マルチメディア通信と分散処理　52－15
（1991．9．25）

通信サービス動作競合検出方法

原田　良雄　　平川　豊　　竹中　豊文

ＡＴＲ通信システム研究所

あらまし　通信サービスを新たに追加する際には、新規サービス単独の設計だけでなく、新規サービスと既存サービス間の相互作用（サービス絡み動作）を考慮し、システム全体として矛盾なく動作させることが必要である。

一般に、相互作用の数は、既存サービス数が増えると組合せ的に増大するため、その相互作用に起因するサービス動作矛盾の検証は、人手で行う事のできる範囲を越えた作業となる。この負担を軽減するためには、動作矛盾の可能性のある動作を自動検出するとともに被疑対象を絞り込んで設計者に提示する設計支援が有効である。

本稿では、サービス動作競合（動作の非決定性）の問題に着目し、ＳＴＲ（State　Transition　Rule)手法により記述された通信サービス動作規定を対象に、通信サービス動作競合検出手法について議論する。

A Conflict Detection Method for Telecommunication Service Descriptions

Yoshio HARADA　Yutaka HIRAKAWA　Toyofumi TAKENAKA

ATR　Communication Systems Research Laboratories
Sanpeidani, Inuidani, Seika-cho, Soraku-gun, Kyoto, 619-02 Japan

Abstract　When designs for additional services are implemented, in order to establish totally specifications for the existing services and the additional service, it needs to specify not only the operation of the additional service, but also the resulting interactions with other services. The interactions sometimes include service conflicts. As the number of interaction specifications increases with the increase in the number of services, it is very hard for a designer to detect service conflicts.

This article proposes a conflict detection method for telecommunication service descriptions.

1．　はじめに

通信サービスを新たに追加する際には、新規サービス単独の設計だけでなく、新規サービスと既存サービス間の相互作用（サービス絡み動作）を考慮し、システム全体として矛盾なく動作させることが必要である。

一般に、相互作用の数は、既存サービス数が増えると組合せ的に増大するため、その相互作用に起因するサービス動作矛盾の検証は、人手で行う事のできる範囲を越えた作業となる。この負担を軽減するためには、動作矛盾の可能性のある動作を自動検出し、被疑対象を絞り込んで設計者に提示する設計支援が有効である。

本稿では、サービス動作競合（動作の非決定性）の問題に着目し、ＳＴＲ（State Transition Rule)手法[1､2]により記述された通信サービス動作規定を対象に、通信サービスの相互作用[3,4,5]に含まれる通信サービス動作競合検出手法について議論する。

本稿の２章では、サービス動作競合の具体例と、競合検出の基本的な考え方について述べる。３章では、詳細なサービス動作競合検出手法について述べ、４章では、競合検出実験の結果について述べる。５章では、考察とまとめについて述べる。

2．サービス動作競合

2．1　サービス競合の具体例

サービス動作競合の具体例について説明する。図１にＳＴＲ手法で記述された、話中着信動作と着信転送動作の記述例を示す。図１の記述で、「out-dialtone(A)」などは、状態記述要素と呼ばれる物であり、１つの端末の状態は、複数の状態記述要素の集合で記述されている。

図１は、次のような動作を規定している。

話中着信動作：

ある端末（A）が話中着信に加入していて（m-cw(A)）、話中（path(A,B)）のとき、他端末（C）がダイヤル可能状態（out-dialtone(C)）で端末（A）にダイヤル（dial(C,A)）を行うと、端末（A）への話中着信（out-cwringing(A,C),out-ringback(C,A)）となる。

着信転送動作：

ある端末（A）が着信転送を登録（m-cfv(A,B)）しており、着信転送登録先（B）がアイドル（idle(B)）のとき、他端末（C）がダイヤル可能状態（out-dialtone(C)）で、話中（path(A,D)）の端末（A）にダイヤル（dial(C,A)）を行うと、着信転送登録先への着信（out-ringing(B,C),out-ringback(C,B)）となる。

```
(話中着信動作)
out-dialtone(C),m-cw(A),path(A,B)
    dial(C,A):
  out-cwringing(A,C),out-ringback(C,A),m-cw(A),path(A,B).

(着信転送動作)
out-dialtone(C),idle(B),m-cfv(A,B)
    dial(C,A):
  out-ringing(B,C),out-ringback(C,B),m-cfv(A,B).
```

図１　通信サービス動作記述例

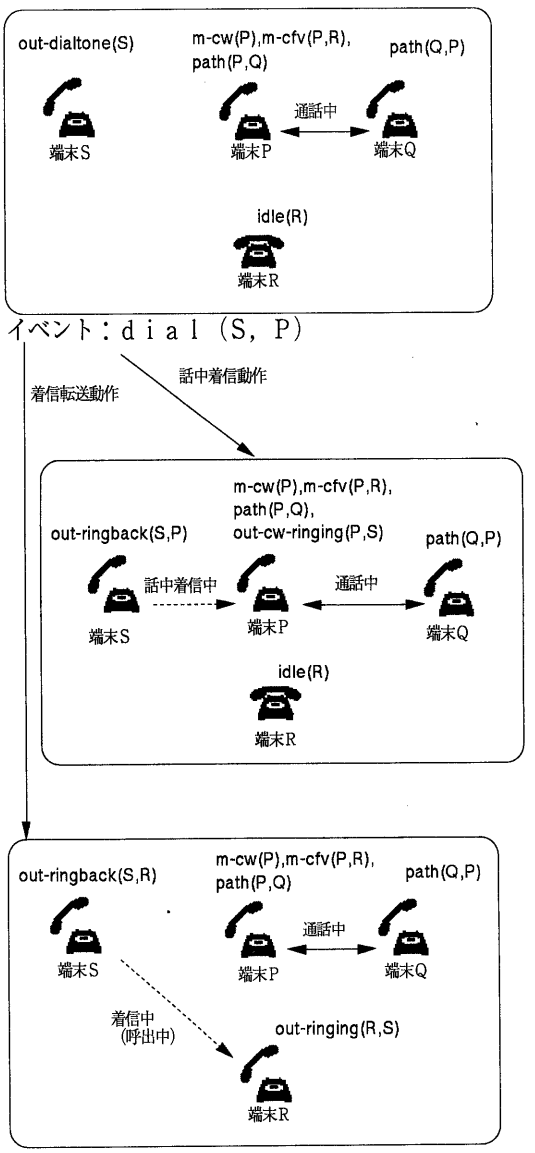

図２　サービス動作競合例

いま、次の状況を想定する(図２、上段)。　端末Pが話中着信に加入し（m-cw(P)）、着信転送を登録し（m-cfv(P,R)）、かつ話中（path(P,Q)）である。端末Rがアイドル状態（idle(R)）である。端末Sはダイヤル可能状態（out-dialtone(S)）である。それぞれの端末の状況を表すと以下のようになる。

端末P＝ {m-cw(P),m-cfv(P,R),path(P,Q)}
端末R＝ {idle(R)}
端末S＝ {out-dialtone(S)}

この状況で、端末Sから端末Pへダイヤルしたとき（イベント：dial(S,P)）、図１の２つの動作規則の条件（始状態）は両方とも満足され、２つの異なる動作が可能な状況(図２、中段、下段)となる。これを動作競合と呼ぶ.

２．２　サービス競合検出の考え方

前節で示したようなサービス競合は、あるイベントが生起した際に、２つの動作規則が共に適用可能な状況に遭遇することにより発生する。この競合状況（例えば前節の端末P,R,Sで作られる状況）は、２つの動作規則（例えば図１に示した２つの規則）から構成することができる。即ち、同じイベントに関する動作規則を２つ選べば、簡単にその両者の条件を満足する状況を形式的に構成できる。しかし、形式的に構成した状況が、常に競合する状況となるわけではない。というのは、形式的に構成した状況が、実際には遭遇し得ない状況となる場合があるためである。これを機械的に判別するためには、複数の端末状態で構成される任意の状況に対して、実際に起こり得る状況か否かの判定が必要になる。我々は、解析対象として不特定多数の端末で構成されるサービスをも考慮しており、この条件では機械が探索しなければならない範囲は無限となり機械判定は一般に不可能である。

これらを踏まえ、本稿では、以下に述べるアプローチを取る。まず、競合動作検出の基本的な流れを以下に示す。

ステップ１）１つの端末動作に着目し、取り得る可能な状態を列挙する。
ステップ２）競合判定対象の２つの規則から、両者の条件を共に満足する状況（規則の競合する状況）を形式的に構成する。
ステップ３）形式的に構成した状況の各端末状態が真に端末の状態として有り得る状態（ステップ１で求めた集合の要素）であるか否かを判定する。

本稿でのアプローチは、厳密な意味での競合の自動検出と次の点が異なる。

（１）ステップ１の端末の取り得る状態の列挙には、設計者の介在が必要となる。

（２）ステップ３で、形式的に構成した状況は競合を引き起こす可能性のある状況であり、最終的な判断は設計者の介在を必要とする。以上のように本手法は設計者の介在を必要としているが、競合の可能性のある状況を絞り込む効果を持っており、競合検出の効率化に有効である。

次節では、この競合検出手法の詳細を説明する。

３．サービス動作競合検出手法

３．１　端末の取り得る状態の数え上げ

端末状態の数え上げは以下の手順による。

手順１）Ｓ＝{ {idle(A)} }とおく。
手順２）Ｓに含まれる端末状態に対して適用可能性のある規則を見つけて、適用した場合の次状態を作成し、それがＳに含まれていない状態であれば、Ｓに登録する。この手順２を、Ｓに登録すべき状態が無くなるまで繰り返す。

上記の手順２において、規則には、一般に複数端末の条件が記述されているが、１つの端末の状態遷移のみに着目しているため、適用可能性があるというだけの判断基準では、実際のシステムでは適用されることのない規則が適用され、有り得ない遷移動作を生ずることがある。従って、設計者が介在し、正しく端末状態の数え上げが行われていることを確認する必要がある。

以下では、数え上げられた端末の可能な状態の集合を、「実状態集合」と呼ぶ。

３．２　競合検出手順

（１）簡単な例による説明

図１に示した規則を用いて、簡単に説明する。

既に、動作記述の集合を用いて、端末の実状態集合（端末の取り得る状態の集合）の作成が終了していると仮定する。また、実状態集合中に、「out-dialtone(A)」「path(A,B),m-cw(A),m-cfv(A,C)」「idle(A)」の各状態が含まれているものとする。

まず、図１に示した２つの規則から、規則適用の条件を抜き出すと以下のようになる。

規則１）　out-dialtone(C1),m-cw(A1),path(A1,B1)
規則２）　out-dialtone(C2),idle(B2),m-cfv(A2,B2)

この記述では、２つの規則で用いられている変数を、添字1と2で区別している。生起するイベント（規則１では、dial(C1,A1)、規則２では、dial(C2,A2)）が同一のものなので、C1=C2, A1=A2なる条件を付加する。

まず、端末(C1)を考えると、「out-dialtone(C1)」という状態が規則１、２から作成できる。予め求めておいた実状態集合を検索することにより、これが端末の取り得る状態であることが確認できる。

次に、端末(A1)を考えると、規則１の「m-cw(A1),path(A1,B1)」はA1=A2という条件から規則２の「m-cfv(A2,B2)」と複合されて、「m-cw(A1),path(A1,B1),m-cfv(A1,B2)」なる状態が作成できる。この状態も、端末の取り得る状態であることが確認される。

規則２より端末B2に関しては、「idle(B2)」という状態が作成でき、これが、端末の取り得る状態であることが確認できる。

従って、２つの規則から作成した端末の状態は実際に実現する可能性があり、競合を引き起こす可能性があると判断する。

競合検出手順の全体の流れを図３に示す。基本的な流れは、２つの規則の条件を合成して、各端末毎に、状態を作成し、それが実状態として存在するか否かを判定する手順の繰り返しである。

ここで示した例では、比較的簡単に競合検出ができるが、一般的には、もう少し複雑な処理が必要となる。以下では、やや詳細に立ち入って、ポイントを絞って説明する。

（２）変数間の制約の管理

上記でも一部触れているが、検出過程では、２つの規則で用いられている変数間（添字1の付いた変数と、添字2の付いた変数間）の制約関係を管理する必要がある。変数間の制約は、２つの規則のイベントの変数を同一化する際に付加されるが（図３(1)）、これ以外でも、２つの規則の条件を同時に満足する端末の状態を作成する際や（図３(2)）、作成した端末状態にマッチした実状態を発見した場合に発生する（図３(3)）。以下に例を示す。

例１）端末の状態を「p1(A1,B1),p2(A2,C2)」（A1= A2）とする。このとき、該当する実状態として、「p1　(A,B), p2(A, B)」という状態のみが存在していたとする。この実状態では、p1の第二引数と、p2の第二引数が同一であるという制約を持っているので、これを反映し、B1=C2なる制約条件が付加される。

例２）上記と同じ端末の状態で、該当する実状態が「p1(A,B),p2(A,C)」なる状態のみであった場合には、逆

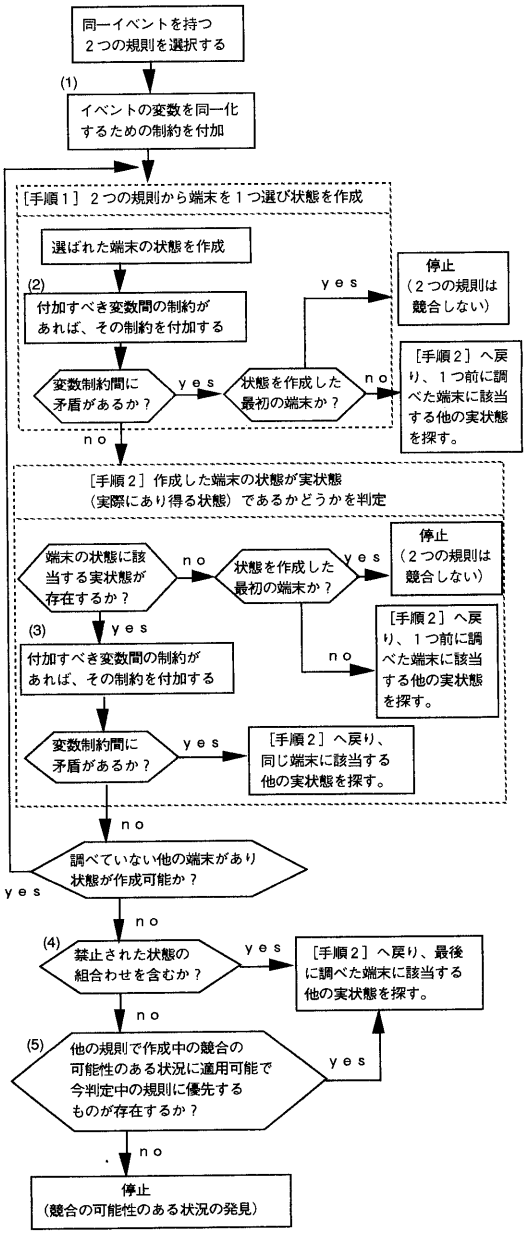

図３　競合検出手順の流れ

に、p1の第二引数と、p2の第二引数が異なっているという制約条件があるため、これを反映し、B1≠C2なる制約条件が付加される。

また、このように付加された制約条件は、更に他の制約条件を生み出す場合がある。例えば、規則から抽出した条件が、以下である場合を考えてみよう。

規則1） p1(A1,B1),p3(B1,C1)
規則2） p2(A2,B2),p3(B2,C2)

ここで、イベントの変数の同一化により、A1=A2なる制約が付いているとする。このとき、端末A1に対する状態を作成すると、「p1(A1,B1),p2(A1,B2)」となる。このとき、該当する実状態が「p1(A,B),p2(A,B)」のみである場合を考えると、B1=B2という制約条件が加わる。このとき、端末B1の状態を作成すると、B1=B2であるため、「p3(B1,C1),p3(B2,C2)」となるが、p3という名前の状態記述要素を複数持つことになる。しかし、「状態記述中には同じ名前の状態記述要素は（特定の物を除いて）1つしか存在しない」という制約を我々のシステムでは採用しているので、これを、「p3(B1,C1)」と書き、制約条件C1=C2を追加する。

更に、次々に付加される制約条件全体に矛盾が無いことを管理する必要がある。例えば、規則から抽出した条件が次のような場合を考えてみよう。

規則1） p1(A1,B1),p3(A1,C1),p4(B1,D1)
規則2） p2(A2,B2),p3(A2,C2),P4(B2,C2)

イベントの変数の同一化により、A1=A2,C1=C2なる制約があるとする。端末A 1に対応する状態は、「p1(A1,B1), p3(A1,C1),p2(A1,B2)」である。このとき、端末A 1に該当する実状態として「p1(A,B), p3(A,C),p2(A,B)」のみが存在しているとすると、B1=B2なる制約が付加される。この制約が加わることにより、端末B1の状態作成時に、D1=C2なる制約が新たに付加される。しかし、この制約の追加により、D1=C2とC1=C2が同時に成り立つことが必要となる。これは、規則1で異なる値を持たなくてはならない2つの変数D1とC1に同じ値を持つことを要求するため、矛盾を生ずる。この様に矛盾が生じた場合は、ここで作成した状態は実現し得ない状況であると判断できるので、競合の可能性はない。

（3）他の規則との相互関係のチェック

次のような2つの規則を考えてみよう。

規則1） p1(A,B) ev(A): p3(A,B).
規則2） p2(A,B) ev(A): p4(A,B).

このとき、実状態集合に「p1(A,B),p2(A,B)」が含まれている場合には、この規則1）と2）による複合状態は、競合の可能性があると判定される。

ここで、次のような別の規則が存在している場合を考えてみよう。

規則3） p1(A,B),p2(A,B) ev(A): p3(A,B).

規則3の適用条件は、規則1と規則2の条件を含んでおり、規則3が適用可能な状況では、常に、規則1と規則2が適用可能である。しかし、ＳＴＲ記述手法では、この様に、「適用条件に完全な包含関係が存在する場合には、包含する規則を適用する」としており、この場合は規則3を優先的に適用する。

従って、規則間で競合の可能性があると思われてもその規則に優先する規則があれば、その状況では競合とならない。このケースでは、規則3の存在により、規則1と規則2の競合する状況とはならない（図3(5)）。

（4）検出精度向上のための知識（排他条件）

これまで述べた処理では、競合する状況の候補を構成している各端末の状態が、実際に有り得る状態か否かを判定している。ここでの判定は、競合する状況の候補に対して、広域的（複数端末にまたがる条件）で有り得ない状態の組合せを導入し、これに該当するものを「競合する可能性のある状況」から取り除くものである。以下に例を示す。

例：「path(X,Y),idle(Y)」

これは、「端末Ｘが端末Ｙと通話状態」と「端末Ｙが空状態」とは同時に存在しないことを規定している。「競合の可能性のある状況」が、この指定された記述を含む場合には、競合状況としては不適当であると判断して、他の競合状況の探索を進める（図3(4)）。

（5）競合する状況の探索制御

端末の状態に対して、該当する実状態が複数存在する場合には、その候補の各々に対して、異なる競合状況が構成される可能性がある。従って、競合の可能性が無いことを判定するには、全ての可能な状況を調査し、判定を下す必要がある。

例えば、次のような2つの条件が規則から抽出された場合を考えよう。

規則1） 「p1(A1,B1),p3(B1,C1),p4(C1,B1)」
規則2） 「p2(A2,B2),p4(B2,C2),p4(C2,D2)」

また、関係する実状態として、以下がある場合を考える。

実状態１）　「p1(A,B),p2(A,B)」
実状態２）　「p1(A,B),p2(A,C)」
実状態３）　「p3(A,B),p4(A,B)」

このとき、イベントの変数の同一化により、A1=A2であるとき、端末A1の状態に該当するものは、実状態１と実状態２の２つがある。実状態１を選択した場合には、B1=B2なる制約が付加され、さらに、端末B1の状態に実状態３が該当して、C1=C2なる制約が付加される。すると、最後に、端末の状態にp4は１つという制約からB1=D2が付加され、B2=D2が推論されるので、規則２で異なる値を持つべき2つの変数B2とD2が同じ値を持つことになり、矛盾が生ずる。これにより、最初の競合状況作成の試みは失敗した訳であるが、ここで、２つの規則が競合の可能性なしと判断することはできない。というのは、競合の状況として、他の可能性があるためである。これ以外の可能性として、端末A1の該当状態として、実状態２を選択することができる。この場合には、B1≠B2なる制約が付加され、矛盾を生ずることなしに、競合の可能性のある状況が１つ発見される。

このように、競合の可能性ありと判定する場合には、１つの状況を発見すればよいが、競合の可能性なしと判定するためには、可能な全ての状況を探索する必要がある。

４．実験

POTS(Plain Old Telephone Service)、話中着信,着信転送の３サービス記述をもとに、端末の取り得る状態の集合を求め、これを用いて、通信サービス競合検出手法を、一部手作業にて適用し、評価を行った．その結果を以下に示す。

サービス動作規則数（３サービスの合計）：５４
端末の取り得る状態の数　　：７３
（遷移動作数：　１１９２）

判定結果：
動作競合の可能性がある動作規則の組合せ数：３

５．考察とまとめ

通信ソフトウェア設計の上流工程である、サービス設計工程におけるサービス動作競合の検出手法について述べた。

提案した手法により、新規サービス設計の際に、サービス動作競合の可能性のある動作規則の組合せと、その状況を設計者へ絞り込んで提示し、設計者の検討を促す設計支援を行うことが可能となる。（最終の判断は設計者が行う）

実験によりその有効性を一部の例により確認したが、以下のような問題があり、今後、システム試作を行いながら検討、改善を加えていく必要がある。

問題点：

提案したサービス動作競合検出手法は、「動作競合の可能性」がある動作規則の組合せを検出するが必ずしも検出したものがすべて実際に動作競合するとは限らない．その理由としては、以下の２つがある。

（１）端末の取り得る状態集合を求める際、ノイズ（実際には有り得ない状態）が入る可能性がある．（ただし、ノイズに気づけば、外部より指定することで排除することは可能である）

（２）検討対象を絞り込む為に、複数端末の取り得る状況についての制約として、外部から排他条件という形で端末間の同時に成立しない条件を指定する機能をもたしている。しかし、新規サービスを追加するときあらかじめ排他条件を設定することは難しく、実際には起こり得ない状況について「動作競合の可能性がある」と判定される可能性がある。

今後は、試作システムを用いて、対象サービスを増やし、本手法の評価を行う。

参考文献

[1]Hirakawa, Harada, Takenaka, "Behavior Description for A System which Consist of An Infinite Number of Processes", 1990 BILKENT International Conference on New Trends In Communication, Control, and Signal Processing, pp.59 - 68, Ankara, Turkey, July 1990

[2]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」情処３９回全国大会，5S-5, 1989

[3]原田、平川、竹中、「通信サービス追加時のサービス絡み動作設計支援方法」’９１信学春期全国大会，B-534

[4]原田、平川、竹中、「通信サービス追加設計支援方法の一考察」 交換システム研究会 SSE 90-137, 1991

[5]Harada, Hirakawa, Takenaka, "A Design Support Method for Telecommunication Service Interactions", GLOBECOM'91, Phoenix, Arizona, December 2-5, 1991 (to be published)