Ｙサービスを登録することで実現する仕様が規定されたＲｙｄに属するルールによる動作との矛盾をもとめれば十分であると考えられる。

生成された状態Syを構成する端末単位で見ると、Ｙサービスを登録した端末は検証に不可欠だが、それ以外の端末には、Ｒｙｄに属するルールを適用した結果としても、Ｒｙｄに属するルールを適用する条件としてもあらわれない端末がある。この端末がＹサービスにとって異常な状態であっても、Ｒｙｄに属するルールの適用条件にも適用結果にも矛盾しない。すなわち、Ｙサービス登録により実現する仕様と矛盾しないので競合ではない。このように、Syを構成する端末のなかで、Ｙサービス仕様の競合検証に必要な端末の集合ＴｎｙはＲｙｄを用いて以下のように定義される。

・Sy＝W（ryd,Spy）となるＲｙｄに属するすべてのルールrydについて、そのルールの動作部nsの記述により書き換える対象となったSyの端末の集合をＴｓｐｙとする。

・Sny＝W（ryd,Sy）となるＲｙｄに属するすべてのルールrydについて、そのルールの条件部csが一致するSyの端末の集合をＴｓｎｙとする。

・Ｙサービスを登録した端末をTryとする。

（Spy,Sny∈Ｓｙ）

Ｔｎｙ＝Ｔｓｐｙ＋Ｔｓｎｙ＋Try

５・４　異常判定基準の変更の提案

（１）Ｘ，Ｙサービス合成前にＲｙをＲｙｄとＲｏに分離する。

（２）Ｓｙに属するすべての状態Sy（Syを構成する端末Ｔの状態τ (T) の集合）から、Ｙサービス仕様の検証に必要な端末の集合Ｔｎｙに基づいてSyの部分の集合Ssy（T∈Ｔｎｙとなる端末Ｔの状態τ (T) の集合）を求める。

（３）異常な状態として検出された状態Syjについて、Syjを構成する端末から問題となった端末の状態をのぞいた状態がSsyと一致するならばそのSyjは正常と判定する。

６　提案方式の評価

提案方式で正常と判定したSyjはSsyを満たすため、Ｙサービス仕様とは矛盾しないので、非競合である。

また、２項で述べたプロトタイプシステムによる異常な状態への遷移検出結果について、８つのサービスの異常判定基準状態Ｓｓを生成して、競合判定を行った。その結果、提案方式で検出したなかに異常な状態への遷移による競合はすべて含まれていた。また、図６に示すようにプロトタイプの検出数と提案方式による検出数を比較すると１１２６：８７であり、約９０％の削減ができた。したがって、検出精度向上の効果があると判断できる。

プロトタイプによる検出数（総数１１２６）

	CCBS	CFV	CND	CW	DC	DT	DO	3WC
CCBS		236	72	91	192	58	52	201
CFV	3		6	6	3	6	0	37
CND	2	1		12	3	0	0	30
CW	21	2	4		6	8	4	37
DC	3	1	1	4		0	0	28
DT	2	1	0	1	0		0	28
DO	0	0	0	4	0	0		10
3WC	14	5	2	6	4	4	2	

提案方式による検出数（総数８７）

図６　従来の方式と提案方式の比較

サービス合成時に生成される状態数と従来方式による異常な状態検出数の比率は１０：１であるとの研究結果［１］があることから、提案方式により異常な状態を検出することで、検証対象数をサービス合成時に生成される状態数とくらべて１／１００に削減することができた。また、検出した状態数と実際に競合である状態数の比率が４：１であり、精度の高い検出が実現できた。

７　まとめと今後の課題

本稿ではサービス合成検証の異常な状態への遷移について、より精度の高い検出手法を提案し、評価した。

今後の課題としては、４項で述べたように、特定のサービスだけに定義された状態記述要素が存在する場合の対策の形式化がある。新規サービスが増加すると、そのサービスだけに定義された状態記述要素も増加して、冗長な検出として無視できなくなることが考えられるからである。

参考文献

［１］高見、井上、太田「サービス競合における異常な状態への遷移検出法」情報処理学会論文紙１９９５年１月

［２］Bellcore,"LSSGR Features Common to Residence and Business Customers 3 Issue2",1989

［３］Hirakawa, Y. and Takenaka, T : Telecommunication Service Description Using State

Transition Rules, Sixth Int.Workshop on Software Specification and Design,Como,Itary (Oct.1991)

［４］原田、平川、竹中、門田「サービス仕様の自動生成に関する考察」情報処理学会第３９回、５Ｓ－５（１９８９）

［５］井上、高見、太田「通信サービス仕様における異常な状態への遷移解消支援手法」１９９４年電子情報通信学会春季全国大会Ｂ－６２０

［６］原田、河原崎「通信サービス仕様における異常な状態への遷移の解消支援に関する一考察」１９９５年電子情報通信学会春季全国大会Ｂ－７１８

［７］原田、河原崎、太田「異常な状態への遷移の検証手法」電子情報通信学会交換システム研究会ＳＳＥ９５－６２，１９９５年９月