サービス記述検証支援システム

4H−9

原田良雄　平川豊　竹中豊文
ATR通信システム研究所

1. はじめに

筆者等は、通信サービスを使用者の側からみた、サービス記述手法について検討を進めている。これまでに、通信サービス記述言語の提案[1,2,3]、サービス記述から仕様記述を段階的に詳細化していく手法の構想[4]を提案している。本稿では、サービス記述段階での記述正当性検証を支援するシステムについて述べる。

2. サービス記述手法[1,2,3]

通信サービスの記述法として、現状態、イベント、次状態の形式を取るルール記述を既に提案している。記述例を以下に示す。

```
out(A,dial-tone),idle(B)  dial(A,B):
        out(A,ringback,B),out(B,ringing,A).
```

この記述手法では、一般的には複数個の端末（プロセス）に対する状態遷移（上記例では、プロセスAとプロセスB）を記述する点が特徴である。

このサービス記述手法では、各プロセスの状態をout(A,dial-tone)のような状態記述要素の集合で規定している。そして、ルールは、現状態で指定された状態記述要素を次状態で指定された状態記述要素に置き換えることを意味している。

以下では、サービス記述に対する矛盾検出機能と解釈実行機能について述べる。

3. ルール適用の非決定性

まず、次のような2つのルールを考える。

```
rule1)  p1(A)  flash(A):  p2(A).
rule2)  p1(A)  flash(A):  p3(A).
```

ここで、p1,p2,p3は状態記述要素のラベル、Aはプロセスを識別する変数、flashはイベント名である。

プロセスq1が{p1(q1)}で表現される状態の時、プロセスq1でイベントflashが生起したとする。このとき、rule1に従うと次状態は{p2(q1)}であり、rule2に従うと{p3(q1)}となり、1つのイベントに対して複数の次状態が定義されているという非決定性の問題が生じる。

また、上記2つのルールに加えて、次のようなルールを考える。

```
rule3)  p4(A)  flash(A):  p5(A).
```

このとき、{p4(q1)}のような1つの状態記述要素からなる状態からの動作を考えた際には、rule3)とrule1)との間で、上記で述べたような非決定性の問題は生じない。しかし、プロセスq2の状態が複数の状態記述要素の集合{p1(q2),p4(q2)}で表現され、プロセスq2でイベントflashが生起した場合には、rule1とrule3が共に適用可能であり、rule1を適用した場合には{p2(q1),p4(q1)}、rule3を適用した場合には{p1(q1),p5(q1)}と次状態が非決定的となる。

以下、記述上の利便性を考慮して導入したルール適用時のメタルール（一部の非決定性を解消する役割を担う）と矛盾の定義について述べる。

状態記述要素p1(A)を含み、2つの状態記述要素で表現される以下のような状態を考える。

{p1(A),pi(A)}、pi(A)∈{p2(A),..,pn(A)}

このとき、イベントe1が生起したときの動作として、状態{p1(A),p2(A)}の場合には{p3(A),p2(A)}に遷移し、{p1(A),pj(A)},j≠2の場合には{p2(A),pj(A)}に遷移したいとする。この動作は、p1は、p2が共に状態記述に含まれている場合に限りp3に書き換えられるが、それ以外の場合は、p2に書き換えられるという動作である。

このような例外を含む動作は、次に述べるメタルールを導入することにより、以下の2つのルールで簡潔に記述可能である。

```
rule4)  p1(A)  e1(A):  p2(A).
rule5)  p1(A),p2(A)  e1(A): p3(A),p2(A).
```

[メタルール]

1つのイベントに対して、2つのルールrule-aとrule-bが存在したとする。ここで、rule-aの現状態の状態記述要素の一部でrule-bの現状態が構成されている場合、rule-bの適用を優先する。

メタルールは、より詳細に記述されたルール記述の適用を優先することを意味しており、本来非決定性が生ずるべき記述を記述性の観点から許容している。例えば、上記2つのルールが共に適用可能な場合には、rule5が適用される。

Service Description Verification Support System
Yoshio HARADA, Yutaka HIRAKAWA, Toyofumi TAKENAKA
ATR Communication Systems Research Laboratories

［矛盾の直観的な定義］
１つのイベントに対して２つのルールがあり、このルール間には、メタルールによる優先順位が無いとする。このとき、この２つのルールにより非決定性の問題が生ずるとき、矛盾であると定義する。

4．矛盾の例とその検出

以下の３つの矛盾例を示す。

1)［自明な例］
前記のrule1とrule2では非決定性が生じる。

2)［１つのプロセス状態の書換えルールの例］
rule1とrule3は、状態が次式で表現できるプロセスq1が存在するとき、非決定性が生ずる。
{p1(q1),p4(q1)} または、これを含む集合

3)［複数プロセスの状態書換えルールの例］
次のような２つのルールを考える。
rule6) p1(A),p2(B) ev(A): w1(A),w2(B).
rule7) p3(A),p4(B) ev(A): w3(A),w4(B).
この場合、次のように状態が記述されるプロセスq1,q2が同時に存在する場合に非決定性が生ずる。
プロセスq1: {p1(q1),p3(q1)} を含む
プロセスq2: {p2(q2),p4(q2)} を含む

これら1),2),3)の矛盾は、次の方法により検出可能になる。

J1)ルールシンタックスのチェックによる検出
J2)１つのプロセスの取り得る状態列挙による検出
J3)対象システム上での複数プロセスが同時に取り得る状態列挙による検出

4-1．有限状態機械に対する解析

ルールにより規定される各プロセスの動作が有限状態機械で記述される場合には、J2及びJ3は解析可能である。（条件J2,J3の解析可能性に関しては、文献[5]の状態の到達可能性の議論を参照）従って、この場合には、全ての矛盾は解析により検出可能である。

4-2．拡張形有限状態機械に対する解析

状態変数を持つ有限個の状態で各プロセス動作が規定される場合には、J3の解析は困難である。しかし、J2に関しては人間と機械の共同作業による列挙手法が可能である。詳細は別の機会に譲る。

5．解釈実行システム

サービス記述の視覚的な確認とサービスイメージの伝達の場（ユーザからシステム設計者へ）およびサービスイメージ検証の合意の場を提供するものである。

解釈実行システムは稼働確認系と稼働テスト支援系で構成される。

5-1．稼働確認系

画面に表示された、ある電話機（プロセス）よりイベントを入力し、そのイベントと電話機の現状態とに応じたルールを選択し、状態遷移して、その結果を画面に表示する。この操作を繰り返すことで、サービス記述の動作を検証する。

5-2．テスト支援系

稼働中の異常発生時に、その停止理由を表示する。この異常が実行時にしか判定できないルール間の矛盾によるものである場合には競合したルールが表示されルール間の影響解析とルール修正等の対応を容易にすることができる。

6．おわりに

サービス記述に対し、非決定性により定義された記述の矛盾とその検出手法について述べると共に、解釈実行機能を有する記述検証支援システムの概要について述べた。

今後は、これらの機能の具体化，及び，諸機能を総合化したシステムにおけるマンマシンインターフェイスについて、システム試作を通して検討を進める。

参考文献

[1]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

[2]Y.Hirakawa, Y.Harada, and T.Takenaka, 'A Description Method for Advanced Telecommunication Services', 電子情報通信学会、日韓合同交換システム研究会、SSE89-87,1989

[3]Y.Hirakawa, Y.Harada, and T.Takenaka, 'Behavior Description for a System which consists of an Infinite Number of Processes', BILKENT International Conference on New Trends in Communication, Control, and Signal Processing, 1990

[4]原田、平川、竹中、「サービス仕様の自動生成に関する考察 -- 自動生成機構の考察 --」情処40回全国大会、3R-2、1990

[5]柴田、平川、竹中、「プロセス数に依存しない動作記述における状態の到達可能性解析」情処41回全国大会