

〔非公開〕

TR-C-0107

通信サービス設計および
検証における
形式的支援に関する研究

原 田 良 雄
Yoshio HARADA

1 9 9 4 3 . 1

A T R 通信システム研究所

通信サービス設計および検証における
形式的支援に関する研究

平成7年3月

原田 良雄

内容梗概

近年、電気通信分野での著しい技術革新の進展と通信サービスの普及に伴い、通信サービスに対する利用者の要求が多様化・高度化している。このような需要に対して、迅速な新規サービス開発と提供が必要となり、通信ソフトウェアの効率的かつ迅速な開発技術の確立が早急の課題となっている。通信サービス設計においては、新規サービス単独の動作仕様を規定するだけでなく、新規サービス仕様と既存サービス仕様との相互作用（サービスインタラクション）を解析し、サービス仕様間に生じる矛盾を解消し、サービス全体として矛盾のない仕様を作成する必要がある。ところが、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し、その相互作用に隠れたサービス仕様矛盾の検証は、人手で行う事のできる範囲を越えた作業となっている。相互作用解消の問題は、1991年12月、CCITTによってインテリジェントネットワークのための技術的勧告として提示されたCS1（IN Capability Set 1）においても、今後の重要な課題のひとつに挙げられている。

サービス相互作用解析を機械的に支援するためには、まず、サービス仕様を形式的に記述する必要がある。そのため、通信サービス仕様として利用者の観点から見える動作を対象とし、サービス動作を形式的な規則形式により記述する手法（STR：State Transition Rule）を提案している。また、STR手法を用いて通信サービスは記述されていることを前提とし、新規サービスと既存サービスからサービス合成動作を作成し、そのサービス合成動作を基盤に相互作用に含まれる矛盾検出支援方式を提案している。

本論文では筆者がこれまでに行ってきた、通信サービス設計および検証における形式的支援に関する以下の研究項目について述べる。

① 通信サービス記述方式

- ② 通信サービス設計支援方式
- ③ 通信サービス競合（非決定性）検出方式
- ④ 通信サービス相互作用に含まれる意味的矛盾動作検出支援方式
- ⑤ 通信サービス課金仕様競合検出方式

第1章では，研究の背景，研究範囲について述べ，目的を明確にする．また，本論文の各章の概要について述べる．

第2章では，通信サービス記述方式について述べる．利用者の視点から通信サービス動作を，形式的な規則形式の集合で記述するS T R（State Transition Rule）手法について述べる．具体的には，S T R手法の規則記述と規則適用方法について述べる．また，S T R手法を前提とした場合の新規サービス設計時において，1）既存サービスの規則を再利用することにより特徴的な動作のみ記述すればよく，記述量の削減化が可能になること（例えば，着信転送サービスの記述では40％程度の記述量ですむこと），2）P B Xサービスでは94％のサービス機能が記述できることを示す．

第3章では，通信サービス（相互作用）設計支援方式について述べる．新規サービスの設計支援の手法および相互作用設計支援の手法を述べる．相互作用に含まれる矛盾の検出方式の詳細については，第4章以降に述べる．

新規サービス設計支援においては，設計者が視覚的な動作確認画面を用いて，サービスの動作に従ってイベントを入力することにより，画面に表示される適用規則と状態の推移（動作）を確認していくプロトタイピング・アプローチを示し，また，2）初期状態から適用可能な規則を順次適用させて生成した動作を状態推移表の形式で設計者に提示し，網羅的な確認を促す方法を示す．両者を組み合わせて用いることによ

り、新規サービスの要求確認が容易に実現できた。

相互作用設計支援においては、1) 検証項目の分類、2) 検証項目の説明と検出方法について述べる。また、規則修正時の影響波及範囲の検出も可能になることを示す。また、新規サービス設計支援および相互作用設計支援に共通に必要な状態および推移を、任意の端末に着目して生成する方法について述べる。

第4章では、通信サービス競合（非決定性）検出支援方式について述べる。相互作用に含まれる競合（非決定的動作）の検出の問題は、STR手法を用いることによって規則競合の問題として扱うことができる。具体的には、イベントの同じ2規則の組み合わせを比較して、2規則が同時に適用される可能性のある状態を検出し、設計者に提示する方法について述べる。実験結果から、提案した競合検出手法は、機械的に検討対象を絞り込む効果が大きく（POTS+CW+CFV+3WC+CCBSでは1%以下）設計支援に有効であることが確認できた。また、検出結果は、競合する可能性のある2規則の組み合わせを網羅していることを示す。

第5章では、通信サービス設計段階における相互作用設計支援の一環として、相互作用に含まれる意味的矛盾動作検出方式について述べる。提案手法では、矛盾状態知識を用い、サービス合成動作生成時に矛盾状態を判定し、初期状態から矛盾状態に至る状態推移を検出する。また矛盾状態知識の獲得支援についても述べる。提案手法のシステム試作を行い、実際の5つのサービスを対象に実験を行った。その結果より、矛盾状態の検討範囲として12%以下の絞り込み効果があり、検証作業の効率化が図れることを示した。また、初期状態から矛盾状態に至る状態推移を示すことで、意味的矛盾動作の解析が効率的に行えることが確認できた。

第6章では、通信サービス課金仕様競合検出方式について述べる。通信サービスには、着信転送サービスや話中着信サービスのような接続サービスだけでなく、フリーダイヤルサービス（着信側課金サービス）のような課金を主眼とするサービスもある。このような課金を主眼とするサービスが引き起こす課金仕様競合の問題に対しては、課金仕様記述方式と課金仕様競合検出方式を提案した。具体的には、課金仕様に必要な条件として、課金対象端末、課金レートに着目し、それらを規則形式で記述する方式について述べる。次に、この課金規則が同時適用可能なときに生じる課金仕様競合検出方式（課金端末競合検出方式、課金レート競合検出方式）について述べる。実際の6個のサービスを対象に課金仕様を記述し、その課金仕様間で生じる課金仕様競合検出の机上実験結果を示すとともに、課金仕様競合解消の方法を示し、提案した課金仕様記述方式と課金仕様競合検出方式の有効性を示す。

第7章では、本論文の結論として、通信サービス設計および検証における形式的支援に関する提案内容とその特徴を総括し、今後の課題について述べる。

通信サービスの記述実験においてはNTTのPBX操作マニュアルおよびBELLCOREのサービス仕様を参考にし、システム試作ではSymbolics3650 (Lisp) およびSun-spark10 (C) を用いた。

目次

第1章 序論	1
1.1 研究の背景と目的	1
1.2 本論文の概要	6
第1章の参考文献	10
第2章 通信サービス記述方式	13
2.1 第2章の概要	13
2.2 形式記述技法の特徴と限界	14
2.3 通信サービス設計の範囲	17
2.4 STR手法の記述方式	20
2.5 STR手法の適用方式	25
2.5.1 規則適用の基本的な考え方	25
2.5.2 STR手法の適用規約	27
2.6 新規サービス設計	35
2.7 考察	39
2.8 第2章のまとめ	41
第2章の参考文献	42

第3章 通信サービス（相互作用）設計支援方式	44
3.1 第3章の概要	44
3.2 新規サービス設計支援	46
3.2.1 プロトタイピング・アプローチ	46
3.2.2 状態推移動作提示による確認支援	49
3.3 サービス動作（状態推移）生成	51
3.4 通信サービス相互作用設計支援	57
3.4.1 検証項目の分類	57
3.4.2 検証支援方法の概要	59
3.5 考察	63
3.6 第3章のまとめ	66
第3章の参考文献	68
第4章 通信サービス競合（非決定性）検出支援方式	71
4.1 第4章の概要	71
4.2 通信サービス競合（非決定性）	72
4.3 通信サービス競合（非決定性）検出方式の基本概念	75
4.4 競合候補検出	80
4.5 競合検出実験と評価	86
4.6 考察	89
4.7 第4章のまとめ	93
第4章の参考文献	94

第5章 通信サービス相互作用に含まれる意味的矛盾動作

検出支援方式	95
5.1 第5章の概要	95
5.2 意味的矛盾動作	96
5.3 意味的矛盾動作検出方式の概要	98
5.4 矛盾状態知識の設定	100
5.4.1 矛盾状態知識	100
5.4.2 矛盾状態知識の獲得支援	101
5.5 合成状態推移生成	107
5.6 矛盾状態検査	107
5.7 意味的矛盾動作表示	108
5.8 考察	111
5.9 第5章のまとめ	117
第5章の参考文献	119

第 6 章 通信サービス課金仕様競合検出方式	120
6. 1 第 6 章の概要	120
6. 2 課金仕様競合検出の課題	121
6. 3 課金仕様競合	122
6. 4 課金仕様記述方式	124
6. 4. 1 仕様記述要素	124
6. 4. 2 課金規則記述	126
6. 5 課金規則適用	128
6. 6 課金仕様競合検出方式	130
6. 6. 1 課金仕様と接続仕様の関係	130
6. 6. 2 課金仕様競合検出方式の概要	132
6. 6. 3 課金仕様競合検出アルゴリズム	135
6. 7 考察	140
6. 8 第 6 章のまとめ	146
第 6 章の参考文献	147
第 7 章 結論	148
第 7 章の参考文献	153
謝辞	154
研究業績	155

第1章 序論

1.1 研究の背景と目的

近年，電気通信分野での著しい技術革新の進展と通信サービスの普及に伴い，通信サービスに対する利用者の要求が多様化・高度化している．このような需要に対して，迅速な新規サービス開発と提供が必要となり，通信ソフトウェアの効率的かつ迅速な開発技術の確立が早急の課題となっている．

通信ソフトウェアの開発工程を，（１）利用者からみたサービス仕様設計段階と，（２）サービス仕様からプログラム作成段階，の２つに分類したとき，後者に対しては国際電信電話諮問委員会（CCITT）勧告のSDL [1] を代表とする詳細な状態推移表現からプログラムコードの自動生成を行う研究 [2-6] が盛んに進められていると共に，計算機によるソフトウェア開発環境（CASE環境：Computer Aided Software Engineering Environment）の利用検討が盛んになっており，インプリメンテーションの構造を規定した場合，プログラムコードの約60％程度は自動生成でき，実用化が問われる段階に到達していると報告されている [6]．一方，前者に対しては，サービス内容が多様であるため，利用者の視点から通信サービスを記述し，その段階で通信サービス相互作用解消を行うための検証技術が必要であるが，通信サービス仕様をどのような形態で記述するのか，どのような検証項目をどのようなアルゴリズムで検証支援するのか等問題が多く残されており，研究段階である．

通信サービス設計における課題として、代表的なものを以下に示す。

(1) サービス相互作用に含まれる矛盾の解消

新規サービスを既存サービスに追加する際、新規サービス仕様と既存サービス仕様との相互作用（サービスインタラクション）を解析し、サービス仕様間に生じる矛盾を解消し、サービス全体として矛盾のない仕様を作成する必要がある。ところが、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し、その相互作用に隠れたサービス仕様矛盾の検証は、人手で行う事のできる範囲を越えた作業となっている。相互作用解消の問題は、1991年12月、国際電信電話諮問委員会（CCITT）によってインテリジェントネットワークのための技術的勧告として提示されたCS1（IN Capability Set 1）においても、今後の重要な課題のひとつに挙げられている[7-8]。

(2) 修正影響検出

サービス相互作用の解消のために、既存サービス動作規定を修正する必要がある場合、その修正によって影響を受ける動作の検出は、サービス相互作用解消と同様以上に重要な課題である[8]。

(3) 状態推移表現の記述の困難さ

状態推移表現の特徴は、状態数、状態推移数が少ない簡単なモデルでは、記述は容易であり理解しやすいが、状態数、状態推移数が多くなると、とたんに記述が難しくなるとともに理解性も劣化する。通信システムは、状態推移表現に基づいて表現される。状態推移表現は、基本的には、有限状態機械（FSM: Finite State Machine）として扱うことができる。

サービス仕様からプログラムコードを生成する研究においては、詳細な状態推移表現

が正しく記述されていることを前提としている。ところが、通信サービスが高度化、多様化する今日、サービスの状態数、状態推移数は多くなり、個々のサービス記述はともかく、サービス全体としての状態推移表現の記述自体が難しくなっている。サービス相互作用まで想定し、解消済の状態推移表現を記述することは、設計者の能力の限界を超えたものになっている。この問題を解消するためには、より上位の記述から詳細な状態推移表現を生成し、状態推移表現を検証する技術が必要となっている[9]。

(4) 通信サービス相互作用解析のための形式的仕様記述技法の問題点

通信サービス相互作用に含まれる矛盾検出の困難さの主な要因は、設計者がサービスを組み合わせた時に生じるであろう状態および推移を把握・予測することが困難なことに起因する。この点を解決する枠組みの確立が、サービス相互作用設計支援を行う上での鍵となる。

形式的仕様記述方式としては、SDL [1]，Petri Net [10, 11]，LOTOS [12, 13]，等があるが、SDL [1]，Petri Net [10, 11] は、状態推移を明示的に表現するけれども、状態を合成する概念がなく、サービス相互作用の矛盾を検出する機能には限界がある。LOTOS [12, 13] は、サービス動作をイベント系列の集合で規定する手法であるが、状態を明示的に表現できないため、サービス合成における矛盾する状態の解析を行うには自ずと限界がある。

通信サービス設計の効率化を図るためには、形式的記述技法だけではなく、形式的記述技法と設計・検証支援機能を組み合わせた新たな枠組みが必要となっている。

(5) 通信ソフトウェア開発パラダイムの問題点

通信ソフトウェア開発工程の概要を図1-1に示す。通信サービス設計段階では、

サービスの振る舞いを規定する。S D L設計段階では、対象システムに応じて信号等のタスクを付加する。プログラム作成段階では、S D L仕様をプログラムコードに変換する。一般に、検証は下位仕様（またはプログラム）が上位仕様を満足しているかどうかの検証（Verification）と、利用者の視点から開発システムが利用者の要求を満足しているかどうかの要求確認（Validation）の2種類がある。システム開発が終わった段階で、利用者の要求が満足されていない場合や利用者の要求定義もれがある場合は、再度、設計の上流工程に戻り開発・テストを行う必要があり、予定外の開発工数、費用の増加が発生する。Boehmは、ひとつの誤りを要求定義段階で発見する場合と運用段階で発見する場合の修正にかかる費用はすくなくとも前者：後者＝1：100と報告している[14]。このような背景から、近年、ソフトウェア工学においては、設計・開発工程におけるプロトタイピング・アプローチが必要であるという認識に至っている。つまり、通信サービス設計段階では通信サービス仕様を視覚的な動作で表現する枠組みを作成し、要求定義段階で利用者の要求確認（Validation）を行う手法である。通信ソフトウェア開発・検証においても、基本的には上記と同様に、設計の上流工程である通信サービス設計段階においてサービス要求の確認（Validation）ができれば、開発効率が向上する。このような観点から、本研究の位置付けは、設計の上流工程である通信サービス設計・検証支援としている（図1-1の網掛け部分）。

（6）通信サービス仕様の検証技術

従来の通信サービス仕様検証においては、余すところなく記述された仕様を前提とし、仕様の実行手順に沿った動作確認を支援する方法に主眼をおいている[15-18]。仕様の規定を行うためには、あらかじめ、実際に起こり得る全ての場合を洗い出す必要がある。仕様は、それぞれの場合に対してどのような動作をすべきかを規定

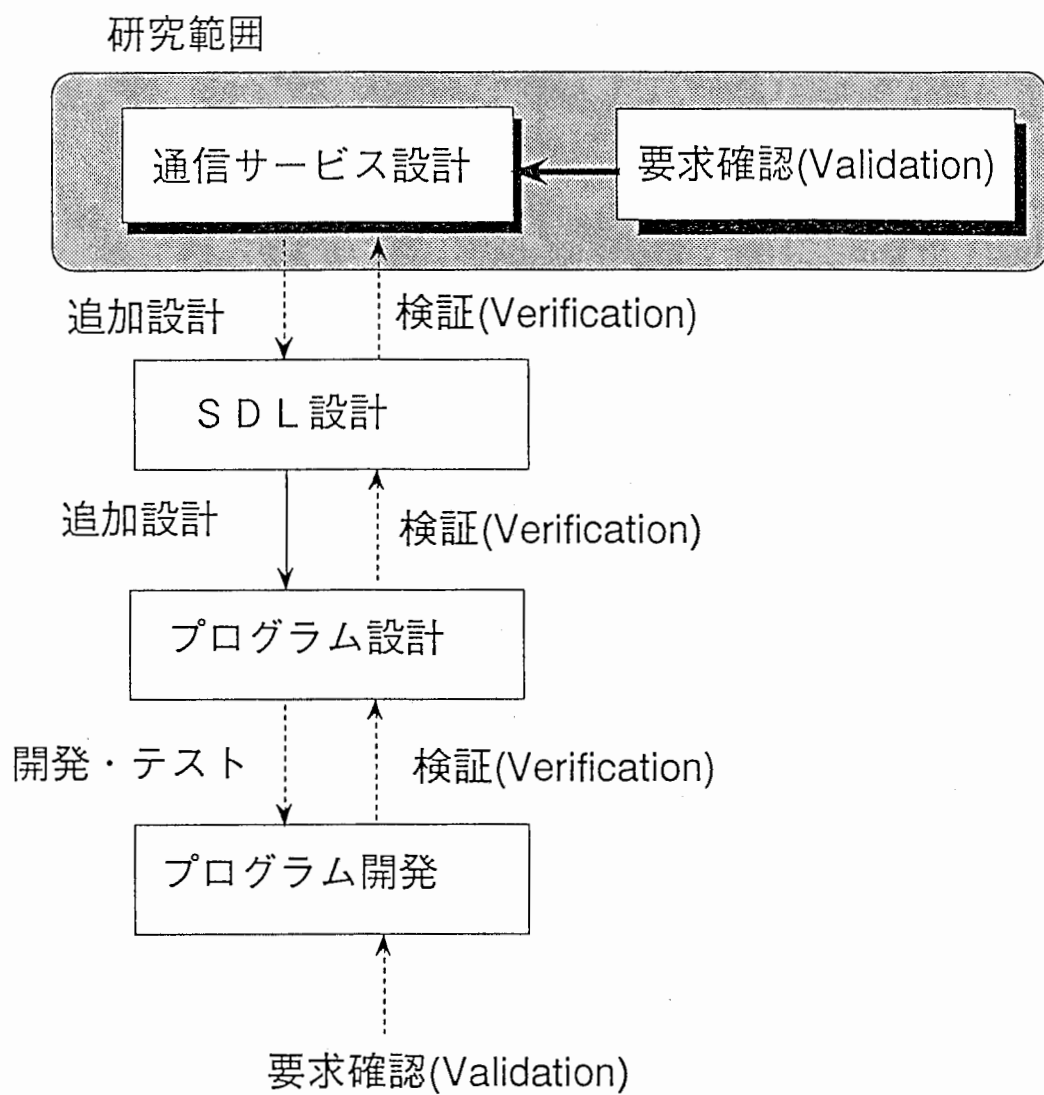


図 1 - 1 開発工程と研究範囲

したものである。検証の結果、誤りを発見した際、再び同様の作業を繰り返す必要があるが、実際に起こり得るすべての場合を予測し、把握している保証はなく、仕様が決定されるまで検証、修正を対象療法的に繰り返す結果になり作業効率が悪い。これは、設計と検証を完全に分離しているために生じる問題である。サービス規定を追加することによりどのような動作が生じるのか、機械支援により実際に起こり得る動作をすべて洗いだし、その結果、どのような矛盾が生じるのかを自動的に検出できる枠組があれば、矛盾に対して効率よく対策を立てることが可能になる。

1. 2 本論文の概要

本論文では、(1) 通信サービス仕様として利用者の観点から見える動作を対象とし、サービス動作を形式的な規則形式により記述する手法 (STR: State Transition Rule) について記述する。また、(2) STR手法を用いた新規サービス記述および設計支援、(3) 新規サービスと既存サービスからサービス合成動作を作成し、そのサービス合成動作を基盤に相互作用に含まれる各種の矛盾検出支援方式、(4) 課金を主眼とするサービスに対しては課金仕様記述方式、および、課金仕様の競合検出方式、について述べる。具体的には、筆者がこれまでに行ってきた、通信サービス設計および検証における形式的支援に関する以下の研究項目について述べる。

- ① 通信サービス記述方式
- ② 通信サービス設計支援方式
- ③ 通信サービス競合 (非決定性) 検出方式
- ④ 通信サービス相互作用に含まれる意味的矛盾動作検出支援方式
- ⑤ 通信サービス課金仕様競合検出方式

第2章：通信サービス記述手法 [文献19, 20]

利用者の視点から通信サービス動作を，形式的な規則形式の集合で記述する S T R (State Transition Rule) 手法を提案した．具体的には，S T R 手法の規則記述と規則適用方法について述べる．S T R 手法を前提とした場合の新規サービス設計時において，1) 既存サービスの規則を再利用することにより特徴的な動作のみ記述すればよく，記述量の削減化が可能になること（例えば，着信転送サービスの記述では40%程度の記述ですむこと），2) P B X サービスでは94%のサービス機能が記述できることが確認できた．

第3章：通信サービス（相互作用）設計支援方式 [文献 21, 22, 30]

本章では，発表文献に基づき，新規サービスの設計支援の手法および相互作用設計支援の手法を述べた．相互作用に含まれる矛盾の検出方式の詳細については，第4章以降に述べる．

（1）新規サービス設計支援：

1) 設計者が視覚的な動作確認画面を用いて，サービスの動作に従ってイベントを入力することにより，画面に表示される適用規則と状態の推移（動作）を確認していくプロトタイピング・アプローチを示し，また，2) 初期状態から適用可能な規則を順次適用させて生成した動作を状態推移表の形式で設計者に提示し，網羅的な確認を促す方法を示す．両者を組み合わせて用いることにより，新規サービスの要求確認が容易に実現できた．

（2）相互作用設計支援：

1) 検証項目の分類，2) 検証項目の説明と検出方法の考え方を示す．また，規則修正時の影響波及範囲の検出も可能になることを示す．

新規サービス設計支援および相互作用設計支援に共通に必要な状態および推移を、任意の端末に着目して生成する方法を示す。提案手法により、相互作用の検討範囲、修正影響範囲の絞り込みが可能になり、サービス設計の効率的な支援が可能となる。

第4章：通信サービス競合検出支援方式 [文献 23-25]

相互作用に含まれる非決定的動作矛盾の検出の問題は、STR手法を用いることによって規則競合の問題として扱うことができる。本章では、イベントの同じ2規則の組み合わせを比較して、2規則が同時に適用される可能性のある状態を検出し、設計者に提示する方法について述べる。実験結果から、提案した競合検出手法は、機械的に検討対象を絞り込む効果が大きく（POTS+CW+CFV+3WC+CCBSでは1%以下）設計支援に有効であることが確認できた。また、検出結果は、競合する可能性のある2規則の組み合わせを網羅していることを示す。

第5章：通信サービス相互作用に含まれる意味的矛盾動作検出方式 [文献 26]

通信サービス設計段階における相互作用設計支援の一環として、相互作用に含まれる意味的矛盾動作検出方式について述べる。提案手法のシステム試作を行い、実際の5つのサービスを対象に実験を行った。その結果より、矛盾状態の検討範囲として12%以下の絞り込み効果があり、検証作業の効率化が図れることを示した。また、初期状態から矛盾状態に至る状態推移を示すことで、意味的矛盾動作の解析が効率的に行えることが確認できた。

第 6 章：通信サービス課金仕様競合検出方式 [文献 27]

フリーダイヤルサービスのような課金を主眼とするサービスが引き起こす課金仕様競合の問題に対しては、課金仕様記述方式と課金仕様検出方式を提案した。具体的には、課金仕様に必要な条件として、課金対象端末、課金レートに着目し、それらを規則形式で記述する方式について述べる。次に、この課金規則が同時適用可能なときに生じる課金仕様競合検出方式（課金端末競合検出方式、課金レート競合検出方式）について述べる。実際の 6 個のサービスを対象に課金仕様を記述し、その課金仕様間で生じる課金仕様競合検出の机上実験結果を示すとともに、課金仕様競合解消の方法を示し、提案手法した課金仕様記述方式と課金競合検出方式の有効性を示す。

第 7 章は、結論であり、各章の結果を総括する。

通信サービスの記述実験においては N T T の P B X 操作マニュアル [28] および B E L L C O R E のサービス仕様 [29] を参考にし、システム試作では Symbolics3650 (Lisp) および Sun-spark10 (C) を用いた。また、本論文中の参考文献に用いる雑誌の略式名称と正式名称の対応表を表 1 - 1 に示す。

表 1 - 1 雑誌の略式名称と正式名称の対応表

雑誌種別	略式名称	正式名称
論文誌	信学論文誌	電子情報通信学会論文誌
	情処論文誌	情報処理学会論文誌
学会誌	信学会誌	電子情報通信学会誌
	情報処理	情報処理
研究会	信学技報	電子情報通信学会技術研究報告
	情処研報	情報処理学会研究報告
全国大会	〇〇〇〇信学 春／秋季大会	〇〇〇〇年（西暦）電子情報通信 学会春／秋季全国大会講演論文集
	情処第〇〇回 大会	情報処理学会第〇〇回（元号〇〇年前／後期） 全国大会講演論文集

第1章の参考文献

- [1] CCITT Recommendations Z. 100 to Z. 104 (1985).
- [2] Jackson, L. N., Cheng, K. E., Choong, T. S., Pascoe, R. S. V., Cain G. J. :
MELBA At The Age of Eight: An Automatic Code Generation System,
SDL'87: State of the Art and Future Trends, Saracco, R. and Tilanus, P. A. J (eds.),
pp. 371-381 (1987).
- [3] Vefsnmo, E. : DASOM - A SDL-TOOL, SDL'87: State of the Art and Future
Trends, Saracco, R. and Tilanus, P. A. J (eds.), pp. 35-42 (1987).
- [4] 田中功一, 佐藤文明, 水野忠則 : プログラム部品 利用による通信ソフトウェア開
発支援, 情処研報, Vol. 89, No. 41, pp. 17-24 (1989).
- [5] 伊藤正樹, 市川晴久 : 通信分野における自動プログラミング, 情報処理,
Vol. 28, No. 10, pp. 1405-1411 (1987).
- [6] 市川晴久 : 通信ソフトウェアを対象としたCASEの現状と動向, 情報処理,
Vol. 31, No. 8, pp. 1049-1056 (1990).
- [7] Duran, J. M and Visser, J. : International Standards for Intelligent Networks,
IEEE Communication Magazine, Vol. 30, No. 2, pp. 34-42 (1992).
- [8] Collet P. and Kung R. : The Intelligent Network in France,
IEEE Communication Magazine, Vol. 30, No. 2, pp. 82-89 (1992).
- [9] 小山田正史, 岩戸伝一 : ソフトウェア動作仕様の階層的設計支援技法とその支援
系, 情処研報, Vol. 90, No. 10, PP. 1-8 (1990).
- [10] Peterson, J. L. : Petri Net Theory and Modeling of Systems, Prentice-Hall (1981).
- [11] J. L. ピータースン著, 市川惇信/小林重信訳 : ペトリネット入門 ;
情報システムのモデル化, 共立出版 (1984).

- [12] ISO8807: Information processing systems - open systems interconnection - LOTOS - a formal description technique based on the temporal ordering of observational behavior, ISO publication (1988).
- [13]高橋薫, 神長裕明, 白鳥則郎: LOTOS 言語の特質と処理系の現状と動向, 情報処理, Vol. 31, No. 1, pp. 35-46 (1990).
- [14]Boehm, W. B. : Guidelines for Verifying and Validating Software Requirements and Design Specifications, EURO IFIP 79, pp. 711-719 (1979).
- [15]Wakahara, Y. and Kakuda, Y. : Verification of Requirements Specifications for Telecommunications Software by the Investigation of Specifications Execution Sequences, GLOBECOME'87, pp. 661-666 (1987).
- [16]Wakahara, Y. and Ito, A. : Prototyping System for Telecommunications Software based on Abstract Execution of Requirements Specifications, Computer Networks and ISDN Systems, Vol. 12, No. 2, pp. 119-128 (1987).
- [17]水野修, 新津善弘: メッセージシーケンス図入力による通信サービス仕様設計方式, 信学論文誌, Vol. J74-B-I, No. 6, pp. 1042-1055 (1991).
- [18]長谷川晴朗: ペトリネットを用いた通信サービス仕様設計支援, 信学論文誌, Vol. J74-B-I, No. 6, pp. 445-455 (1991).
- [19]原田良雄, 平川豊, 竹中豊文, 門田充弘: サービス仕様の自動生成に関する考察, 情処第39回大会, 5S-5 (1989).
- [20] Hirakawa, Y., Harada, Y. and Takenaka, T. : 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 (1990).

- [21]原田良雄, 平川豊, 竹中豊文: 通信サービス追加設計支援方法の一考察,
信学技法, Vol. 90, No. 454, pp. 13-18 (1991).
- [22] Harada, Y., Hirakawa, Y. and Takenaka, T. :A Design Support Method for
Telecommunication Service Interactions, GLOBECOM'91, pp. 1661-1666
(1991).
- [23]原田良雄, 平川豊, 竹中豊文: 通信サービス動作競合検出方法,
情処研報, Vol. 91 No. 83, pp. 87-92 (1991).
- [24]原田良雄, 平川豊, 竹中豊文: 通信サービス競合検出システムの試作と評価,
信学技法, Vol. 92, No. 75, pp. 13-18 (1992).
- [25] Harada Y., Hirakawa Y., Takenaka T., Terashima N.: A Conflict Detection
Support Method for Telecommunication Service Descriptions,
信学論文誌, Vol. E75-B, No. 10, pp. 986-997 (1992).
- [26]原田良雄, 高見一正, 太田理, 寺島信義: 通信サービス相互作用に含まれる意味
的矛盾動作検出方式, 情処論文誌, Vol. 35, No. 8, pp. 1602-1613 (1994).
- [27]原田良雄, 高見一正, 太田理, 寺島信義: 通信サービス課金仕様競合検出方式,
情処論文誌, Vol. 34, No. 5, pp. 1064-1073 (1993).
- [28]NTTビジネスホンE操作マニュアル.
- [29]Bell Communication Research: Technical Reference, Issue 2 (1987).
- [30] Takami, K., Harada, Y., Ohta, T. Terashima, N. : A Visual Design System for
Telecommunication Services, Proc of IPCCC, pp. 593-599 (1993).

第2章 通信サービス記述方式

2. 1 第2章の概要

通信ソフトウェア設計の上流工程である通信サービス設計段階において、機械的な設計・検証支援を行うためには、外部仕様として、通信サービスを利用者の視点から通信サービスを形式的に記述する手法が必要であるとともに、その記述手法と組み合わせた設計・検証支援アルゴリズムが必要である。

状態推移モデルに基づく通信ソフトウェア仕様を記述するための形式記述技法（FDT：Formal Description Technique）として、SDL（Specification Description Language）[1,2]，Estelle（Extended State Transition Model）[3]，LOTOS（Language Of Temporal Ordering Specification）[4,5]，Petri Net[6,7]が提案されている。従来は、これらの記述技法を用いて、設計者が通信ソフトウェア仕様を記述し、仕様検証を行っている。状態推移モデルの特徴としては、状態数および状態推移数が少ない場合は、理解性がよく、検証も容易であるが、状態数および状態推移数が多くなるにつれて、理解しにくくなり、検証も難しくなる。一般に、通信ソフトウェアは大規模、複雑、かつ分散処理という特徴がある。大規模、複雑な通信ソフトウェア設計上の問題として、サービスの追加あるいは修正を行うとき、既存仕様のどの部分に追加するのか、あるいは、どの部分を修正すべきなのかを決定することは難しい作業となっている。特に、近年、新規サービスを追加するときに既存サービスとの間で生じる相互作用（サービスインターラクション）に含まれる矛盾を効率よく解析する技術の確立が望まれている[8-11]。

従来、議論の中心になっている通信ソフトウェア仕様は、利用者の視点から見たサービスの振る舞いと、実現システムのシステム構成やプロトコルに応じたシステム

内部の振る舞いの両方を記述したプログラム仕様レベルである。事務系のソフトウェア設計手法では、利用者の要求を外部仕様とし、外部仕様の実現方法を実現システムに応じて規定した仕様を内部仕様として区別している。このように区分することで、外部仕様レベルでの要求定義を明確にした上で（利用者の要求誤りや設計者の誤解を解消した上で）次工程に進むことにより、要求定義誤りが開発の下流工程に及ぼす影響（仕様修正、プログラム修正）を防ぐことが可能になり、結果として、開発工数の削減化が可能になる。通信ソフトウェア設計においても、外部仕様と内部仕様の区分を行い、外部仕様レベルでの通信サービスの振る舞いを定義し、次工程に進む前に、サービスの本質的な議論を可能とする環境が必要である。

本章では、利用者の視点から（システムをブラックボックスとみなす）、通信サービス動作を形式的な規則形式の集合で記述する手法について述べる。この記述手法をSTR（State Transition Rule）手法[12,13]と呼ぶ。STR手法を用いた新規サービスの設計では、既存規則の再利用による記述量の削減化が可能になることを示す。また、PBXサービス[14]を対象にした記述実験の結果、記述の網羅性は94%であったことを示す。設計及び検証支援については、3章以降に記述する。

2. 2 形式記述技法の特徴と限界

形式記述技法の特徴の概要と限界について述べる。

SDL[1]の特徴と限界：

SDLは、通信システムの機能に関する仕様や動作を曖昧なく簡潔に記述するため、CCITT（国際電信電話諮問委員会）で開発し勧告として標準化した形式記述技法である。SDLの特徴と処理系の現状と動向は文献[2]に詳しい。記述の特徴は、①シス

テムを構成する機能単位であるブロックやプロセスの間の静的な信号送受関係によりシステムの枠組みと構造を表現する静的構造記述と、②各プロセスの動的な動きを表す動作記述とから構成する。SDLのシンタックスには、図形式のSDL/GR (Graphical Representation) とテキスト形式のSDL/PR (Phrase Representation) とがある。両者は基本的に等価であるが、一般的にGRによる記述は理解しやすいといわれており、SDL支援システムの多くがGRをサポートしている。静的構造記述に使われる要素（システム、ブロック、プロセス、等）、静的構造記述とプロセス動作記述の両者に使われる要素（テキスト枠、コメント、等）、プロセス動作記述に使われる要素（開始、停止、状態/次状態、信号入力、等）、の表現形式は標準化されている。しかしながら、サービス記述と検証は設計者に委ねており、SDLとしては明確にサービス記述・検証を支援する機能はない。SDL形式の記述を前提としたサービス設計・検証支援研究が行われているが[15-17]、状態を組み合わせて新たに状態推移を生成するメカニズムはなく、サービス追加にともなう既存サービスとの相互作用解析の支援には限界がある。

Estelle[3]の特徴と限界：

Estelle は I S O (International Organization for Standardization) により、Pascalをベースにした構文を用い通信システムを拡張有限状態機械として記述することを目的に設計された言語である。Pascalと拡張有限状態機械という一般的な道具立てを用いているので多くの技術者にとって比較的容易に理解しやすいのが最大の利点である。Estelle言語の特質と処理系の現状と動向は文献[3]に詳しい。Estelleは、副作用を避けるため、サービスに規定している状態以外の状態生成を禁止している。追加サービスと既存サービスの相互作用を解析するためには、サービス合成により生じる状態と状態推移が必要であるため、Estelleでは相互作用解析の支援には限界がある。

LOTOS[4]の特徴と限界：

LOTOS はISO において OSI (Open Systems Interconnection) のプロトコルを規定するために開発された外部挙動規定法によるものである。LOTOS では、時間順序の概念のもとで、外部との観察点でのイベントの発生順序にしたがって、通信システムの仕様を記述する。通信システムを実現するプロセス（通信プログラム）に対して、環境と通信するブラックボックスとみなすことができる。プロセスは、インタラクションの元になるイベントの列によって規定し、データに関してはSDLと同様に抽象データ型を採用している。LOTOS 言語の特質と処理系の現状と動向は文献[5]に詳しい。LOTOS を用いた通信サービス仕様は、任意状態、イベント系列の集合、目標状態の組の集合として規定する。任意状態から目標状態へ至る中間の状態は陽に表現しない。サービス合成における、相互作用の解析はイベント系列の発火タイミングでしか分からず、相互作用における非決定的動作の検出報告[18]があるが、他の矛盾検出については研究段階である。状態生成を陽に行っていないので、状態に関係する矛盾検出については、自ずと限界がある。

Petri Net[6]（ペトリネット）の特徴と限界：

ペトリネットは、非同期・平行系システムのモデル化および解析をするための手法であり、現在各方面で幅広く使用されている。構成は極めて簡単で、2種類の接合集合（プレースとトランジション）とそれらの間の有効枝（アーク）からなるグラフである。一般に、プレースは事象生起に関する条件を示し、トランジションは離散事象の生起と完了を表現する。そして、プレースに対応する条件が現時点で成立している場合には、そのプレース中にトークンを置き、トランジションの発火によるトークンの移動によりシステムの状態推移を表す。ペトリネットは、文献[6-7]に詳しい。ペトリネットを用いた通信サービス設計支援が研究されているが[16]、SDL、

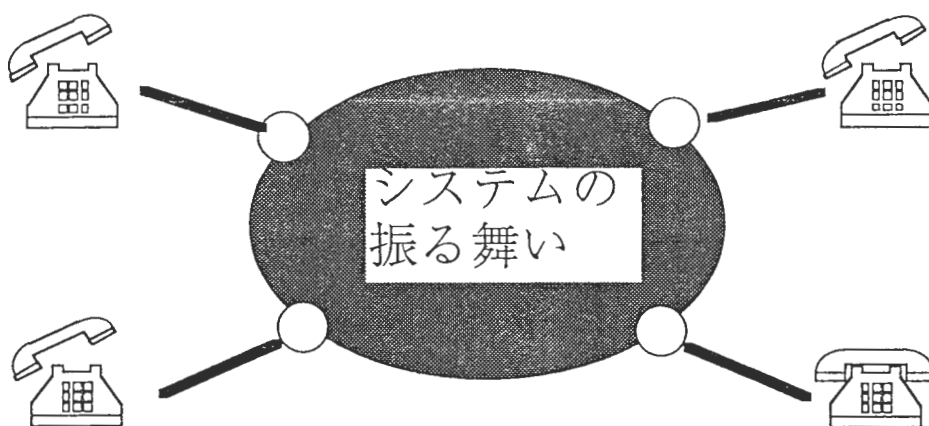
Estelleと同様に、複数サービスを合成する場合に、個別サービスの仕様を適用して、合成状態を生成する概念はなく、サービス合成の検証範囲が狭い点に限界がある。

また、上記の形式記述技法では、設計者が通信サービス仕様あるいはシステム仕様を記述することを前提としている。これに対して、STR手法ではサービスの特徴的動作を規則集合で記述し、他の基本的な動作は、ベースとなるサービス（通常、基本電話サービス）の仕様を再利用することで、仕様生成を行うことが可能となる。一般に、サービス仕様を決定するまでには、何度かの仕様の推敲が必要になるが、書き直し作業の軽減化が図れるのも、STR手法の特徴である。

2. 3 通信サービス設計の範囲

本節では、通信サービスの設計対象とする範囲について述べる。利用者の視点から、外部仕様として、通信サービス動作を規定する。システム（ネットワーク）の振る舞いはブラックボックスとして扱う（図2-1）。図2-2および図2-3にサービス動作の例を示す。図2-2は、端末Aがアイドル状態のとき、受話器を挙げると、ダイヤル可能音が聞こえる状態（ダイヤル可能状態）に推移する動作を示す。図2-3は、端末Aがダイヤル可能状態、かつ、端末Aがアイドル状態のとき、端末Aが端末Bにダイヤルを行うと、端末AからBへの呼出中の状態（端末Aには呼戻音が出ており、端末Bには呼出音が出ている状態）に推移する動作を示す。

上記の外部仕様レベルでの、通信サービス記述および、サービス設計・検証支援を研究対象としている。



(注) システムの振る舞いを
ブラックボックスとして扱う

図 2 - 1 通信サービス検討範囲

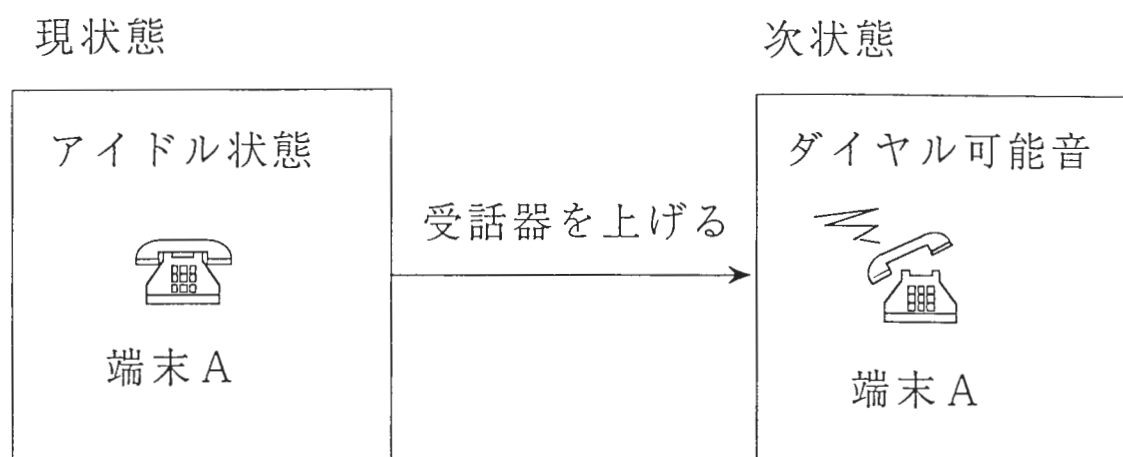


図 2 - 2 通信サービス動作例 1

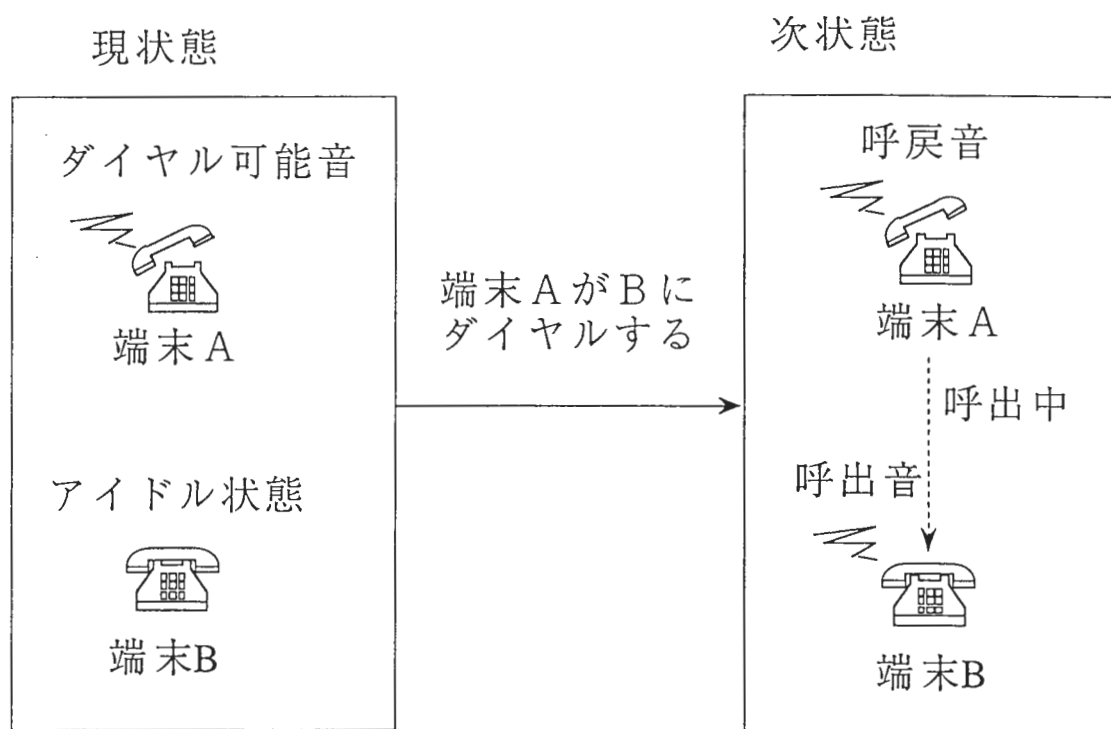


図 2 - 3 通信サービス動作例 2

2. 4 STR手法の記述方式

状態推移を規定する方法として、1) 規則記述、2) 禁止状態集合記述、3) タイマー記述を用いる。サービスの特徴的な動作は、それらの集合で表す。また、STR手法のBNFを図2-4に示す。上記3つの記述方式について述べる前に、前提となる、状態記述要素、端末状態、およびイベントの表記方式について述べる。

[定義2. 1] (状態記述要素)

状態記述要素とは、ある端末の状態を規定するための最小の規定要素をいう。状態記述要素 (BNF-7; <Primitive>) は、状態識別名 (BNF-6; <Name-string>) と左括弧に続く1つ以上の端末識別子 (端末変数) (BNF-10; <Var>) から構成される。複数の端末変数を記述する際には、間にカンマを記述する。最後は右括弧で閉じる。括弧内の1番目の端末変数は、状態記述要素を保持する端末を示し、2番目以降は1番目の端末と関連する相手の端末を示す。 □

[定義2. 2] (端末状態)

(1) ある端末の状態は、状態記述要素の1番目の端末変数が同じ複数の状態記述要素の集合を用いて規定する。

(2) 複数の端末の状態は、状態記述要素の1番目の端末変数が違う複数状態記述要素の集合を用いて規定する。

(BNFの表記上は、いずれも状態記述要素の集合として扱う (BNF-9; <Primitive-string>)) 。 □

```

1  <STR-description>
   ::= <Rule>
      <Inhibitted-primitive-set>
      <Timer>

2  <Rule> ::= <IS> <Event> ":" <NS> "."

3  <IS> ::= <Primitive-string>
           | "not[" <Primitive-string> "]" { " , " <Primitive-string> }
           | "cond:" <Primitive> { " , " <Primitive-string> }

4  <Event> ::= <Name-string> "(" <Var> { " , " <Var> } ")"
              | "timeover(" <Primitive-string> ")"
              | "[" <Primitive-string> "]"

5  <NS> ::= <Primitive-string>
           | "empty"

6  <Name-string> ::= (<Number> | <Char>) <Ch-num-string>

7  <Primitive> ::= <Name-string> "(" <Var> { " , " <Var> } ")"

8  <Primitive-string> ::= <Primitive> { " , " <Primitive-string> }

9  <Ch-num-string> ::= (<Number> | <Char> | "A" | "B" | ... | "Z") <Ch-num-string>

10 <Var> ::= "A" | "B" | "C" | ... | "Z"

11 <Number> ::= "0" | ... | "9"

12 <Char> ::= "a" | "b" | ... | "z" | "-"

13 <Inhibitted-primitive-set> ::= "inhibitted primitive set:" <Inhibitted-string> "."

14 <Inhibitted-string> ::= "{" <Primitive-string> }" "(" <Name-string> }"
                        | "{" <Primitive-string> }" "(" " " ")"

15 <Timer> ::= "timer:" <Primitive-string> <Number> "sec."

```

図 2 - 4 STR手法のBNF表現

[定義 2. 3] (イベント)

状態推移を引き起こす契機（トリガー）となる事柄をイベントと呼ぶ。STRの表記上、以下の3種類の表記方式を用いる（BNF-4；<Event>）。

(1) 利用者が行うイベント操作の表記：英数字のシンボルを用いて表現されたイベント名、左括弧、イベントが起動された端末変数（必要ならばカンマの後にイベントの相手端末変数を記述する）を記述した後、右括弧で閉じる。

(2) タイムオーバーによるイベントの表記："timeover"の後に左括弧、タイマーの監視状態集合（タイマー記述を参照）を記述し、右括弧で閉じる。

(3) 疑似イベント：ある状態になることにより起動がかかる場合、左鍵括弧（[]），監視状態集合の後に右鍵括弧（]）で閉じる。 □

[定義 2. 4] (規則記述)

Rule = 規則， I S = 規則の始状態， Event = 規則のイベント， N S = 規則の次状態，とする。規則 Rule は以下のように「始状態」「イベント」「次状態」の三つ組で表す（BNF-2；<Rule>）。

I S Event : N S .

規則の始状態 = I S は、状態記述要素の集合で規定する。ただし、状態記述要素の否定条件（状態記述要素の集合が存在しない）の表記方法は、not [状態記述要素の集合] と表記する。条件判定のみに用いて書き換えの対象としない状態記述要素に対しては cond: を状態記述要素の前に記述する（BNF-3；<IS>）。規則の次状態 = N S は、状態記述要素の集合で規定するが、規則上、始状態の状態記述要素を取り去る場合のみ、明示的に「empty」と記述する（BNF-5；<NS>）。 □

規則記述例：

規則を説明上区別するために、規則の識別子（例えば，pots-1）を便宜上付加す

る.

pots-1) idle(A) offhook(A): dial-tone(A).

pots-2) dial-tone(A), idle(B) dial(A, B): ringback(A, B), ringing(B, A).

pots-1)は、端末Aがアイドル状態「idle(A)」のとき、オフフックする「offhook(A)」とダイヤル可能状態「dial-tone(A)」になる動作を規定する。pots-2)は端末Aがダイヤル可能状態「dial-tone(A)」かつ端末Bがアイドル状態「idle(B)」のとき、端末Aが端末Bにダイヤルをする「dial(A,B)」と端末Aが端末Bを呼び出している状態（端末Aには呼戻音「ringback(A,B)」，端末Bには呼出音「ringing(B,A)」が表示されている状態）に推移する動作を規定している。 [規則記述例終]

[定義 2. 5] (禁止状態集合記述)

禁止状態集合記述は、規則が適用された結果の次状態が、あらかじめ与えておいた禁止状態集合を含む場合、規則適用を行わず、イベント生起端末の状態を、禁止状態集合記述の指定した状態記述要素に基づいて書き換える機能である。記述の方法を以下に示す (BNF-13, 14参照)。

inhibitted primitive set:

{<Primitive-string>} (<Name-string>)

<Primitive-string> : 状態記述要素の集合を表す。

<Name-string> : 状態識別子を表す。

|| で括り指定した状態集合が、禁止状態集合を表す。この部分を禁止状態集合記述の条件部 (I F 部) と呼ぶ。後ろの () で指定した状態識別子が、書き換えに用いる状態記述要素である。この部分を禁止状態集合記述の動作部 (T H E N 部) と呼ぶ。状態識別子を指定する場合と、省略する場合がある。

□

禁止状態集合記述例：

inhibitted primitive set:

{m-tcs(A, B), ringing(A, B), ringback(B, A)} (busy)

これは、端末Aが端末Bからの着信を拒否するためにスクリーニングリストへ登録していた場合（m-tcs(A, B)），端末Bからの呼びだし（ringing(A,B), ringback(B, A)）は拒否し、イベント生起端末をビジーにすることを表している。

[禁止状態集合記述例終]

[定義 2. 6] （タイマー記述）

状態記述要素または状態記述要素の集合に対して、指定した時間の監視を行い、指定した時間が経過したとき、タイマ処理のためのイベントを起動させる機能である。記述は、以下のように監視状態集合と監視時間を指定する（BNF- 1 5 参照）。

timer: <Primitive-string> <Number>"sec."

<Primitive-string>：状態記述要素の集合（監視状態集合）を表す。

<Number>：監視する時間（秒）を表す。



タイマー記述例：

timer: dial-tone(A) 5sec.

端末変数Aに対応する端末において、ダイヤル可能状態「dial-tone(A)」で5秒間ダイヤルを行わなかった場合、ハウラー音に推移する。タイマー記述により、イベント「timer(dial-tone(A))」が生起される。推移規則を以下のように記述しておけば、ダイヤル可能状態からハウラー音に推移する動作が可能となる。

dial-tone(A) timer(dial-tone(A)): howler(A).

[タイマー記述例終]

2. 5 STR手法の適用方式

2. 5. 1 規則適用の基本的な考え方

論理型言語 Prolog においては、知識や規則に変数が含まれている場合には、一致させるための変数の書き換え操作が必要になる。このような同一化の操作をユニフィケーションと呼んでいる [20]。その特徴としては、1) 変数の書き換え処理、2) デレファレンス（変数セルには直接値が格納されている場合とほかの変数セルのアドレスが格納されている場合があり、ユニファイを繰り返すことにより参照ポインタは2段以上になりえる。参照ポインタをたぐる操作をデレファレンスという）、3) 変数への単一代入則（関数的プログラミング [21] と同様な、ユニフィケーションにおいて一致したとみなされた引数の関係は変えないこと）、4) 構造体データの生成／分解といった特徴がある。（詳細は文献 [20] を参照）

STR規則では、Prolog のような複雑な変数の同一化操作は行わず、ひとつの規則の条件部が満足されれば、基本的には規則は適用可能と判定される。変数の扱いは、図 2-5 に示す端末変数の決定順に従い、順次、決定した変数に値を設定しつつ、全ての変数が決定できるかどうかの操作を行う。当然、「変数への単一代入則」は守られる。複数の規則が適用可能な場合は、規則の優先適用戦略を用いて適用規則を選択する（定義 2. 8 参照）。

規則適用を行う概念的な枠組みは、プロダクションシステム [22] と同様である。プロダクションシステムは、プロダクションメモリ（規則集合が格納）、ワーキングメモリ（規則により参照・更新対象のデータが格納）、インタプリタ（規則とデータの両者を参照、実行可能な規則を見い出し、その中のひとつを実行に移す）から構成され、規則競合解消戦略を持つ。プロダクションメモリは、STR手法で規定した仕様の集合に対応し、ワーキングメモリは規則を適用して行くときに生成される状態／推移情報に対応し、インタプリタはSTRの適用規定に対応している。プロダクションシ

システムでは競合解消戦略には様々なものがあるが、STRではより具体的に条件を規定し、かつ他の規則の条件を包含している規則を優先する戦略を採用している（定義2.8参照）。

以下では、規則の条件部の変数の同一化の操作について述べる。また、規則の始状態とイベントの記述から端末変数の辿り方を図2-5に示す。

操作1：規則のイベント名と実際に生じるイベント名が等しいとき、端末変数に実端末名を代入し、イベントが等しいとみなす。

操作2：実端末名の端末状態を探索し、規則の始状態にある同一端末（第1引数が同じ変数）の状態記述要素の有無を調べる。無い場合は、規則の条件を満たさないの
で規則は適用されない。探索した状態記述要素の第2引数以降に実際の端末名を代入し、全ての端末変数の関係が一致するまで、操作2を繰り返す。

ここで、以下のような状況を想定してみよう。

端末P, Q, Rの状態：

端末P： SE1(P) 端末Q： SE2(Q, R) 端末R： SE3(R)

このとき、端末PからQにイベント＝「Event(P, Q)」が発生した。

上記の状況では、図2-5のイベントと比較するとA＝P，B＝Qと代入することにより、イベントは等しいとみなすことができる（操作1）。端末Pの状態を探索し、「SE1(P)」の存在を確認する。一方、端末Qに対して同様に状態を調べる。

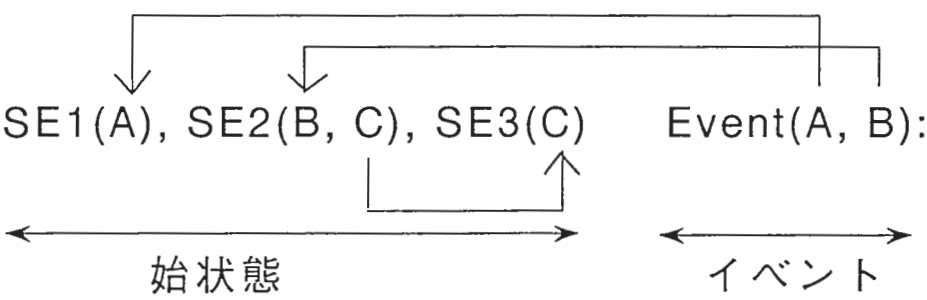


図2-5 端末変数の辿り方

「SE2(Q, R)」と「SE2(Q, C)」は端末変数C=Rと代入することで等しくなることが分かる（操作2）．さらに，端末Rに「SE3(R)」が存在していることを確認する（操作2）．この手順により，規則のIF部（始状態，イベント）が満足される．このような操作を行うことにより，規則の適用可能性を調べる．規則の記述では，イベントの端末変数から始状態に記述された端末変数が辿れるように記述することが前提である．詳細なSTR手法の適用規約は，次節に記述する．

2. 5. 2 STR手法の適用規約

規則適用は以下の2つの規約に従って行なう．

[定義2. 7] 規約1（規則の基本適用）

$$(G \supseteq I S) \wedge (G E = E) \rightarrow R$$

G： 任意の端末数から構成される状態，

GE： Gの状態で生起するイベント，

R： 規則，

IS：規則の始状態，

E： 規則のイベント，

NS：規則の次状態，とする．

（注）→の左側は規則適用の条件，→の右側は適用規則を示す．

Gの状態イベントGEが生起したとき，条件「 $(G \supseteq I S) \wedge (G E = E)$ 」を満足したとき，規則「R」は適用可能であり，始状態ISに該当するGの状態記述要素を次状態NSの状態記述要素に書き換える． □

[定義 2. 8] 規約 2 (規則の優先適用)

$$(I S_i \subset I S_j \rightarrow R_j) \vee (I S_i \supset I S_j \rightarrow R_i)$$

R_i, R_j : G において適用可能な規則, (G は定義 2. 7を参照)

$I S_i$: R_i の始状態,

$I S_j$: R_j の始状態, とする.

(注) $\rightarrow$ の左側は規則適用の条件, $\rightarrow$ の右側は適用規則を示す.

規約 1により適用可能な規則が複数あるとき, 規則の始状態の包含関係をみて, 完全に包含している方の規則を包含されている方の規則に優先して適用する. $\square$

事例を用いた規則適用の説明:

[定義 2. 9] (メモリ関連の状態記述要素)

状態記述要素として, 「m-」で始まるものを端末の状態として表面上には現われないメモリ関連の状態記述要素と呼ぶ. 例えば, サービスの契約を表す状態記述要素がある. $\square$

事例を用いた説明中において, 「idle(P)」を初期状態, あるいは, アイドル状態と表現するが, それぞれの違いを以下に定義する.

[定義 2. 10] (初期状態, アイドル状態)

初期状態: 端末 P の状態が「idle(P)」の状態記述要素だけから構成されている状態をいう.

アイドル状態: 端末 P の状態が「idle(P)」の状態記述要素とメモリ関連の状態記述要素(定義 2. 9)と組み合わさっている状態をいう. $\square$

規約 1 (規則の基本適用) :

規則適用規約 1 の例を図 2-6 に示す. 図 2-6 の (A) の状態は端末 P, Q が共に初期状態である. この状態で端末 P が受話器をあげる操作 (イベント: offhook(P)) をしたとき, 規則「pots-1」の端末変数 $A = P$ とみなすことにより, 規則の始状態「idle(A)」は端末 P の状態「idle(P)」と等しく, 規則のイベント「offhook(A)」も端末 P が起こしたイベント「offhook(P)」と等しいことがわかり, 規則適用され, 端末 P の状態はダイヤル可能状態「dial-tone(P)」に推移する (図 2-6 の (B)). 同様に (B) の状態において, 端末 P が端末 Q にダイヤルをおこなう操作 (イベント:

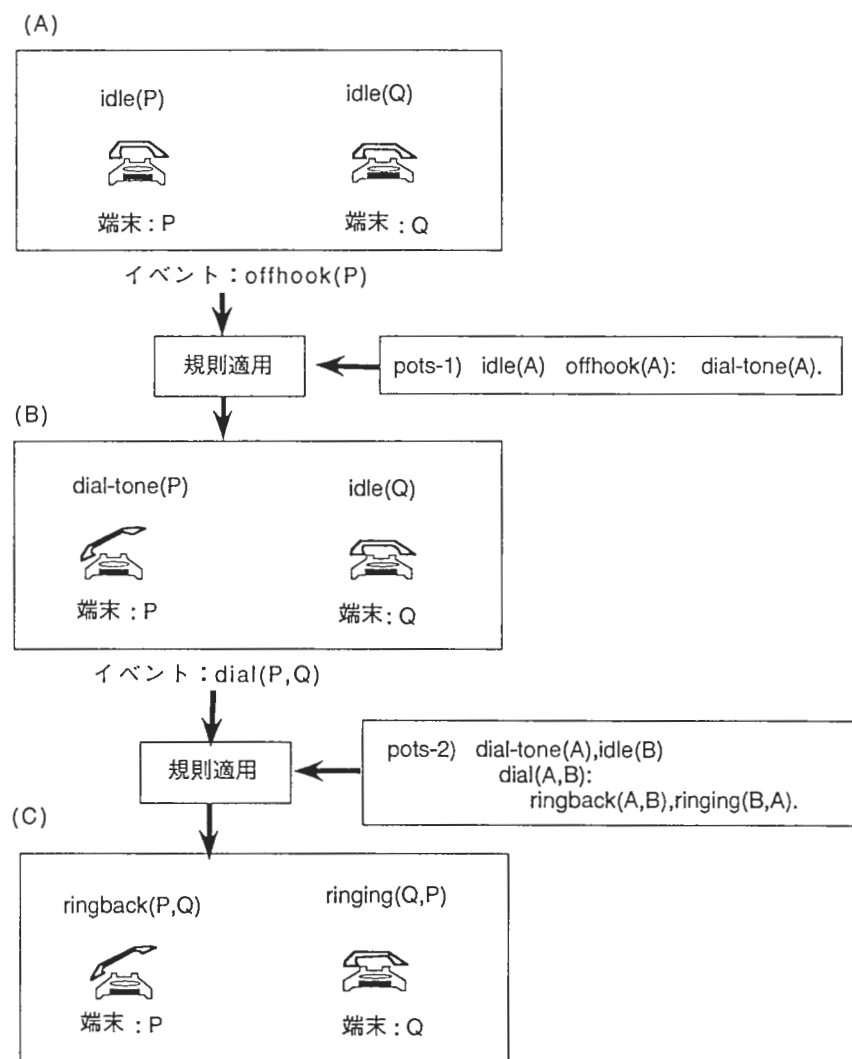


図 2-6 規約 1 (規則の基本適用)

dial(P,Q)) をした場合、規則「pots-2)」の端末変数A = P，B = Qとみなすことにより，規則適用され，端末PからQへの呼び出し状態に推移する（図2-6の（C））。

規約2（規則の優先適用）：

規則適用規約2の例を図2-7に示す。図2-7の（A）の状態は端末Pがダイヤル可能状態「dial-tone(P)」，端末Qがアイドル状態「idle(P)」かつ端末Rに着信転送を登録している状態「m-cfv(Q,R)」，端末Rが初期状態である。このとき端末Pが端末Qに対してダイヤルをおこなう操作をすると，規約1により，2つの規則「cfv-1」，「pots-2」が適用可能であるが，規約2によって始状態を完全に包含している規則「cfv-1」が優先されて，次状態として，端末PからRを呼び出す状態となる（図2-7の（B））。

[事例を用いた説明終]

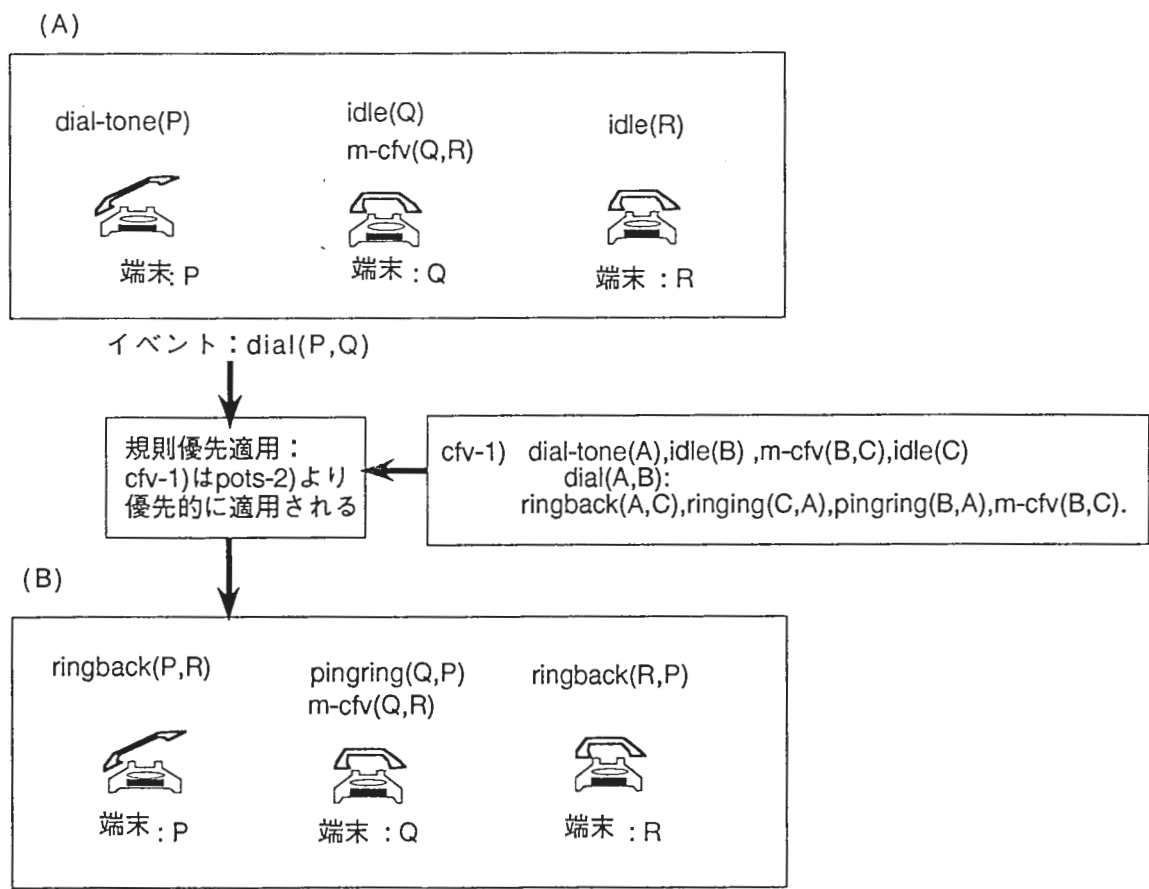


図2-7 規約2（規則の優先適用）

禁止状態集合の適用方法は、ある規則が適用された結果の状態が禁止状態集合を含む場合、状態を指定した場合と指定しない場合によって、イベント生起端末の次状態の扱いが異なる。禁止状態集合の適用方法を以下に示す。

[定義 2. 1 1] (禁止状態集合適用方法)

規則を適用した後の仮の状態 = G_t ，規則適用前のイベント生起端末の状態 = E_{Tc} ，規則を適用した後のイベント生起端末の仮状態 = E_{Tt} ，禁止状態集合の適用結果のイベント生起端末の状態 = E_{Tn} ，禁止状態集合の条件部 = I_C ，禁止状態集合の動作部 = I_A ，イベント生起端末の設定後の禁止状態集合の動作部 = I_A' ，とする。動作部の指定がある場合を規約 1，指定がない場合を規約 2 とする。動作部の指定がない場合は、適用された規則は無視されることになる。それぞれの、適用方法を以下に示す。

$$\text{規約 1)} \quad (G_t \supseteq I_C) \wedge (I_A \neq \emptyset) \rightarrow E_{Tn} = (E_{Tc} \cap E_{Tt}) + I_A'$$

$$\text{規約 2)} \quad (G_t \supseteq I_C) \wedge (I_A = \emptyset) \rightarrow E_{Tn} = E_{Tc}$$

(注) $\rightarrow$ の左側は適用の条件， $\rightarrow$ の右側は適用後の次状態を示す。

動作部の指定がない場合は、状態は変化せず、適用された規則は無視されることになる。 □

事例を用いた説明：

着信拒否サービス (TCS: Terminating Call Screening) と基本電話サービス (POTS: Plain Old Telephone Service) の以下の二つの規定があるときの、禁止状態集合の適用例を示す。

tcs-1) inhibited primitive set: $\{m\text{-tcs}(A, B), \text{ringing}(A, B), \text{ringback}(B, A)\}$ (busy)

pots-2) dial-tone(A), idle(B) dial(A, B): ringback(A, B), ringing(B, A).

ここで、端末Pはダイヤル可能状態、端末Qはアイドルであるが、端末Pからの着信（呼出）は拒否するように指定している以下の状況を想定する。

規則「pots-2」の適用前の状態：

端末P：dial-tone(P)（規則適用前のイベント生起端末の状態；E Tc）

端末Q：dle(Q), m-tcs(Q, P)

この状況で、端末PがQにダイヤルを行うと、規則「pots-2」が適用され、以下の状態となる。

規則「pots-2」の適用後の仮状態（Gt）：

端末P：ringback(P, Q)（適用後の仮状態；E Tt）

端末Q：ringing(Q, P), m-tcs(Q, P)

この状態は、端末変数 $A = Q$ 、 $B = P$ とみなすことで、禁止状態集合に該当することが分かる。イベント生起端末Pの規則適用前の状態（E Tc）「dial-tone(P）」と適用後の仮状態（E Tt）の共通集合は空であり、禁止状態の動作部「busy」にイベント生起端末Pを付加した「busy(P)」が端末Pの次状態となる。

[事例を用いた説明終]

基本電話サービス（POTS）の記述例を、図2-8に示す。また、状態記述要素の説明を表2-1に、イベントの説明を表2-2に示す。

pots-1) idle(A) offhook(A): dial-tone(A).
 pots-2) dial-tone(A), idle(B) dial(A,B): ringback(A, B), ringing(B, A).
 pots-3) ringback(A, B), ringing(B, A) offhook(B): path(A, B), path(B, A).
 pots-4) path(A, B), path(B, A) onhook(A): idle(A), busy(B).
 pots-5) dial-tone(A), not[idle(B)] dial(A,B): busy-dial(A,B).
 pots-6) dial-tone(A) onhook(A): idle(A).
 pots-7) ringback(A, B), ringing(B, A) onhook(A): idle(A), idle(B).
 pots-8) busy(A) onhook(A): idle(A).
 pots-9) busy-dial(A,B) onhook(A): idle(A).
 pots-10) dial-tone(A) timer(dial-tone(A)): howler(A).
 pots-11) busy-dial(A, B) timer(busy-dial(A, B)): howler(A).
 pots-12) busy(A) timer(busy(A)): howler(A).
 pots-13) howler(A) onhook(A): idle(A).
 pots-14) timer: dial-tone(A) 5sec.
 pots-15) timer: busy-dial(A, B) 5sec.
 pots-16) timer: busy(A) 5sec.

図 2 - 8 基本電話サービス記述例

表 2-1 状態記述要素の一覧

状態記述要素	状態記述要素の意味
idle(A)	端末 A がアイドル状態
dial-tone(A)	端末 A にダイヤル可能音がでている状態
busy-dial(A, B)	端末 A にダイヤル後のビジー音が表示されている状態
ringing(B, A)	端末 B に端末 A からの呼出音がでている状態
ringback(A, B)	端末 A に端末 B に対する呼戻音がでている状態
path(A, B)	端末 A が端末 B と通話状態
busy(A)	端末 A にビジー音が表示されている状態
howler(A)	端末 A にハウラー音が表示されている状態

表 2-2 イベントの一覧

イベント	イベントの意味
offhook(A)	端末 A が受話器を挙げる
dial(A, B)	端末 A が端末 B にダイヤルを行う
onhook(A)	端末 A が受話器を降ろす
timer(dial-tone(A))	状態「dial-tone(A)」に対するタイマーイベント
timer(busy(A))	状態「busy(A)」に対するタイマーイベント
timer(busy-dial(A, B))	状態「busy-dial(A, B)」に対するタイマーイベント

2. 6 新規サービス設計

新規サービス設計を行う際には、新規サービスの特徴的な動作（基本電話サービスとの差分動作）を規定し、基本電話サービス規定を再利用することで、新規サービスに必要な動作を補足する方法を用いる。

基本電話サービス（POTS：Plain Old Telephone Service）があるとき（図 2-8）、話中着信サービス（CW：Call Waiting）のサービス仕様記述の例を示す。

話中着信サービス記述例：

話中着信サービスの基本的な仕様は、話中着信サービス加入端末が通話中のとき、他端末から呼び出しを受けることができ（話中着信音が表示される）、話中着信サービス加入端末が応答すると、通話中の相手を保留にし、他端末と通話ができるサービスである。さらに詳細な規定を含めた話中着信サービスの仕様記述例を図 2-9 に示し、それぞれの規則の意味を以下に示す。

cw-1) 端末 A が話中着信サービスに加入しており ($m\text{-}cw(A)$)、端末 B と通話中 ($path(A,B)$) であり、端末 C がダイヤル可能状態のとき、端末 C が端末 A にダイヤルを行うと、話中呼び出し状態（端末 A には話中呼出音 ($cw\text{-}ringing(A, C)$)、端末 C には呼戻音 ($ringback(C,A)$ が表示される状態）となる。

cw-2) 話中呼び出し状態のとき、端末 A がフラッシュ操作（イベント： $flash(A)$ ）により応答すると、割り込み端末と通話中となり、通話相手は保留 ($path\text{-}passive(A,B)$) となる。

cw-3) 通話相手と保留相手はフラッシュ操作で切り替わる。

cw-4) 端末 A が一方の端末と通話中かつ一方の端末を保留しているときオンフック

操作 (onhook(A)) をすると、通話相手とは終話となるが保留端末からの呼び出しがかかる。

cw-5) 保留端末からの呼び出しがかかっているとき、応答 (offhook(A)) を行うと、2 者通話状態となる。

cw-6) 端末 A が一方の端末 C と通話中かつ一方の端末 B を保留しているとき、通話中端末 C がオンフック操作 (onhook(C)) をすると、保留先端末 B との通話状態となる。

cw-7) 話中呼び出し状態のとき、話中呼び出し端末がオンフック操作をするともとの通話状態となる。

cw-8) 話中呼び出し状態のとき、話中着信サービス加入端末がオンフック操作をすると、通話は終了し呼び出し状態となる。

[話中着信サービス記述例終了]

基本電話サービスの規則を再利用することを前提とした、話中着信サービスの状態推移を図 2-10 に示す。

着信転送サービス加入端末がアイドル状態から通話状態に至る動作は、基本電話サービス規定により実現できる。また、着信転送サービス加入端末が通話状態からアイドル状態に至る動作も基本電話サービス規定を利用することで実現できる。一例を挙げると、図 2-10 の状態 1) 「idle(P),m-cw(P)」においてイベント「offhook(P)」が生起した場合、規則「pots-1」は端末変数 $A = P$ とみなすと、規則適用規約 1 (定義 2. 8) により適用可能となり、状態 2) が実現する。

本来は、話中着信サービス加入端末の動作規定は、着信転送サービス加入端末がアイドル状態から通話状態に至る動作、あるいは通話状態からアイドル状態に至る動作

- cw-1) path(A,B),m-cw(A),dial-tone(C)
 dial(C,A):
 path(A,B),m-cw(A),cw-ringing(A,C),ringback(C,A).
- cw-2) path(A,B),m-cw(A),cw-ringing(A,C),ringback(C,A)
 flash(A):
 path-passive(A,B),m-cw(A),path(A,C),path(C,A).
- cw-3) path-passive(A,B),m-cw(A),path(A,C)
 flash(A):
 path(A,B),m-cw(A),path-passive(A,C).
- cw-4) path-passive(A,B),m-cw(A),path(A,C)
 onhook(A):
 path-passive(A,B),ringing(A,B),m-cw(A).
- cw-5) path-passive(A,B),ringing(A,B),m-cw(A)
 offhook(A):
 path(A,B),m-cw(A).
- cw-6) path-passive(A,B),m-cw(A),path(A,C)
 onhook(C):
 path(A,B),m-cw(A).
- cw-7) path(A,B),m-cw(A),cw-ringing(A,C),ringback(C,A)
 onhook(C):
 path(A,B),m-cw(A).
- cw-8) path(A,B),m-cw(A),cw-ringing(A,C),ringback(C,A)
 onhook(A):
 ringing(A,C),m-cw(A),ringback(C,A).

図 2 - 9 話中着信サービス記述例

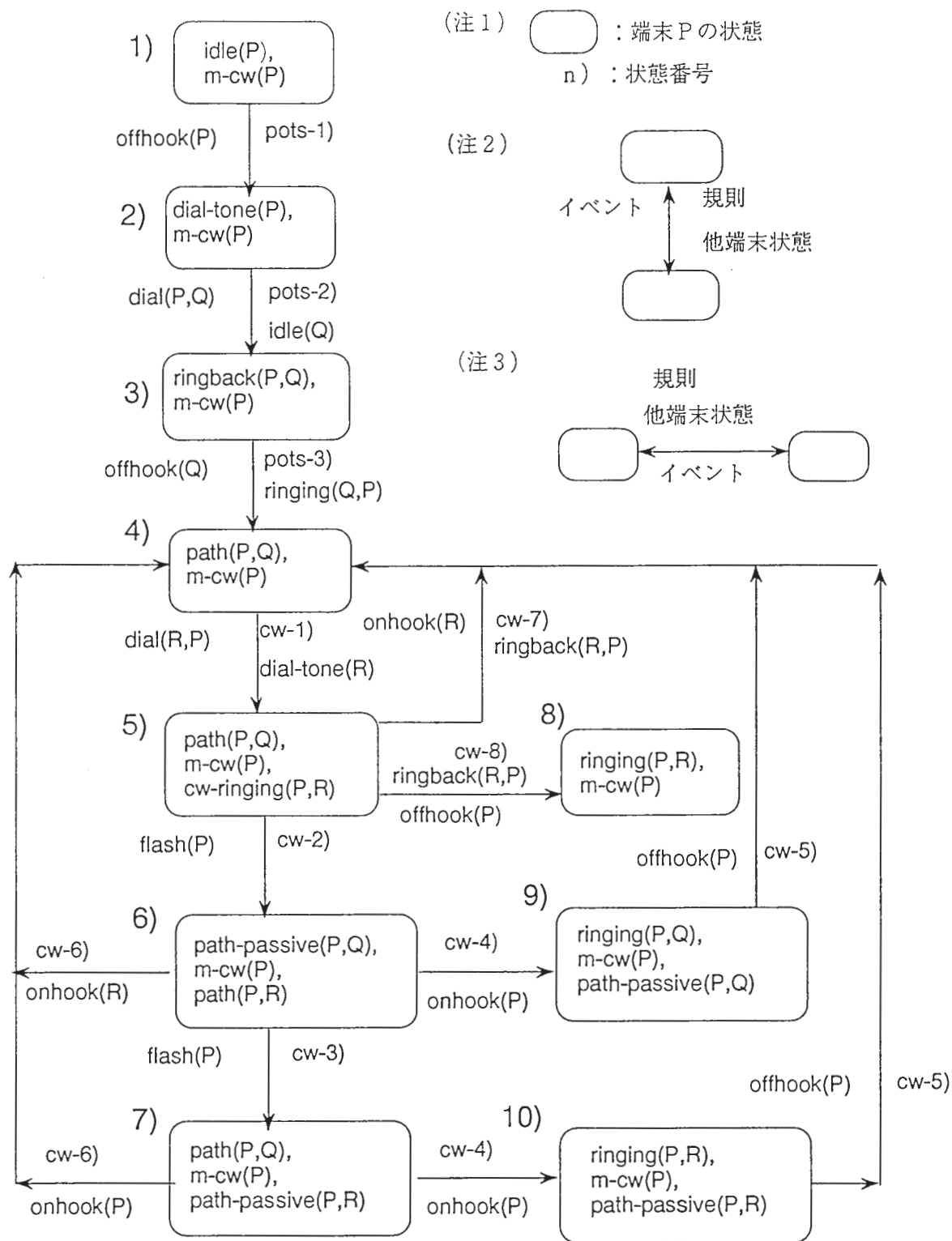


図 2-10 話中着信サービス動作推移

等も規定する必要があるが、基本電話サービス規定を利用することで、これらの推移規定の記述が省略できる。基本電話サービスの規則数、話中着信サービスの規則数とその合計数を以下に示す。

基本電話サービスの規則数： 16

話中着信サービスの規則数： 8

合計数： 24

上記の数値は、本来、規定すべき話中着信サービスの動作規定数は24であるが、基本電話サービスの動作規定を再利用することで、8つの動作規定を行えばよいことを示している。前者：後者＝24：8であり、4割弱程度（33％）の記述ですみ、記述量の削減効果は6割程度であることを意味している。

このように、新規サービスを設計する際には、既存サービス動作の再利用を前提として、新規サービスの特徴的な動作を規定すれば記述量の削減化が可能となる。

2. 7 考察

（1）新規サービスの差分記述

通信サービスを効率よく設計するためには、既存仕様を再利用することが重要な要素である。サービスの個々の動作を規則形式で表現することにより、既存仕様の再利用は規則の再利用の問題として扱うことが可能となる。話中着信サービスの設計例では、既存の基本電話の規則を再利用することで、本来記述すべき仕様の40％程度の記述量ですむ。つまり、サービスの特徴的な動作を記述し、新規サービス仕様と利用するサービス仕様とを組み合わせて仕様を合成し、利用したサービス仕様の状態推移

を取り除けば，目的のサービス仕様が合成でき，新規サービスの設計の効率化が実現できる．

（２）修正の効率化

現状の形式的仕様記述技法では，設計者があますことなく動作を記述することを前提としている．修正が必要な場合，同じ修正を複数箇所に対して行う必要がある．提案手法では既存規則の再利用も含めて仕様合成が可能になる．つまり，同じ修正はひとつの規則修正でよいいため，修正にかかる記述労力の負担が軽減される．

（３）通信サービス記述の網羅性

S T R手法を用いてどれだけのサービスが記述できるかP B Xサービスを対象にして記述実験を行った．

記述実験では4 1個のサービスを対象に5 1個の機能について記述実験を行った．5 1個の機能中4 8個の機能について記述が可能であった．つまり，9 4 %の網羅性があった．記述できなかった機能は，複数端末に対する繰り返し操作や，グループ内の複数端末の制御操作であった．例えば，会議通話サービスにおいて，制限端末数になるまで同じ操作を繰り返す操作の記述は，S T R手法では，制限端末数に達するまで同様な操作を記述する必要があり制限端末数の設定と基本操作を記述するといった記述方法の拡張が必要である．

2. 8 第2章のまとめ

利用者の視点から通信サービス動作を，形式的な規則形式の集合で記述するSTR手法を提案した．STR手法の規則記述と規則適用方法を示した．STR手法を前提とした場合の新規サービス設計時において，既存規則の再利用による記述量の削減化とサービスの特徴的な記述のみ（差分記述）で通信サービスの記述が可能になり効率的な通信サービス設計ができることを示した．PBXサービスを対象にした記述実験から，記述の網羅性は94%であったことを示した．

今後の課題として以下の項目を検討している．

（1）STR手法の機能拡張：

繰り返し操作に関するサービスや，グループ内の端末操作等が記述できるように機能拡張する必要がある．

（2）次工程（システム仕様設計）への引き継ぎ情報の作成：

設計の自動化を想定した場合，通信サービス仕様レベルで検証済みの状態推移情報を生成し，次工程への入力とすれば，生産性の向上が期待できる．

第2章の参考文献

- [1] CCITT Recommendations Z. 100 to Z. 104 (1985).
- [2] 若原恭：SDL言語の特質と処理系の現状と動向，情報処理，Vol. 31，No. 1，pp. 23-34 (1990).
- [3] 岡田康治：Estelle 言語の特質と処理系の現状と動向，情報処理，Vol. 31，No. 1，pp. 47-55 (1990).
- [4] ISO8807: Information processing systems - open systems interconnection - LOTOS - a formal description technique based on the temporal ordering of observational behavior, ISO publication (1988).
- [5] 高橋薫，神長裕明，白鳥則郎：LOTOS 言語の特質と処理系の現状と動向，情報処理，Vol. 31，No. 1，pp. 35-46 (1990).
- [6] Peterson, J. L. : Petri Net Theory and Modeling of Systems, Prentice-Hall (1981).
- [7] J. L. ピータースン著，市川惇信／小林重信訳：ペトリネット入門；情報システムのモデル化，共立出版 (1984).
- [8] Duran J. M and Visser J. :International Standards for Intelligent Networks, IEEE Communication Magazine, Vol. 30, No. 2, pp. 34-42 (1992).
- [9] Collet P. and Kung R. : The Intelligent Network in France, IEEE Communication Magazine, Vol. 30, No. 2, pp. 82-89 (1992).
- [10] Bowen, T. F. , et al. : The Feature Interaction Problem in Telecommunications Systems, Proc. of the 7th IEE Int. Conf. on Soft. Eng. for Telecom. Systems, pp. 59-62 (1989).
- [11] Cameron, E. J., et al. , : A Feature-Interaction Benchmark for IN and Beyond, IEEE Communication Magazine, Vol. 33, No.3, pp. 64-69 (1993).
- [12] 原田良雄，平川豊，竹中豊文，門田充弘：サービス仕様の自動生成に関する考察，情処第39回大会，5S-5 (1989).
- [13] Hirakawa, Y., Harada, Y. and Takenaka, T. :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 (1990).

- [14]NTTビジネスホンE操作マニュアル.
- [15]宗森純, 水野忠則:SDLグラフィックエディタの設計と製作, 情報論文誌, Vol. 29, No. 7, pp. 676-685 (1988).
- [16]Jackson, L. N., Cheng, K. E., Choong, T. S., Pascoe, R. S. V., Cain, G. J. :
MELBA At The Age of Eight: An Automatic Code Generation System,
SDL'87:State of the Art and Future Trends, Saracco, R. and Tilanus, P. A. J (eds.),
pp. 371-381 (1987).
- [17] Vefsnmo, E. : DASOM - A SDL-TOOL, SDL'87: State of the Art and Future
Trends, Saracco, R. and Tilanus, P. A. J (eds.), pp. 35-42 (1987).
- [18] Boumezbeur, R.: Specifying Telephone Systems in LOTOS and Feature Interaction
Problems, Proc. of Int. Workshop on Feature Interactions in Telecommunication
Software Systems, pp. 59-108 (1992).
- [19]長谷川晴朗:ペトリネットを用いた通信サービス仕様設計支援,
信学論文誌, Vol. J74-B-I, No. 6, pp. 445 - 455 (1991).
- [20]横田 実:論理型言語の逐次実行処理方式, 情報処理, Vol. 32, NO. 4,
pp. 421-434 (1991).
- [21] Backus, J. : Can Programminig be Liberated from the von Neumann Style? A Functional
Style and 1st Algebra of Programs, Comm. of the ACM, Vol. 21, No. 8, pp. 613-641
(1978).
- [22] 小林重信:プロダクションシステム, 情報処理, Vol. 26, No. 12, pp. 1487-1495
(1985).

第3章 通信サービス（相互作用）設計支援

3.1 第3章の概要

新規サービスを既存サービスに追加する際、新規サービス仕様と既存サービス仕様との相互作用（サービスインタラクション）を解析し、サービス仕様間に生じる矛盾を解消し、サービス全体として矛盾のない仕様を作成する必要がある。ところが、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し、その相互作用に隠れたサービス仕様矛盾の検証は、人手で行う事のできる範囲を越えた作業となっており、相互作用に含まれる矛盾の検出と解消は重要な課題となっている [1-4]。サービス相互作用に含まれる矛盾の検出・解消は、検討段階である [5-6]。相互作用の矛盾検出の困難さの主な要因は、設計者がサービスを組み合わせた時に生じるであろう状態および推移を把握・予測することが困難なことに起因する。この点を解決する枠組みの確立が、サービス相互作用設計支援を行う上での鍵となる。

形式的仕様記述方式としては、SDL [7, 8] , PetriNet [9, 10] , LOTOS [11, 12] , 等があるが、SDL [7, 8] , PetriNet [9, 10] は、状態推移を明示的に表現するけれども、状態を合成する概念がなく、サービス相互作用の矛盾を検出する機能には限界がある。LOTOS [11, 12] は、サービス動作をイベント系列の集合で規定する手法であるが、状態を明示的に表現できないため、サービス合成における矛盾する状態の解析を行うには自ずと限界がある。

このような状況から、サービスをSTR手法 [13,14] で記述した規則の集合として規定し、サービス合成時における合成状態／推移を生成し相互作用解析の支援を行う手法を提案する。通信サービス設計においては、先ず、新規サービスの設計を完了し、次に、新規サービスと既存サービスとの相互作用の解析を行い、矛盾があればその解消（規則の修正）を行う手順の繰り返しである（図3-1）。

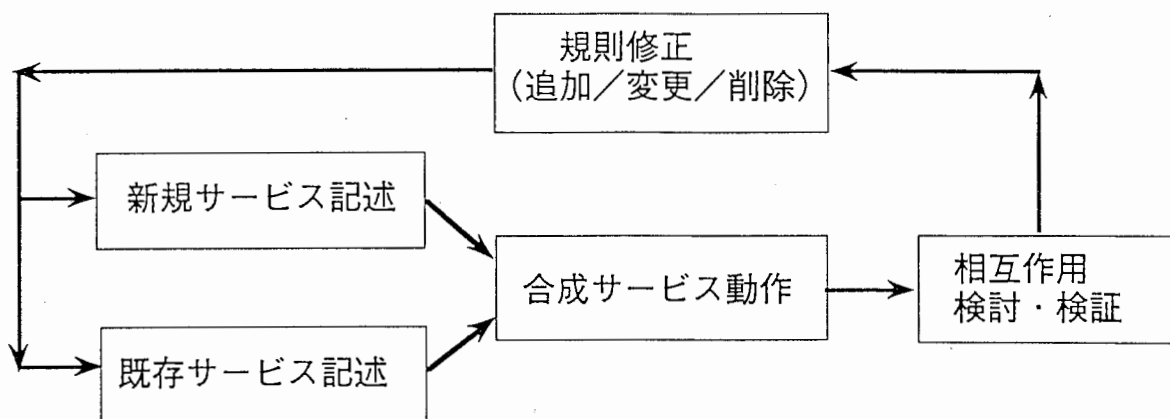


図 3-1 通信サービス相互作用設計の流れ

本章で提案する，新規サービス設計支援と相互作用設計支援の概要を以下に示す．

(1) 新規サービス設計支援

新規サービス単独の設計自体は，扱う状態数，推移数は多くはなく，比較的容易である．設計支援としては，1) 設計者が視覚的な動作確認画面を用いて，サービスの動作に従ってイベントを入力することにより，画面に表示される適用規則と状態の推移（動作）を確認していくプロトタイピング・アプローチ，2) 初期状態から適用可能な規則を順次適用させて生成した動作を状態推移表の形式で設計者に提示し，網羅的な確認を促す方法，を用いる．

(2) 相互作用設計支援

新規サービスと既存サービスを組み合わせたときの，検討すべき相互作用の範囲（状態および推移）が機械的に提示できれば，検討対象の範囲は限定される．設計支援としては，機械的に生成した相互作用の情報（サービス合成状態と推移）を利用して機械的に矛盾を検出できるものと，機械的には矛盾の検出できないものがあるが，後者の場合においても検討範囲が限定されるため，状態および推移を把握・予測する

困難さが解消される分、設計支援として有効である。本章では、相互作用の設計支援として必要な検証項目の分類を示し、検証項目の説明を行い、検出方法の基本的な考え方を示す。研究として提案してきた検証項目に対する検出（支援）方式については第4章以降の章にそれぞれ詳細に述べるものとする。

初期状態から規則を順次適用することによりサービス状態および推移を生成する機構は、サービス合成時および新規サービス動作確認時に共通であり、また、第4章以降も状態および推移の情報を活用して各種の検証支援を行うため、本章でその手法について述べる。

3. 2 新規サービス設計支援

3. 2. 1 プロトタイピング・アプローチ

新規サービス動作の妥当性（設計者の意図どおりのものかどうか）を確認する必要がある。設計者がイベントを入力することにより、適用可能な規則を選択し、動作を視覚的に図で現わすシステムを作成した。このシステムを、解釈実行システム（インタープリタ）と呼ぶ。設計者と解釈実行システムの上記の関係を簡単に図3-2に示し、解釈実行システムの中心画面を図3-3に示す。

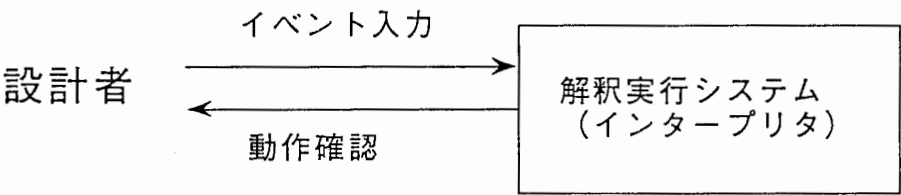


図 3-2 解釈実行システムを用いた動作確認

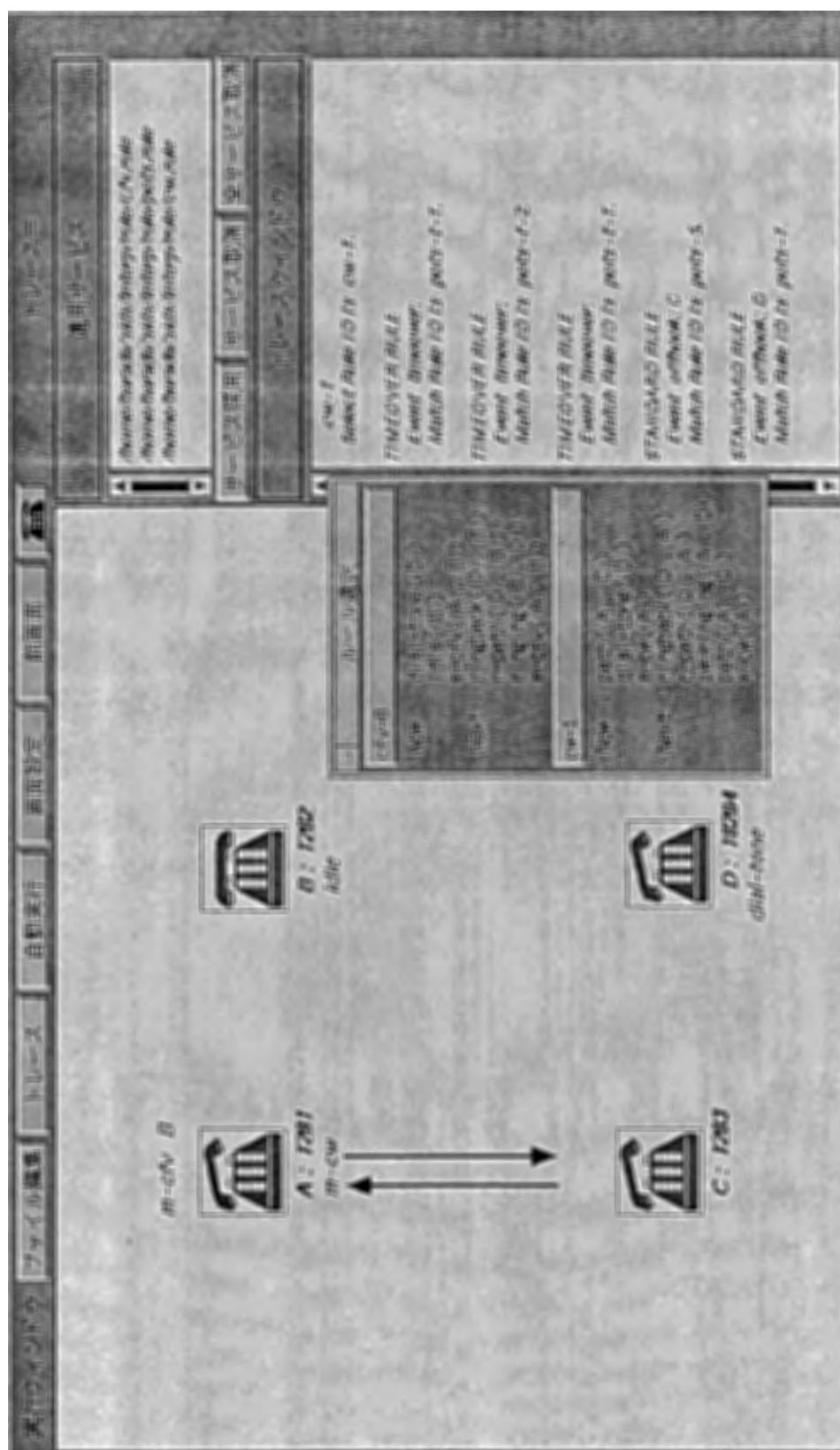


図 3-3 解釈実行システムの画面例

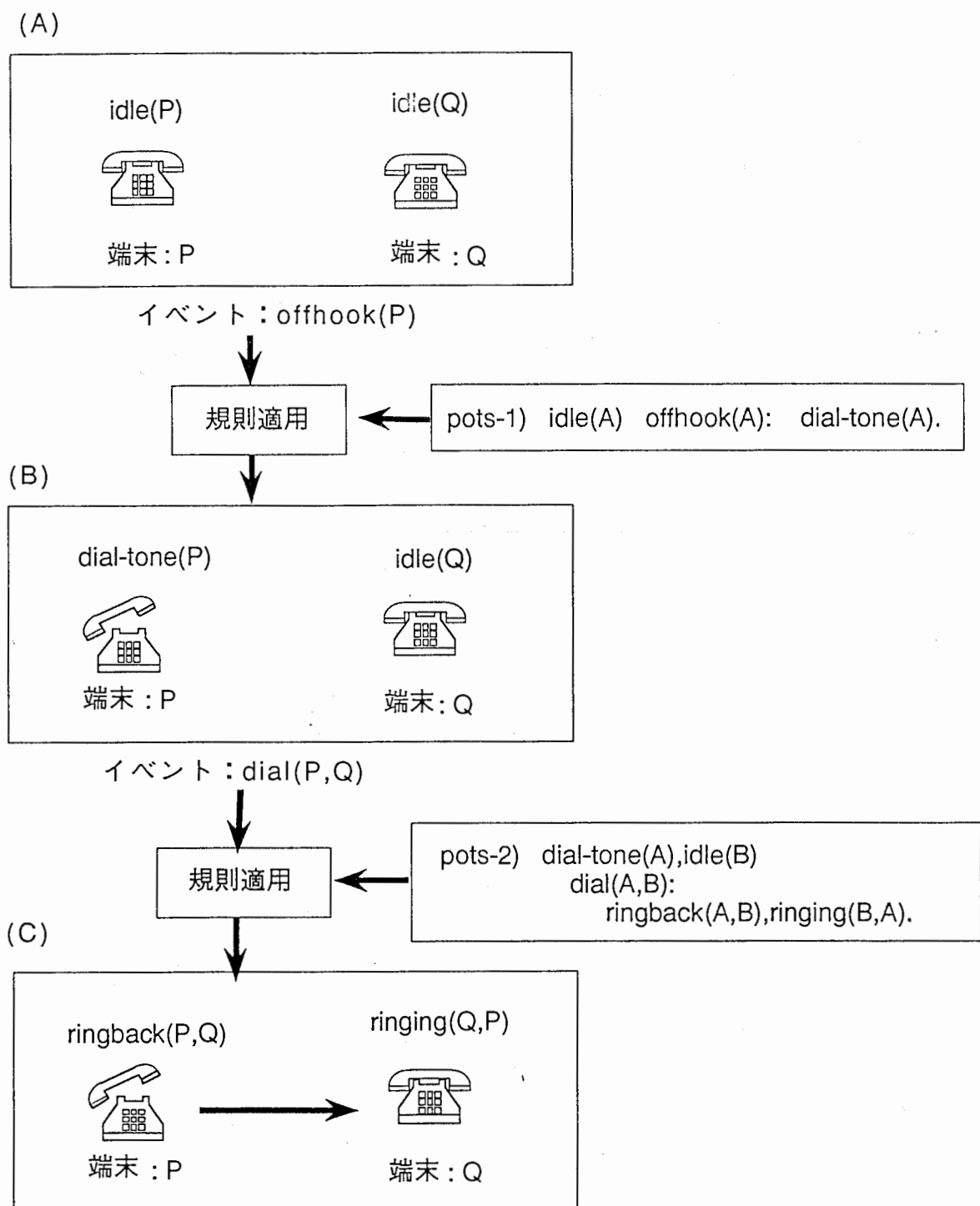


図 3 - 4 確認動作例

例えば、図 3-4 に示す動作の確認を行う場合、端末P、Qの初期状態をそれぞれ「idle(P)」 「idle(Q)」と設定しておき（図 3-4 -(A)の状況に対応する）、端末Pが受話器を上げることを意味するイベント「offhook(P)」を入力すると規則「pots-1)」が適用され、端末Pはダイヤル可能音が聞こえる状態となる（図 3-4 -(B)の状況に対応する）。さらに、端末Pが端末Qにダイヤルを行うことを意味するイベント「dial(P, Q)」を入力すると規則「pots-2)」が適用され、端末Pは端末Qを呼び出している状態となる（図 3-4 -(C)の状況に対応する）。このように、確認したい動作に対応するイベントを順次入力することにより、動作の妥当性の確認が実現可能となる。また、イベント系列をあらかじめファイルに設定しておき、そのファイルを読ませることで、自動的に画面上で動作推移を確認できる機能を付けており、連続して動作確認を行うことができる。

3. 2. 2 状態推移動作提示による確認支援

サービスは規則の集合で記述されており、規則集合を入力とし規則適用の結果、意図どおりの動作であるか、あるいは、意図していない動作が現われていないかどうかの網羅的な確認が必要である。そのため、規則集合を入力として、任意の端末を中心に捉えた動作（状態推移）を生成し、設計者に提示し、妥当性の確認を促す支援を行う。

〔具体例〕生成した状態推移例を表 3-1 に示す。最左欄は任意端末（例では、端末P）の状態を示す。他端末状態は、規則の始状態（複数端末状態の集合を指定可）の中の任意端末以外の端末の状態を示し（例では、端末Qの状態）、イベントは発生可能なイベントを示し、規則は適用された規則番号を示す。次状態は、規則適用後の状態を示す。説明上、各状態推移に「（推移番号）」を付加した。1 番目の状態推移

表 3 - 1 状態推移例

状態	他端末状態	イベント	規則	次状態	(推移番号)
idle(P)		offhook(P)	pots-1)	dial-tone(P)	(1)
	dial-tone(Q)	dial(Q,P)	pots-2)	ringing(P,Q) ringback(Q,P)	(2)
dial-tone(P)	idle(Q)	dial(P,Q)	pots-2)	ringback(P,Q) ringing(Q,P)	(3)

は、端末Pが受話器を上げることを意味するイベント「offhook(P)」を入力すると規則「pots-1」が適用され、端末Pはダイヤル可能音が聞こえる状態となることを示す（図3-4-(B)の端末Pの状態に対応する）。3番目の状態推移は、端末Pが端末Qにダイヤルを行うことを意味するイベント「dial(P, Q)」を入力すると規則「pots-2」が適用され、端末Pは端末Qを呼び出している状態となる（図3-4-(C)の状況に対応する）。2番目の状態推移は、端末Pがアイドル状態かつ端末Qがダイヤル可能状態のときに、端末Qが端末Pにダイヤルを行うことを意味するイベント「dial(Q, P)」を入力すると規則「pots-2」が適用され、端末Qは端末Pを呼び出している状態となることを示す。

[具体例終]

任意の端末を中心に捉えた、状態推移生成のアルゴリズムは、3.3節で示す。

解釈実行システムは、想定しているイベント系列に対する確認に対しては有効であるが、想定していない動作には対応できない。生成された状態推移を確認することと、併用することで、新規サービスの確認ができた。

3.3 サービス動作（状態推移）生成

規則集合と規則適用を用いて、任意のひとつの端末に着目した状態推移を生成する。新規サービス、既存サービス、および合成サービスは規則の集合として捉えることができる。提案する方式を用いれば、規則集合の組み合わせを操作することにより、目的（サービス単独、サービス合成）のサービス動作生成が可能となる。

状態推移生成の方式を以下に示す.

[状態推移生成方式]

着目する端末を P と設定し, 他端末は順次 Q, R, S, . . . を設定する.

手順 1: 端末 P の状態集合に初期状態「idle(P)」を設定する.

手順 2: 端末 P の状態を規則の始状態に含む規則を選択し, (他端末の状態は成立すると仮定して) 規則を適用させ, 次状態を生成する.

手順 3: 矛盾状態知識検査 (第 5 章で説明)

矛盾状態知識検査の結果, 次状態が矛盾状態知識に該当していれば, 手順 2 に戻り, 他規則の適用をおこなう.

手順 4: 次状態のなかで, 端末 P の状態に着目し, その状態が端末 P の状態集合になればその状態を端末 P の状態に付加する.

端末 P の状態集合に付加する状態がなくなるまで, 上記手順 2 から手順 4 を繰り返す.

処理制約: 処理の停止性を保証するために, 同じ状態記述要素の重複がある場合は, 以降, その状態に対する規則適用は行わない. 書き換え規則における仕様規定の誤りであるので, 設計者に提示し解析を促す.

状態管理情報の編集: 手順 1 および手順 4 で状態を追加するとき, 以下に示す状態管理情報を編集する. 状態管理情報を K とすると, K は以下のように定義できる.

[定義 3. 1] (状態管理情報)

$K = \langle S, N, PN, R, TS, E \rangle$

S: 端末 P の状態.

N: 端末 P の状態番号. 新たに端末 P の状態集合に追加するとき採番する.

初期状態にはゼロを設定する. 以後, 追加毎に 1 を加算する.

PN: ひとつ前の端末 P の状態番号.

R： ひとつ前の端末Pの状態に適用された規則番号.

TN：規則適用されたときの他端末の状態.

E： ひとつ前の端末Pの状態からこの状態に移移するときに契機となるイベント.□

[状態推移生成方式終]

[事例を用いた説明]

基本電話サービス規定中の3規則「pots-1), pots-2), pots-3)」(第2章;図2-8参照)を用いて,サービス合成状態推移生成の手順を以下に説明する.

手順1: 端末Pに初期値として初期状態 {idle(P)} を設定する.

端末P = {idle(P)}1)

手順2 (1回目): この状態を規則の始状態に含む規則を選択する. 適用可能な規則はpots-1), pots-2)の2規則である. pots-1) の場合, 規則の端末変数A = Pとみなせば「idle(A)」=「idle(P)」となる. 規則を適用すると, 次状態は以下に示すようになる.

端末P = {dial-tone(P)}2)

pots-2) の場合, 規則の端末変数B = Pとみなすと端末P = {idle(P)} は, pots-2)の「idle(B)」に該当する. pots-2)の始状態の中の「dial-tone(A)」の端末変数AにはQを設定する. この操作で, pots-2)は, 以下の定数をもつ規則に書き変わる.

pots-2) dial-tone(Q), idle(P) dial(Q,P): ringback(Q,P),ringing(P,Q).

端末Pの状態は「idle(P)」であり, 端末Qの状態「dial-tone(Q)」を仮定すると, 規則の始状態は満足され, 規則適用の結果, 以下の状態が作成される.

端末P = {ringing(P,Q)}

端末Q = {ringback(Q,P)}3)

手順3 (1回目): 矛盾状態知識検査. 該当しないので手順4へ進む.

手順 4 (1 回目) : 次状態 2), 3) 中, 端末 P の状態「dial-tone(P)」,

「ringing(P,Q)」は端末 P の状態集合に存在しないので, 追加する. 手順 2 へ戻る.

手順 2 (2 回目) : 端末 P の状態集合 = {「idle(P)」, 「dial-tone(P)」,

「ringing(P,Q)」} であり, 適用可能な規則は, 「dial-tone(P)」に対してはpots-2),

「ringing(P,Q)」に対してはpots-3) である. 前者の規則適用を行うと次状態は以下に示すようになる.

端末 P = {ringback(P,Q)}

端末 Q = {ringing(Q,P)} 4)

後者の規則適用を行なうと, 次状態は以下のようなになる.

端末 P = {path(P,Q)}

端末 Q = {path(Q,P)} 5)

手順 3 (2 回目) : 矛盾状態知識検査. 上記次状態 4), 5) は矛盾状態知識に該当しないので手順 4 へ進む.

手順 4 (2 回目) : 上記次状態 4), 5) 中, 端末 P の状態は「ringback(P,Q)」,

「path(P,Q)」であり, 端末 P の状態集合には存在していないので追加する.

この様に, 端末 P の状態集合に付加する端末 P の状態がなくなるまで手順 2 から手順 4 を繰り返す.

状態管理情報の編集 : 手順 1 および手順 4 における状態管理情報 ($K = \langle S, N, P, N, R, TS, E \rangle$) の編集例を以下に示す. 「-」は情報が設定されていないことを示す.

手順 1 : 初期状態 (idle(P)) の設定時

初期状態と状態番号の初期値「0」を設定する. 初期状態の状態管理情報を以下に示す.

$K0 = \langle \text{idle}(P), 0, -, -, -, - \rangle$

手順4 (1回目) : 端末Pの状態「dial-tone(P)」, 「ringing(P,Q)」の追加時

状態「dial-tone(P)」の追加のとき, 状態番号は採番(0+1)後, 「1」となる. ひとつ前の状態番号は「0」, 適用規則は「pots-1」, 他端末状態は無く, イベントは「offhook(P)」である. この情報から状態「dial-tone(P)」の状態管理情報は, 以下のようになる.

$K1 = \langle \text{dial-tone}(P), 1, 0, \text{pots-1}, -, \text{offhook}(P) \rangle$

状態「ringing(P,Q)」の追加のとき, 状態番号は採番(1+1)後, 「2」となる. ひとつ前の状態番号は「0」, 適用規則は「pots-2」, 他端末状態は「dial-tone(Q)」であり, イベントは「dial(Q,P)」である. この情報から状態「ringing(P,Q)」の状態管理情報は, 以下のようになる.

$K2 = \langle \text{ringing}(P,Q), 2, 0, \text{pots-2}, \text{dial-tone}(Q), \text{dial}(Q,P) \rangle$

このように, 端末Pの状態追加時に端末管理情報を編集する.

[事例を用いた説明終]

図2-8の基本電話サービス規定中の3規則「pots-1), pots-2), pots-3)」を用いて生成した状態推移例を表3-2に示す. 表3-2の最左欄の「状態」は端末Pの取りえる状態を示す. 「他端末状態」は規則適用時に必要な端末P以外の状態を示し, 「イベント」は生起するイベントを示し, 「規則」は適用された規則識別子を示し, 「次状態」は規則適用後の状態を示す.

表 3-2 状態推移生成例

状態	他端末状態	イベント	規則	次状態
idle(P)		offhook(P)	pots-1)	dial-tone(P)
	dial-tone(Q)	dial(Q,P)	pots-2)	ringing(P,Q) ringback(Q,P)
dial-tone(P)	idle(Q)	dial(P,Q)	pots-2)	ringback(P,Q) ringing(Q,P)
ringing(P,Q)	ringback(Q,P)	offhook(P)	pots-3)	path(P,Q) path(Q,P)
ringback(P,Q)	ringing(Q,P)	offhook(Q)	pots-3)	path(P,Q) path(Q,P)

3. 4 通信サービス相互作用設計支援

3. 4. 1 検証項目の分類

一般に、通信サービス相互作用の問題点の範囲は、利用者の視点からみたサービスの動作だけではなく、プロトコル、システム構成、実行時のタイミングの問題まで含めた広範囲を対象に議論されている [2,4,5]。しかしながら、具体的な問題点、解決するために用いるサービス記述形式、システムモデル、あるいはアルゴリズム、等は検討段階である。現状では、熟練した設計者があらかじめ問題を予測して、仕様を決定する域を脱していない。STR手法を用いて記述できる範囲は、利用者の視点からみたサービスの動作（状態推移）であるが、サービス設計の上流段階で、矛盾を解析でき、解消できることは上流工程の誤りを下流工程に引き継がないという観点から意義がある。例えば、国際電信電話諮問委員会（CCITT）で開発し勧告として標準化した仕様記述言語SDLを用いた検証項目の報告では、仕様誤りの分類として、誤りを判定するための基準があらかじめ定義できるか否かに応じて、誤りを論理誤りと意味誤りに分けている（表3-3） [8]。ところが、相互作用としてどういうことを検証すべきなのかという観点からの検証項目については明示されていない。

通信サービスはFSM（Finite State Machine）に代表される状態推移モデルとして扱うことができる。従って、検証項目を分類する一般的な基準として、状態推移モデル上の構造的な問題であるか否かを分類の一つの判定基準として設定した（表3-4）。状態推移モデル上の構造的な問題は、状態、イベント、あるいは推移の意味まで検査しなくても検出できる問題であるので、論理検証に分類する。一方、状態推移モデルの構造としては問題はないが、状態、イベント、推移の意味を解析する必要がある問題は、意味検証に分類することとした。

検証項目の分類を表3-4に示す。論理検証としては、1) 推移先のないデッド

表 3-3 SDL 仕様誤りの分類 [文献 8]

仕様誤り	誤りの定義	具体例
論理誤り	誤りの判定基準があらかじめ明確に定義でき多くのシステムに共通なもの	不完全誤り ・宣言されたが使用されないデータ, 信号, ブロック, プロセス, 等 内部矛盾誤り ・信号の入力・出力の対応不一致, 等 実行不可能誤り ・デッドロック, 等
意味誤り	誤りの判定基準が個々のシステムに固有であらかじめ明確には定義できないもの	もとの要求に対する不一致

表 3-4 通信サービス合成仕様誤りの分類

合成仕様誤り	誤りの定義	具体例
論理誤り	状態推移モデル上の構造的な誤り	1) デッドロック 2) ループ 3) 非決定性 等
意味誤り	状態推移モデルの構造上には問題のない意味的な誤り	4) 異常状態 5) サービス干渉 (妨害) 6) 合成動作の妥当性

ロック， 2) 抜け出すことができないループ， 3) 同じイベントによる複数の推移先がある非決定性（非決定的動作），等がある．意味検証としては， 4) 異常状態， 5) サービス干渉（妨害）， 6) 動作の妥当性確認，等がある． 4) 異常状態とは，意味的におかしい状態，例えば，ある端末の状態がアイドルかつ通話中のような状態をいう． 5) サービス干渉（妨害）とは任意サービスの動作が他サービスにより影響を受けることをいう．例えば，任意サービスのある推移可能な状態であるにもかかわらず，他サービスの規定により当該推移が出現しない，等がある． 6) 動作の妥当性とは，サービス設計者の視点から，推移の妥当性の確認を行うことをいう．個別サービス仕様の検証では，個別サービス動作の妥当性確認が必要であることと同様に，サービス仕様合成時の検証では，新たに発生する状態および推移の妥当性の確認を行うことをいう．

3. 4. 2 検証項目に対する検出支援方法の概要

(1) デッドロック検出：

次状態への推移先がない状態をいい，いずれの規則も適用できなくなる状態を検出すればよい．

(2) ループ検出：

正常な状態推移は，初期状態から始まって初期状態に戻る．出口のない推移系列（ループ）を検出するためには，初期状態に至る全ての推移を逆に初期状態まで辿り，辿れた状態に印をつける．印のつかなかった状態を検出すれば，その状態間で出口のない推移系列（ループ）が検出できる．

（３）非決定性検出：

非決定性の例として、ある端末Pが着信転送サービスと話中着信サービスに加入している場合を示す。着信転送サービスは、あらかじめ着信先を登録しておけば、着信は自動的に転送先に転送されるサービスである。一方、話中着信サービスは、通話中に他からの呼び出しを受けるサービスである。端末Pが両方のサービスに加入しており、端末Qへ転送登録をし、かつ端末Rと通話中の状態の時、端末SからPへダイヤルがあったとき、どちらのサービスの動作規定も満足され、機械的には一意に決められない非決定性の問題が生じる。状態推移モデル上の構造上の問題として、ある状態において、同一イベントが発生したとき、異なる次状態が発生する非決定性の検出は、同一のイベントを持つ任意の２規則を調べ、同時に２規則の始状態を満足する状況が存在しえるかどうかを検査する方法を用いる。非決定性の検出方法については、第４章において詳細に述べる。

（４）意味的矛盾状態の検出：

あらかじめ異常な状態記述要素の組み合わせを知識として与えておき、その組み合わせが生成した状態に含まれる場合を検出する。知識獲得支援および検出支援方法は、第５章において詳細に述べる。

（５）サービス干渉（妨害）検出：

サービス干渉（妨害）とは任意サービスの動作が他サービスにより影響を受けることをいう。サービス干渉（妨害）の例を以下に示す。

例１）あるサービスXの動作規定R１が適用できる状態S１のとき、他サービスの規定が適用された結果、別な状態S２に変化し、あるサービスXの動作規定が適用できなくなる。これは、利用者の視点から見たとき、あるサービスの操作中の状態におい

て、突然、状態が変わり、サービス操作が続行できないという現象である。この場合は、状態S 1にあるサービスXの規則R 1が適用可能な状態が、他サービスの規則適用により、適用できなくなる場合を検出すればよい。

例 2) あるサービスXの動作規定R 1が適用できる状態S 1のとき、他サービスの規定が適用された結果、別な状態S 2に変化し、動作規定R 1が適用できなくなるが、サービスXの他の動作規定R 2は適用できる。これは、サービスXとしては、想定している状態への推移であるが、他サービスの適用による推移であり、サービスXとしては想定していない推移である。

これは、利用者の視点から見たとき、あるサービスの操作中の状態において、突然、状態が変わり、サービスXの他の操作により初期状態まで戻ることができるが、そのような状態S 1からS 2への推移は想定していないという現象である。これは、あるサービスXの規則R 1が適用可能な状態S 1に対して、他サービスの規則適用の結果、適用できなくなり、かつ、その状態S 2はサービスXの状態集合に含まれる場合を検出すればよい。

例 3) あるサービスXの動作規定R 1が状態S 1に対して適用されたにもかかわらず、他サービスの規定により妨害を受けた結果、動作規定R 1の指定した次状態S 2とはならない。具体例として、通話相手を一度保留にしておき、再度、通話にするサービス（例えば、3者通話サービス、話中着信サービス等）と保留されることを拒み保留要求を無視する緊急サービスとの組み合わせにおいては、保留要求が無視され、あるサービス側の視点から、正常な手順で操作したにもかかわらず、目的の状態とはならない現象である。これは、S 1から規則R 1適用の結果、S 2に推移しない場合を検出すればよい。

[事例終]

(6) 動作の妥当性確認：

個別サービス仕様の検証では、個別サービス動作の妥当性確認を行うことをいい、サービス仕様合成時の検証では、新たに発生する状態および推移の妥当性の確認を行うことをいう。

相互作用動作の検出の方法を以下に示す。

(A) 追加サービス状態推移の集合

(B) 既存サービス状態推移の集合

(C) 全体サービス状態推移の集合（追加サービス規則と既存サービス規則から生成），が生成されていると想定する。

演算単位は、状態推移表のひとつの行の情報，つまり，「端末Pの状態，他端末状態，イベント，適用規則，次状態」の組とする．演算単位をUとし，上記の3種類の状態推移の集合をそれぞれSA，SE，STとすると，SA，SE，STは以下の用に演算単位の集合として定義できる．

[定義3. 2]

$$SA, SE, ST = \{U_0, \dots, U_n\} \quad \square$$

相互作用動作をIAとし，上記定義3. 2の3種類の状態推移を用いると，IAは以下のような集合演算式で求まる．

[相互作用動作検出式]

$$IA = ST - SA - SE + SA \wedge SE \quad \square$$

また，規則修正を行う際には，修正影響波及範囲の確認も設計支援としては必要な機能である．修正前の状態推移の集合をSO，修正後の状態推移の集合をSNとし，

演算単位は「端末Pの状態，他端末状態，イベント，適用規則，次状態」の組とし，修正によって影響のある動作（修正によって新たに生成された動作と修正によって現われなくなった動作）をEAとすると，EAは以下のような集合演算式で求まる．

[修正影響動作検出式]

$$EA = SO - SN$$

□

3. 5 考察

(1) 新規サービス設計支援

新規サービス設計支援としては，1) 設計者が視覚的な動作確認画面を用いて，サービスの動作に従ってイベントを入力することにより，画面に表示される適用規則と状態の推移（動作）を確認していくプロトタイピング・アプローチを示し，また，2) 初期状態から適用可能な規則を順次適用させて生成した動作を状態推移表の形式で設計者に提示し，網羅的な確認を促す方法を示した．両者を組み合わせて用いることにより，新規サービスの確認は，容易に実現できた．

(2) 相互作用設計支援

相互作用設計の困難さの主要な理由は，サービスを組み合わせるときに生じるであろう状態，推移の予想が難しいことが挙げられる．新規サービスと既存サービスを組み合わせることによって生じる状態および推移は，規則の集合を組み合わせで入力して生成できる方法を示した．

実験の結果を表3-5に示す．サービスの対象は，POTS (Plain Old Telephone Service)，CW (Call Waiting)，CFV (Call Forwarding Variable) であり，「STR

規則数」は、サービス組み合わせ時の規則数の合計を示し、「任意端末Pの状態数」は、サービス組み合わせ時に生成される端末Pの状態数を示す。また、状態推移数は、サービス組み合わせ時に生成される推移；「端末Pの状態，他端末状態，イベント，適用規則，次状態」の組の合計数を示す。POTSの場合は、状態数は「8」，推移数は「2 8」である。POTS+CWの場合は、状態数は「2 0」，推移数は「9 3」，POTS+CFVの場合は、状態数は「3 2」，推移数は「4 4 9」である。3つのサービスを合成した場合、状態数は「7 3」，推移数は「1 1 9 2」である。これは、新たに生じた状態数は2 9 ($73 - 32 - 20 + 8$)，推移数は6 7 8 ($1192 - 449 - 93 + 28$)を示している。提案手法を用いることにより、新規サービスと既存サービスから新規サービスと既存サービスの規定している状態／動作以外の状態／動作が相互作用の検討対象候補として設計者に提示される。設計者はその情報に対して検討を行えばよく、仕様決定における考慮もれの防止を支援することが可能となる。また、相互作用に対する仕様決定時のケース想定の手力が省略される分、設計効率が向上する。既存サービス数が増加すると検討範囲も組み合わせ的に増加するため、さらに項目別の機械的な検出情報と組み合わせると設計支援としてより有効となる。

(3) 停止性について

ソフトウェアの検証として、停止性と正当性が論じられている [23]。STR手法を用いて宣言的に記述された規定を適用し状態推移の生成を行う処理が停止することは、以下の要因により明らかである [19]。1) 状態記述要素の数は有限であり、任意端末の取りうる状態記述要素も有限である。また、発生するイベントの数も有限であり、規則規定の数も有限である。2) 同じ状態記述要素の重複は、処理の制約として規制している。3) 検証項目 (表 3-3, 表 3-4 参照) のループ検出により無限ループに陥る場合は検出される。

表 3-5 サービス合成実験結果

	POTS	POTS+CW	POTS+CFV	POTS+CFV+ CW
STR規則数	14	33	33	54
任意端末Pの 状態数	8	20	32	73
状態推移数	28	93	449	1192

（４）検証（確認）支援の正当性について

サービス規定を機械的に検証するためには、何らかの基準を外部から与える必要がある。通信ソフトウェアでは、論理検証項目（表 3-3 参照）が一般的な論理検証項目（基準）である [9]。この検査項目を通過すれば、論理上なんら問題なく初期状態から初期状態までの状態推移が実現できる。サービス仕様上、意味的な問題の妥当性確認の必要がある。この場合、単体サービスに関しては、本章で述べたプロトタイプ・アプローチ支援と状態推移表の提示による確認支援が経験上有効であった。サービス合成においては、論理検証項目のチェックは当然であるが、個々のサービスの規定（状態、推移）がサービス間で相互に影響を与え合うことがないことを調べることにより合成サービス仕様の正当性は満たされる。

3. 6 第 3 章のまとめ

本章では、発表文献 [15-22] に基づき、新規サービスの設計支援の手法および相互作用設計支援の手法を述べた。相互作用に含まれる矛盾の検出方式の詳細については、第 4 章以降に述べる。

（１）新規サービス設計支援として、１）設計者が視覚的な動作確認画面を用いて、サービスの動作に従ってイベントを入力することにより、画面に表示される適用規則と状態の推移（動作）を確認していくプロトタイプ・アプローチを示した。また、２）初期状態から適用可能な規則を順次適用させて生成した動作を状態推移表の形式で設計者に提示し、網羅的な確認を促す方法を示した。両者を組み合わせて用いることにより、新規サービスの確認が容易に実現できた。

（２）相互作用設計支援として、１）検証項目の分類、２）検証項目の説明と検出方法の考え方を示した。また、規則修正時の影響波及範囲の検出も可能になることを

示した.

また, 新規サービス設計支援および相互作用設計支援に共通に必要な状態および推移を, 任意の端末に着目して生成する方法を示した.

第3章の参考文献

- [1] Duran, M. J. and Visser, J. : International Standards for Intelligent Networks,
IEEE Communication Magazine, Vol. 30, No. 2, pp. 34-42 (1992).
- [2] Collet, P. and Kung, R.: The Intelligent Network in France,
IEEE Communication Magazine, Vol. 30, No. 2, pp. 82-89 (1992).
- [3] Bowen, T. F., et al., :The Feature Interaction Problem in Telecommunications
Systems, Proc. of the 7th IEE Int. Conf. on Soft. Eng. for Telecom. Systems,
pp. 59-62 (1989).
- [4] Cameron, E. J. and Velthuijsen, H.: Feature Interactions in Telecommunications
Systems, IEEE Communication Magazine, Vol. 31, No. 8, pp. 18-23 (1993).
- [5] Wakahara, Y., et al., : A Model for Detecting Service Interactions,
IEEE Communication Magazine, Vol. 31, No. 8, pp. 32-37 (1993).
- [6] 長谷川晴朗：ペトリネットを用いた通信サービス仕様設計支援,
信学論文誌, Vol. J74-B-I, No. 6, pp. 445-455 (1991).
- [7] CCITT Recommendations Z. 100 to Z. 104 (1985).
- [8] 若原 恭：SDL言語の特質と処理系の現状と動向, 情報処理,
Vol. 31, No. 1, pp. 23-34 (1990).
- [9] Peterson, J. L.: Petri Net Theory and Modeling of Systems, Prentice-Hall (1981).
- [10] J. L. ピータースン著, 市川惇信／小林重信訳：ペトリネット入門；
情報システムのモデル化, 共立出版 (1984).
- [11] ISO8807: Information processing systems - open systems interconnection -
LOTOS - a formal description technique based on the temporal ordering of
observational behavior, ISO publication (1988).

- [12] 高橋薫, 神長裕明, 白鳥則郎: LOTOS 言語の特質と処理系の現状と動向, 情報処理, Vol. 31, No. 1, pp. 35-46 (1990).
- [13] 原田良雄, 平川豊, 竹中豊文, 門田充弘: サービス仕様の自動生成に関する考察, 情処第39回大会, 5S-5 (1989).
- [14] Hirakawa, Y., Harada, Y. and Takenaka, T.: Behavior Description for a System which Consists of an Infinite Number of Processes, BILKENT Int. Conf. on New Trends in Communication, Control, and Signal Processing, pp. 59-68 (1990).
- [15] 原田良雄, 平川豊, 竹中豊文: 通信サービス追加設計支援方法の一考察, 信学技法, Vol. 90, No. 454, pp. 13-18 (1991).
- [16] Harada, Y., Hirakawa, Y. and Takenaka, T.: A Design Support Method for Telecommunication Service Interactions, GLOBECOM'91, pp. 1661-1666, (1991).
- [17] 原田良雄, 平川豊, 竹中豊文: 通信サービス動作競合検出方法, 情処研報, Vol. 91, No. 83, pp. 87-92 (1991).
- [18] 原田良雄, 平川豊, 竹中豊文: 通信サービス競合検出システムの試作と評価, 信学技法, Vol. 92, No. 75, pp. 13-18 (1992).
- [19] Harada, Y., Hirakawa, Y., Takenaka, T., Terashima, N.: A Conflict Detection Support Method for Telecommunication Service Descriptions, 信学論文誌, Vol. E-75-B, No. 10, pp. 986-997 (1992).
- [20] 原田良雄, 高見一正, 太田理, 寺島信義: 通信サービス相互作用に含まれる意味的矛盾動作検出方式, 情処論文誌, Vol. 35, No. 8, pp. 1602-1613 (1994).
- [21] Takami, K., Harada, Y., Ohta, T., Terashima, N.: A Visual Design System for Telecommunication Services, IPCCC, pp. 593-599 (1993).

- [22] 原田良雄，高見一正，太田理：通信サービス仕様合成における検証項目の分類の一検討，1994信学春季大会，B-619 (1994).
- [23] 斎藤信男，米崎直樹：ソフトウェアの検証，ソフトウェア工学ハンドブック，オーム社，pp. 177-221 (1986).

第4章 通信サービス競合（非決定性）検出支援方式

4.1 第4章の概要

通信サービス設計段階の検証項目の分類については、第3章において述べた。状態推移モデル上の問題として、同一イベントに対して複数の推移先が生じる非決定性の問題は、規則適用の観点からみると、複数の動作規定が適用可能になり一意に決めることのできない規則競合（非決定性）の問題として扱うことができる。本章では、規則競合（非決定性）検出支援方式について述べる。

提案する競合検出方法は、2規則の始状態から2規則が同時適用される可能性のある状況（競合候補）を作成し、実状態集合（第3章で述べた任意端末状態の集合）を用いて、その状況（競合候補）が実際に在りえる状況かどうかを判定する。在りえない場合は、競合候補から除外する。また、規則適用規約2（定義2.8参照）を考慮して、2規則が優先関係にある場合は競合しないと判定する。競合候補に適用可能な他規則があり、他規則が2規則のいずれか、あるいは、両方に優先適用される場合は競合候補から除外する。つまり、本手法の特徴は、2規則が明らかに競合しない条件を判定基準に用い、その判定基準に該当する場合、競合候補から除外する方法である。この判定を通過した競合候補が在る場合、その競合候補の2規則は競合する可能性があるかと判定する。検出された2規則の競合する状況を画面に表示し、設計者の検証を促す。提案方式をシステム試作し、実際の5つのサービス記述の組み合わせを行い競合検出実験を行った。実験結果から、提案した競合検出手法は、機械的に検討対象を絞り込む効果が大きく（POTS+ CW + CFV + 3WC + CCBSでは1%以下）設計支援に有効であることが確認できた。また、検出結果は、競合する可能性のある2規則の組み合わせ網羅していることを示す。

4. 2 通信サービス競合（非決定性）

S T R手法では、一つの推移は一つの規則で記述される。規則の構文は、「始状態
イベント： 次状態」で表される。イベントが生起したとき、始状態を満足する状況があるとき、規則は適用される。つまり、競合検出は適用可能な規則が複数有り、実行させる規則がひとつに決めることができない規則競合検出の問題として扱うことができる。

S T R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図4-1に示す。図4-1の2つの規則は、以下の動作を既定している。

話中着信動作（C W）：

端末Aがダイヤル可能状態（dial-tone(A)），かつ端末Bが話中着信に加入している（m-cw(B)），話中（path(B, C)）のとき、端末Aが端末Bにダイヤル（dial(A, B)）を行うと、端末Bへの話中着信（cw-ringing(B, A), ringback(A, B)）となる。

着信転送動作（C F V）：

端末Aがダイヤル可能状態（dial-tone(A)），かつ端末Bが着信転送を登録（m-cfv(B, D)）しており通話中（path(B, C)）状態、かつ着信転送登録先端末Dがアイドル（idle(D)）のとき、端末Aが端末Bにダイヤル（dial(A, B)）を行うと、着信転送登録先へ着信（ringing(D, A), ringback(A, D)）となる。

図4-1の2規則が同時に適用できる状況として図4-2の（A）に示す状況が想定される。つまり、端末Pはダイヤル可能状態（dial-tone(P)），かつ端末Qが端末Rと話中（path(Q, R)），話中着信に加入し（m-cw(Q)），着信転送を端末Sに対して登録（m-cfv(Q, S)）しており、かつ端末Sが初期状態（idle(S)）である。端末Pはダイヤ

話中着信 (CW:Call Waiting)

dial-tone(A),m-cw(B),path(B,C)

dial(A,B):

ringback(A,B),cw-ringing(B,A),m-cw(B),path(B,C).

着信転送 (CFV :Call Forwarding Variable) :

dial-tone(A),path(B,C),m-cfv(B,D),idle(D)

dial(A,B):

ringback(A,D),ringing(D,A),path(B,C),m-cfv(B,D).

図 4 - 1 話中着信動作と着信転送動作の記述例

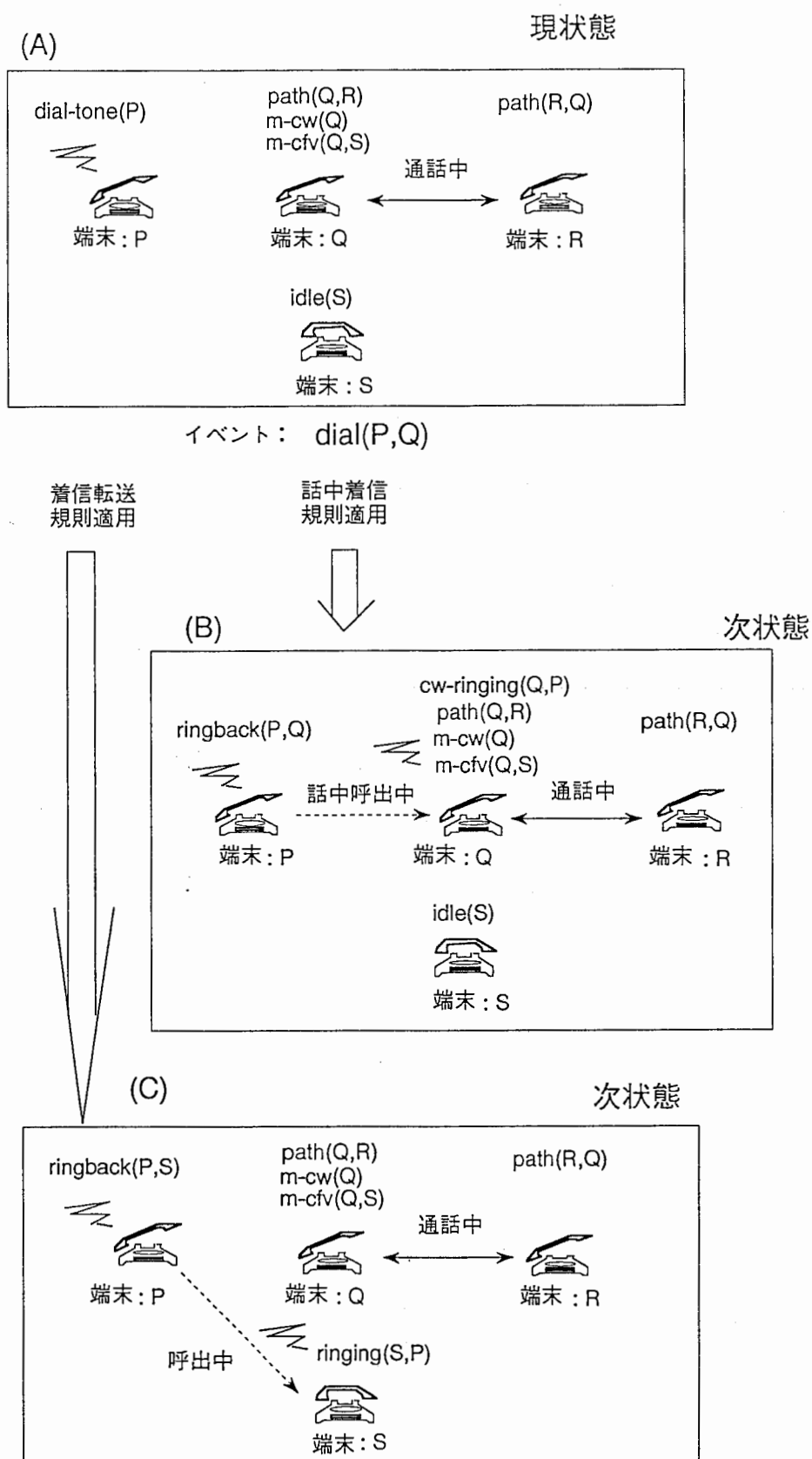


図 4-2 規則競合例

ル可能状態 (dial-tone(P)) である。

この状況で、端末Pから端末Qへダイヤルしたとき (イベント: dial(P,Q)) , 図4-1の2つの規則の条件 (始状態) は両方とも満足され、図4-2に示す2つの異なる状態推移が起動されることになり、動作規定を一意に決定できない動作矛盾が発生する。

4. 3 通信サービス競合 (非決定性) 検出方式の基本概念

2つの規則が同時に適用可能検査の基本的な考え方を以下の2規則を用いて説明する。

規則1) $p1(A, B) \quad ev1(A, B): \quad p2(A, B).$

規則2) $p3(A, B), p4(B, A) \quad ev1(A, B): \quad p5(A, B), p6(B, A).$

規則1)の始状態は「 $p1(A, B)$ 」である。これは端末Aが端末Bに対して「 $p1$ 」の状態であることを示す。このとき、端末Aが端末Bに対して $ev1$ のイベントを行うと、次状態は「 $p2(A, B)$ 」になる。規則2)の始状態は「 $p3(A, B), p4(B, A)$ 」である。これは、端末Aが端末Bに対して「 $p3$ 」の状態であり、端末Bが端末Aに対して「 $p4$ 」の状態であることを示す。このとき、端末Aが端末Bに対して $ev1$ のイベントを行うと、次状態は「 $p5(A, B), p6(B, A)$ 」になる。

端末A, Bごとに、この2つの規則の始状態の状態記述要素の和集合を作成すると以下のようなになる。

端末A = $\{p1(A, B), p3(A, B)\}$

端末B = $\{p4(B, A)\}$

この状況で、端末Aが端末Bに対してev1のイベントを行うと、2つの規則は同時適用可能になり競合が発生する。つまり、2つの規則が同時適用可能かどうかの判定は、2つの規則の始状態から端末毎の状態記述要素の和集合をとった状況が存在可能かどうかの判定問題として扱うことができる。

競合しない例を、以下に示す2つの規則を用いて示す。

規則1) dial-tone(A), idle(B) dial(A, B):

ringback(A, B), ringing(B, A).

規則2) dial-tone(A), not[idle(B)] dial(A, B):

busy-dial(A,B).

端末A, Bごとに、この2つの規則の始状態の状態記述要素の和集合を作成すると以下のようになる。

端末A = {dial-tone(A)}

端末B = {idle(B), not[idle(B)] }

端末Bの状態はアイドル状態 (idle(B)) とアイドルを含まない状態 (not[idle(B)]) の組み合わせになっており、この組み合わせはありえない状態であることがわかる。よって、この2規則は競合しないと判定される。

競合検出の手順の概要を以下に示す。

手順1：実状態集合の作成

任意の端末の取りえる状態を数え上げる。

[定義 4. 1] 実状態集合

任意の端末の取りえる状態を実状態集合と呼ぶ。

□

この実状態集合は、第 3 章で説明した状態推移表の際左欄（端末 P の状態集合）に該当するので、本章では実状態集合は作成済みとする。

手順 2：競合候補検出

(1) 2 規則抽出：

同一イベントの 2 規則の組み合わせを規則集合から抽出する。

(2) 2 規則間の優先適用検査：

規則適用規約 2（第 2 章；定義 2. 8 参照）に該当しているかどうかの検査を行う。該当している場合は、競合しないと判定する。

(3) 競合候補作成：

2 規則の始状態から端末変数の可能な組み合わせを作成し、それぞれの端末変数の組み合わせごとに各端末毎の状態記述要素の和集合を作成する。これを、競合候補と呼ぶ。

2 規則から作成された競合候補は複数存在する場合がある。以下に示す検査（4）～（6）は競合しない条件に該当した競合候補を除外するものであり、その結果、競合候補がひとつでも残った場合は、2 規則は競合する可能性があると判定する。

(4) 競合候補の存在可能性検査：

競合候補の全端末状態が実状態集合に存在しているかどうか調べる。競合候補の全端末状態が実状態集合に存在していない場合は競合候補は競合しないと判定し競合候補から除外する。これは、1) 規則が適用可能なためには、規則の始状態が満足され

る状態が存在する必要がある， 2) もし端末ごとの状態が存在しなければ規則は適用できない，という規則適用の特性を用いて判定するものである．

(5) 端末間の制約検査：

競合候補の全端末状態が実状態集合に存在したとしても，各端末を組み合わせた状況として実際には存在しない状況がある．この状態候補を取り除くために，知識としてありえない端末間の状態の組み合わせをあらかじめ与えておき，競合候補がその知識を含んでいる場合，競合候補から除外する．

ありえない端末間の状態の組み合わせを端末間制約知識とよび，この知識を「TK」とし，競合候補を「C」としたとき，端末間の制約検査は，以下に示す包含関係を判定して，包含関係にある場合は，競合候補から除外する．

$$C \supseteq TK$$

(6) 他規則優先適用検査：

競合候補を満足する状況に対して，2 規則のいずれかに優先して適用される他の規則（イベントは 2 規則と等しく，2 規則のいずれかの始状態を完全に包含している規則）がある場合，あるいは，2 規則の両方に優先して適用される他規則がある場合は，その状況では 2 規則は競合しないと判定し，競合候補から除外する．

上記 2 つの場合の他規則優先適用における条件判定と，適用可能規則の関係の定義を以下に示す．

[定義 4. 2] 他規則優先適用

$$(1) \quad (G \supseteq I S_o) \wedge E_i = E_j = E_o$$

$$\wedge ((I S_o \supset I S_i) \vee (I S_o \supset I S_j)) \rightarrow (R_o \wedge R_j) \vee (R_o \wedge R_i)$$

$$(2) \quad (G \supseteq I S_o) \wedge E_i = E_j = E_o$$

$$\wedge ((I S_o \supset I S_i) \wedge (I S_o \supset I S_j)) \rightarrow R_o$$

G : 2 規則 R_i, R_j から作成した競合候補を満足する状況,

$I S_o$: 他規則の始状態,

R_o : 他規則,

E_i, E_j : 2 規則 R_i, R_j のイベント,

E_o : 他規則 R_o のイベント,

$I S_i, I S_j$: 2 規則 R_i, R_j の始状態, とする.

(注) $\rightarrow$ の左側は規則適用の条件, $\rightarrow$ の右側は適用規則を示す.

規則適用規約 2 (第 2 章: 定義 2. 8 参照) により, 始状態の完全包含関係がある場合は, 包含している方の規則を優先適用するため, 上記の場合は, 検討している 2 規則は競合しないと判定できる. □

手順 3 : 設計者による検証

手順 2 により機械検出した競合候補を参考にし, 設計者が解釈実行システム (第 3 章参照) を用いて, 実際に競合するかどうかをシュミレーションしながら検証する.

4. 4 競合候補検出

2 規則の抽出から、競合候補を検出するまでの処理の流れを図 4-3 に示す。また、図 4-4 に、図 4-1 の 2 規則が競合すると判定されるまでのデータフローを示す。図 4-3 に添って競合候補検出アルゴリズムを図 4-4 の具体例を参照しながら説明する。

(処理 1) 2 規則抽出：イベントの同じ 2 規則の組み合わせを抽出する、抽出する 2 規則がなくなれば処理を終了させる。抽出した 2 規則について以下の処理を行う。図 4-4 では図 4-1 の 2 規則が抽出された場合を示す。

(処理 2) 規則優先検査：S T R 手法では、ある状況において適用可能な規則が複数ある場合、規則の条件（始状態）が他規則の条件を完全に包含している時は、前者を優先して適用する。2 規則を比較して、この規則優先関係にある場合は、2 規則は、競合を起こさないので、処理 1 へ戻り、他の 2 規則の抽出をおこなう。（図 4-4：2 規則間には上記の包含関係はないので処理 3 へ進む）

(処理 3) 競合候補作成：2 規則の条件（始状態）から、競合する可能性のある状況を作成する。

(1) 端末変数の区分：2 規則の端末変数を区分するため、小文字に置き換え、添え字 1 を 1 番目の規則、2 を 2 番目の規則へ付加する。（図 4-4：2 規則から端末変数テーブルを作成する）

(2) イベント制約の付加：競合がおきる必要条件のひとつは、イベントが等しいことである。そこでイベントの端末変数を同一化し、その制約を付加する。（図 4-4：「 $\text{dial}(a1, b1)$ 」 = 「 $\text{dial}(a2, b2)$ 」よりイベント制約（ $a1=a2 \rightarrow a, b1=b2 \rightarrow b$ ）が

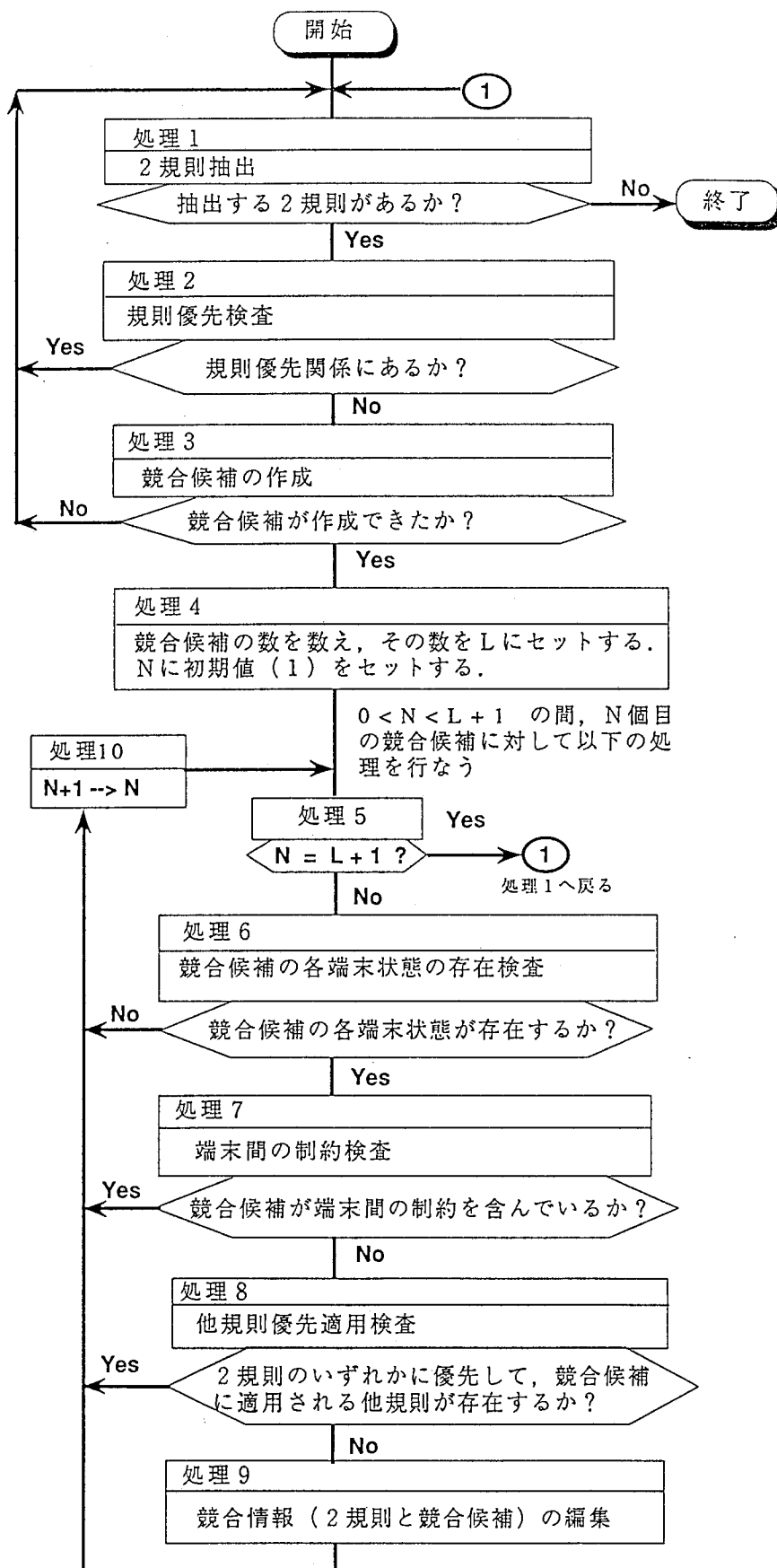


図 4-3 競合候補検出処理の流れ

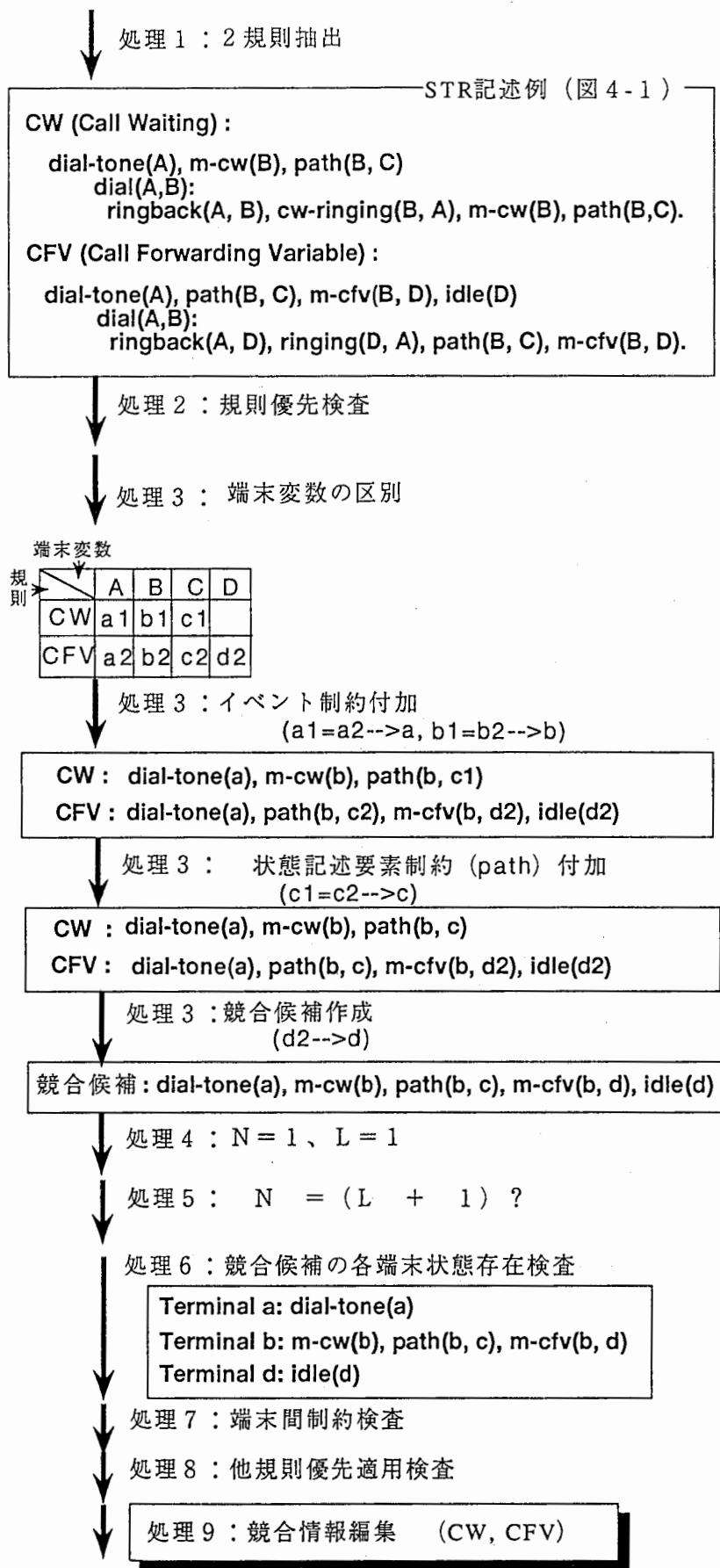


図 4-4 競合候補検出データフロー

付加される)

(3) 状態要素制約の付加：同一端末で重複しない状態要素の場合，その状態要素の端末変数を同一化し，その制約を加える．（図 4-4：「 $\text{path}(b, c1)$ 」=「 $\text{path}(b, c2)$ 」より状態記述要素制約 ($c1=c2 \rightarrow c$) が付加される)

(4) 制約のない端末変数は組み合わせを行う．その組み合わせにより複数の競合候補が作成される．（図 4-4：「 $d2$ 」は a, b, c 以外で場合の数は 1）

(5) 端末変数の矛盾検査：一つの規則の端末変数，例えば「 A, B 」は別な端末を意味している．上記，端末変数の操作により「 A, B 」が同一となる場合は，競合候補から除外する．競合候補がひとつも作成できないときは，処理 1 へ戻る．（図 4-4：競合候補があるので処理 4 へ進む）

（処理 4）処理制御数の設定：競合候補の数を数え， L へセットする． N の初期値 1 をセットする． $0 < N < (L + 1)$ の間以下の処理を， N 番目の競合候補に対して行なう．（図 4-4： $L = 1, N = 1$ ）

（処理 5）競合候補検査完了判定： $N = (L + 1)$ のとき，2 規則から構成したすべての競合候補に対する検査は完了したので，処理 1 へ戻り次の 2 規則の抽出を行なう．（図 4-4： $N = L$ なので処理 6 へ進む）

（処理 6）競合候補の各端末の存在検査：競合候補の各端末が，実状態集合に存在しているかどうか調べる．すべての端末状態が存在している場合は，処理 7 へ進み，そうでなければ，競合候補は除外され，処理 10 へ進む．（図 4-4：各端末状態は実状態集合に存在しているので，処理 7 へ進む）

（処理 7）端末状態間の制約検査：競合候補があらかじめ設定しておいた端末間制約知識（例：「path(X, Y), idle(Y)」）を含む場合、競合候補は除外され、処理 10 へ進む。端末間制約を含まない場合は、処理 8 へ進む。（図 4-4：競合候補は端末間状態制約を含まないので、処理 8 へ進む）

（処理 8）他規則優先適用検査：競合候補に対して、2 規則のいずれかの規則に優先して適用される規則がある場合は、その競合候補では 2 規則は競合しないと判定する。その場合は、処理 10 へ進む。そうでなければ、処理 9 へ進む。（図 4-4：競合候補に対して、他規則の優先適用はないので処理 9 へ進む）

（処理 9）競合情報編集：2 規則と競合状況（競合候補を含む状況）をファイルへ書き込む。処理 10 へ進む。（図 4-4：図 4-1 の 2 規則と競合候補をファイルに書き込む）

（処理 10）制御変数 N 更新：次の競合候補の検査の為に、N に 1 を加算する。処理 5 へ戻る。

試作システムにおける競合候補の検出表示画面を図 4-5 に示す。左上のサブ画面には、競合する 2 規則の組み合わせが表示される。設計者が指定した 2 規則の組み合わせ毎に、左下のサブ画面には、競合する状況（2 規則が同時に適用可能となる状況）が表示され、右上、右下のサブ画面にはそれぞれの規則適用の結果、推移する状況が表示される。

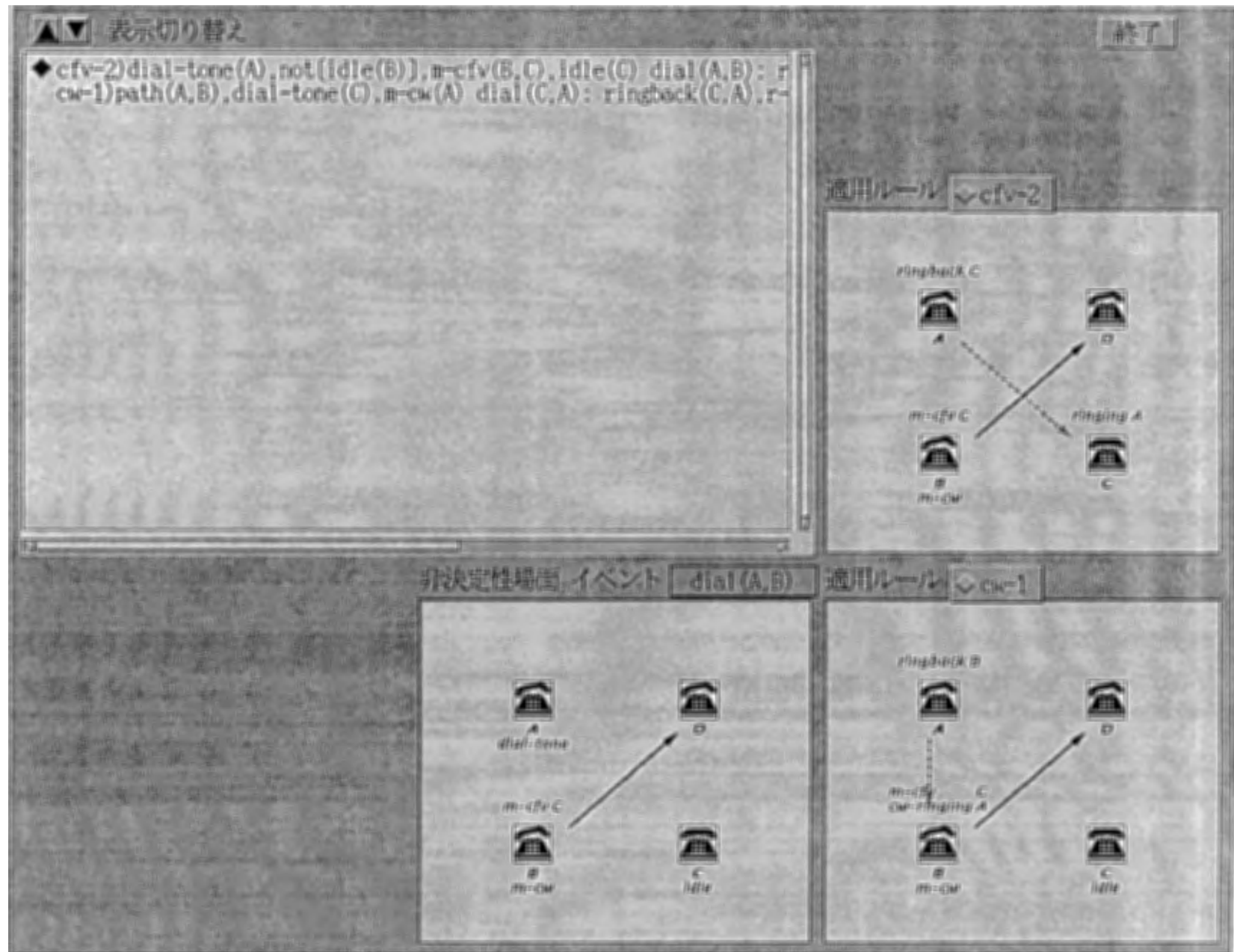


図 4 - 5 競合検出画面例

4. 5 競合検出実験と評価

基本電話 (POTS) , 話中着信 (CW) , 着信転送 (CFV) , 3 者通話 (3WC) , CCBS(Call Completion of Busy Subscriber)の 5 サービス記述を対象に検出実験を行った。個々のサービスは、単体動作は検証済みで、他サービスとの相互作用設計は行っていないものとした。その結果を表 4-1 に示す。表 4-1 は、サービスの組み合わせ毎の「競合する 2 規則の組み合わせ数」 (図 4-3, 処理 9) と「抽出した 2 規則の組み合わせ数」 (図 4-3, 処理 1) を示す。前者は機械検出した競合する 2 規則の組み合わせ数を意味し、後者は人手で調べる必要のある 2 規則の組み合わせ数を意味する。「POTS+CW」のサービス合成では競合はなく、「POTS+CW+CFV」のサービス合成では 2 つの 2 規則の組み合わせが競合すると判定された。「POTS+CW+CFV+3WC」のサービス合成では、「POTS+CW+CFV」のサービス合成と同じ内容の 2 つの 2 規則の組み合わせが競合すると判定された。「POTS+CW+CFV+3WC+CCBS」のサービス合成では、上記の検出結果を含む 5 つの 2 規則の組み合わせが競合すると判定された。

「POTS+CW+CFV+3WC+CCBS」のサービス合成の場合、「競合する 2 規則の組み合わせ数」は 5, 「抽出した 2 規則の組み合わせ数」は 799 である。図 4-6 に検討対象の絞り込み効果を示す。本来、人手で検討すべき 799 の 2 規則の組み合わせ中、794 は競合しないと判定され、5 つが競合する可能性があると判定された。これは、2 規則の競合検討対象が 1 % (5 / 799) 以下に絞り込まれていることを示している。

表 4 - 1 競合検出結果 1

サービスの 組み合わせ 2 規則の組み合 わせ比較	競合する 2 規則の 組み合わせ数	抽出した 2 規則の 組み合わせ数
POTS+CW	0	146
POTS+CW+CFV	2	259
POTS+CW+CFV +3WC	2	726
POTS+CW+CFV +3WC+CCBS	5	799

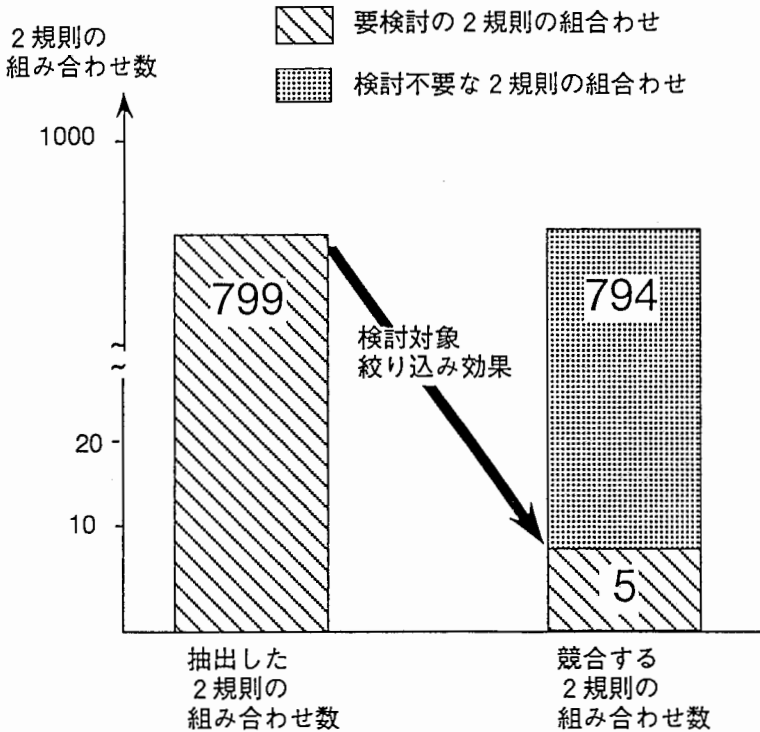


図 4 - 6 競合検出の絞り込効果

「POTS+CW+CFV+3WC+CCBS」のサービス合成により検出された、5つの競合する可能性があるとして判定された2規則の組み合わせと競合するときの状況を表4-2に示す。この情報をもとに、解釈実行システムを用いて、検討したところ、最後のケースは実際にはありえない状況であり競合はしないと判定された。つまり、機械検出した競合する可能性のある2規則の組み合わせは5つであり、そのなかの4つが実際に競合するものであった。

実験結果から、提案した競合検出手法は機械的に検討対象を絞り込む効果が大いことを示した。また、解釈実行システムを用いて、機械検出した競合情報を検証することで最終的な競合検出ができた。

表 4-2 競合検出結果 2（2 規則と競合状況）

(POTS+CW+CFV+3WC+CCBS)

競合する 2 規則	競合状況例
cw-1) dial-tone(A), path(B, C), m-cw(B) dial(A, B): ringback(A, B), cw-ringing(B, A), m-cw(A), cw(A), path(B, C). cfv-15) dial-tone(A), path(B, C), m-cfv(B, C), not[idle(C)] dial(A, B): busy(A), m-cfv(B, C).	dial-tone(A), m-cw(B), path(B, C), m-cfv(B, D)
cw-1) dial-tone(A), path(B, C), m-cw(B) dial(A, B): ringback(A, B), cw-ringing(B, A), m-cw(B), cw(B), path(B, C). cfv-6) dial-tone(A), not[idle(B)], m-cfv(B, C), idle(C) dial(A, B): ringback(A, C), ringing(C, A), not[idle(B)], m-cfv(B, C).	dial-tone(A), path(B, C), m-cw(B), m-cfv(B, C), idle(C)
3wc-11) path-passive(A, B), m-3wc(A) onhook(B): empty. ccbs-3) m-CCBSed(A, B) onhook(B): empty.	path-passive(A, B), m-3wc(A), m-cw(A), m-cfv(A, D), m-CCBSed(A, B), path(A, C)
cw-6) path(A, C), cw(A), path-passive(A, B) onhook(B): path(A, C). ccbs-3) m-CCBSed(A, B) onhook(B): empty.	path-passive(A, B), cw(A), m-cw(A), path(A, C), m-cfv(A, B), m-CCBSed(A, B)
cw-15) cw-ringing(A, B) onhook(B): empty. ccbs-3) m-CCBSed(A, B) onhook(B): empty.	cw-ringing(A, B), busy(A), m-CCBSed(A, B)

4. 6 考察

(1) 競合検出の網羅性の保証

提案した競合検出方法は、2規則の始状態から2規則が同時適用される可能性のある状況（競合候補）を作成し、その状況が実際に在りえる状況かどうかを判定する。在りえない場合は、競合候補から除外する。また、規則適用規約2を考慮して、2規則が優先関係にある場合は競合しないと判定する。競合候補に適用可能な他規則があり、他規則が2規則のいずれか、あるいは、両方に優先適用される場合は競合候補から除外する。つまり、本手法の特徴は、2規則が明らかに競合しない条件を判定基準に用い、その判定基準に該当する場合、競合候補から除外する方法である。この判定を通過した競合候補が在る場合、その競合候補の2規則は競合する可能性があると判定する。

2規則の競合する可能性のある状況（競合候補）は端末変数の組み合わせにより、網羅的に作成される。その競合候補の中より、競合しないものを取り除いて、残ったものを競合候補として提示しているので、結果は競合する可能性のある2規則の組み合わせを網羅している。

(2) 競合検出の絞り込み効果

実験結果から、「POTS+CW+CFV+3WC+CCBS」のサービス合成の場合、本来、人手で検討すべき2規則の組み合わせ数に対して、機械的に検出されたものは、1%（5/799）以下に絞り込まれており、提案手法の絞り込み効果は大きく、設計支援として有効である。

(3) 競合解消

競合解消のためには、競合を起こしている事象と競合を引き起こす原因の解析を行い、真の原因を解消するように規定を明確にする必要がある。競合解消のケースとして、以下の場合が考えられる。

1) 事象の起きている場所の規定を明確にする

競合状況において、何れかの動作規定を優先するか、両方の動作規定を満足するような対応を行う。

2) 事象を引き起こす要因となっている場所の規定を明確にする

この場合は、以下の規定の追加の可能性を調べる必要がある。

2-1) 競合を引き起こすサービスに加入してはいけない。

2-2) 競合を引き起こすサービスに加入することは許容するが、一方のサービスが動作しているときは、他方のサービスは起動してはいけない。

2-3) 合成サービス動作中に問題があり、その問題を解消しなければならない。

上記の解析支援のためには、競合する状況に到達した理由を解析する必要があり、初期状態から解析対象の状態に至る推移を設計者に提示する機能（第5章参照）と解釈実行システムを用いることにより解析支援を行う。

解消を行う方法として、規則の優先適用の規約を利用する。規則適用規約2（第2章：定義2.8参照）により、規則の始状態の完全包含関係がある場合は、包含している方の規則が優先適用される。この規約に基づいて、2規則の一方の規則の始状態を他方の規則に対して優先して適用されるように状態記述要素を追加する。

[事例による競合解消の説明]

図 4 - 2 の話中着信規則と着信転送規則は，図 4 - 3 に示す状況で競合する．この場合，着信転送規定を優先することにした場合（Bellcoreのサービス仕様 [6] では，着信転送が優先になっている）の修正例を示す．始状態は 2 規則の始状態を完全に包含し，イベントは 2 規則と等しく，次状態は目的に応じた（競合する規則の一方を優先させる）規則を追加すれば，2 規則の競合状況（図 4 - 3 -(A)）に対して，追加規則が優先適用されて，2 規則の競合は解消される（本章；定義 4 . 2 の他規則優先適用を参照）．この場合，競合候補（2 規則の始状態から端末変数の可能な組み合わせを作成し，それぞれの端末変数の組み合わせごとに各端末毎の状態記述要素の和集合を作成したもの）が，2 規則の始状態を包含しており，そのまま追加規則の始状態として採用できる．追加規則を図 4 - 7 に示す．規則の優先適用の規約を用いれば，個別のサービス規定は修正せず，サービス合成をおこなった場合の規定を追加していくことにより，個別サービス設計の手戻りは，軽減される効果がある． □

話中着信 (CW:Call Waiting)

dial-tone(A), m-cw(B), path(B, C)

dial(A,B):

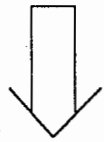
ringback(A, B), cw-ringing(B, A), m-cw(B), path(B, C).

着信転送 (CFV :Call Forwarding Variable) :

dial-tone(A), path(B, C), m-cfv(B, D), idle(D)

dial(A,B):

ringback(A, D), ringing(D, A), path(B, C), m-cfv(B, D).



規則追加

着信転送優先規則

dial-tone(A), path(B, C), m-cfv(B, D), m-cw(B), idle(D)

dial(A,B):

ringback(A, D), ringing(D, A), path(B, C), m-cfv(B, D), m-cw(B).

図 4 - 7 競合解消のための規則追加例

4. 7 第4章のまとめ

相互作用に含まれる非決定的動作矛盾の検出の問題は、STR手法 [1,2] を用いることによって規則競合の問題として扱うことができる。本章では、イベントの同じ2規則の組み合わせを比較して、2規則が同時に適用される可能性のある状況を検出し、設計者に提示する方法を示した[3-5]。実験結果から、提案した競合検出手法は、機械的に検討対象を絞り込む効果が大きく（POTS+CW+CFV+3WC+CCBSでは1%以下）設計支援に有効であることを示した。また、検出結果は、競合する可能性のある2規則の組み合わせを網羅していることを示した。また、競合解消の基本的な考え方と解消事例を示した。

第4章の参考文献

- [1] 原田良雄, 平川豊, 竹中豊文, 門田充弘: サービス仕様の自動生成に関する考察, 情処第39回大会, 5S-5 (1989).
- [2] Hirakawa, Y., Harada, Y. and Takenaka, T.: 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 (1990).
- [3] Harada, Y., Hirakawa, Y., Takenaka, T., Terashima N.: A Conflict Detection Support Method for Telecommunication Service Descriptions, 信学論文誌, Vol. E75-B, No. 10, pp. 986-997 (1992).
- [4] 原田良雄, 平川豊, 竹中豊文: 通信サービス動作競合検出方法, 情処研報, Vol. 91, No. 83, pp. 87-92 (1991).
- [5] 原田良雄, 平川豊, 竹中豊文: 通信サービス競合検出システムの試作と評価, 信学技法, Vol. 92, No. 75, pp. 13-18 (1992).
- [6] Bell Communication Research: Technical Reference, Issue 2 (1987).

第5章 通信サービス相互作用に含まれる意味的矛盾 動作検出支援方式

5.1 第5章の概要

通信サービスを既存サービスに追加する際には、追加サービスと既存サービスとの相互作用（サービスインターアクション）を解析し、サービス仕様間に生じる矛盾を解消し、サービス全体として矛盾のない仕様を作成する必要がある [4-8]。第3章では、通信サービス相互作用の検証項目として、状態推移モデルの構造的な問題（デッドロック、ループ、非決定性、等）と、構造上は問題がないが状態および推移の意味的な問題があるものに分類した。前者が解消できれば、状態推移モデル上は、端末は初期状態から始まり、途中で止まることなく、初期状態に戻る動作は保証される。しかしながら、個別サービスあるいは合成サービスの意味として矛盾する状態あるいは推移が含まれることがある。両者の観点から検証して、はじめて、サービス合成仕様として満足したものとなる。

本章では、相互作用に含まれる意味的矛盾動作に着目し、意味的矛盾動作の検出支援を行う方式を提案する。通信サービス動作は、STR手法 [2,3] を用いて形式的な規則形式で記述されていることを前提とする。提案方式では、矛盾状態知識を用い、サービス合成動作生成時に矛盾状態を判定し、初期状態から矛盾状態に至る状態推移を検出する。また、矛盾状態知識の獲得支援として、1) 状態記述要素の属性の整理を前提として、基本的な基準を与えることで矛盾状態を列挙する方法、2) サービス合成により生成される未定義な状態および推移を列挙することで、矛盾状態知識の獲得を支援する方法、3) サービス干渉の可能性のある状態および推移を列挙し、矛盾状態知識の獲得を支援する方法を示す。

提案方式をシステム試作し、実際に5つのサービスを記述し、サービスを組み合わせて、意味的矛盾動作の検出実験を行った。その結果より、矛盾状態の検討範囲としての絞り込み効果があり、検証作業の効率化が図れた。また、初期状態から矛盾状態に至る状態推移を示すことで、意味的矛盾動作の解析を効率的に行うことができた。

5. 2 意味的矛盾動作

個々のサービス仕様の設計は完了している2つのサービス、CFV (Call Forwarding Variable) サービスとCCBS (Completion of Call to Busy Subscriber) サービスが同時に稼働する環境を想定してみよう。CFVサービスとCCBSサービスの基本動作は以下に示す。

(1) CFV：着信転送を登録している端末に着信があると、1) 転送先がアイドル状態ならば着信を転送し、2) 転送先が空きでない状態ならば転送を行わずダイヤルした端末はビジー音が出る。

(2) CCBS：1) ダイヤル後、ビジー音が出ている時、CCBSを要求すると、ダイヤル先に要求がかかり、2) ダイヤル先の端末状態を監視し、相手端末がアイドル状態になると、自動的に呼び出しを行う。

この2つのサービスが組み合わさったときの動作を図5-1に示す。端末QがRに対してCFVを登録しており、通話中の時、端末Pが端末Qにダイヤルすると、CFV：2)の仕様により端末Pにはダイヤル後のビジー音 (busy-dial(P,Q)) が出る

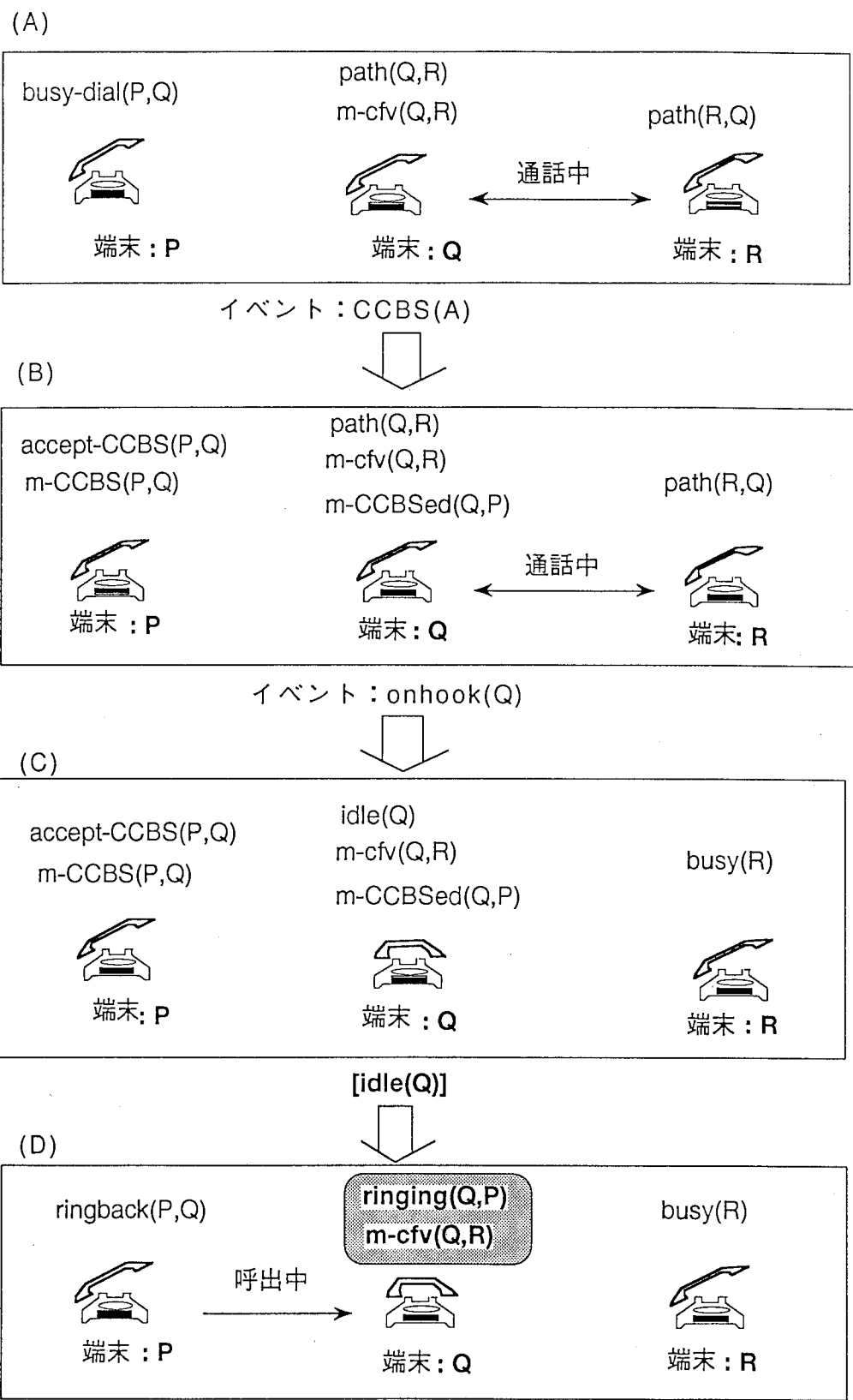


図 5-1 意味的矛盾動作例

(図 5-1-(A)) . この状態で、端末 P が C C B S を要求すると、C C B S : 1) の仕様より端末 Q に対して C C B S が設定される (端末 P : accept-CCBS(P,Q), m-CCBS(P,Q), 端末 Q : m-CCBSed(Q,P)) (図 5-1-(B)) . この状態で端末 Q がオンフックすると端末 Q はアイドル状態になり (図 5-1-(C)) , この瞬間、C C B S : 2) の仕様により、アイドル状態「idle(Q)」になったので自動的に端末 Q に呼び出しがかかる (図 5-1-(D)) .

C F V の単独のサービス仕様では、着信転送を登録している端末への着呼は必ず転送先に転送される為、着信転送登録端末自身が呼び出される状態 (呼出音表示) (図 5-1-(D) : 端末 B の状態) は C F V サービスからみると矛盾した状態となる.

[定義 5. 1] 意味的矛盾動作

あるサービスの視点からみて、意味的に矛盾した状態に至る動作を「意味的矛盾動作」と呼ぶ. □

5. 3 意味的矛盾動作検出方式の概要

意味的矛盾動作の検出手順 (図 5-2 の手順 1 ~ 4) の概要を以下に示す.

手順 1 : 意味的矛盾を判定するための知識 (状態記述要素の組み合わせ) を設定する.

[定義 5. 2] 矛盾状態知識

意味的に矛盾する状態記述要素の組み合わせを「矛盾状態知識」と呼ぶ. □

矛盾状態知識の獲得支援としては、1) 状態記述要素の属性の整理を前提として、

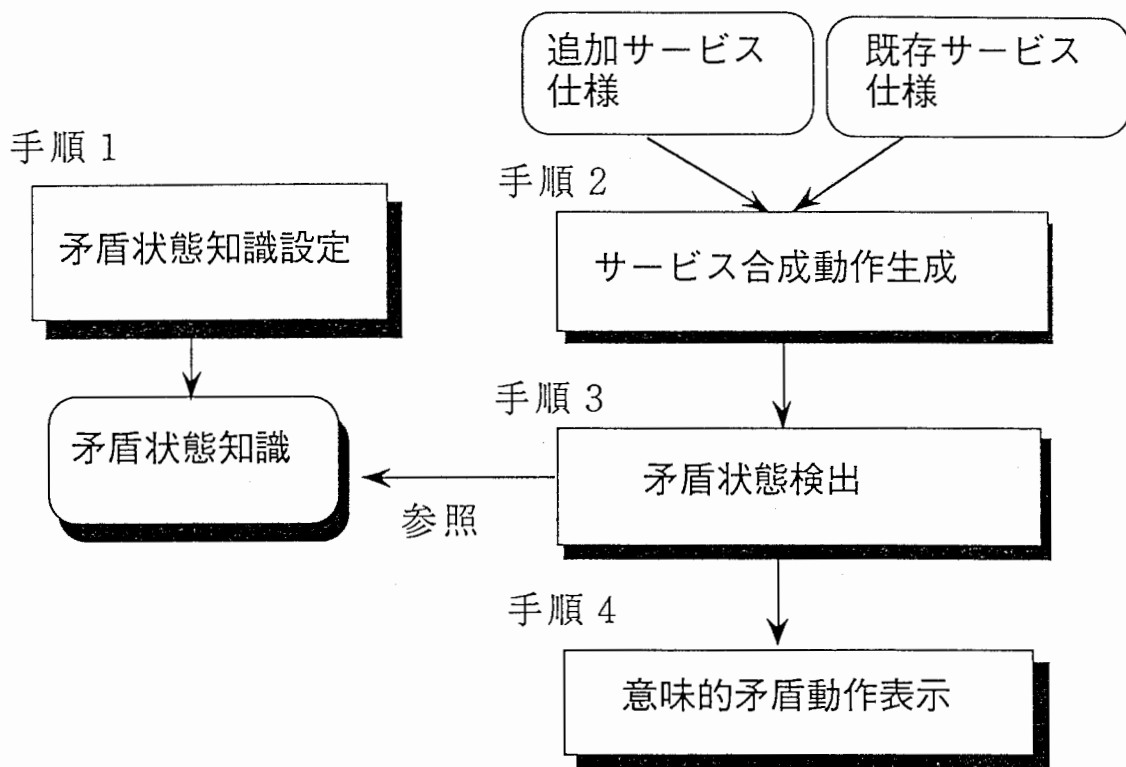


図 5-2 意味的矛盾動作の検出手順

基本的な基準を与えることで矛盾状態を列挙する方法，2) サービス合成により生成される未定義な状態および推移を列挙することで，矛盾状態知識の獲得を支援する方法，3) サービス干渉の可能性のある状態および推移を列挙し，矛盾状態知識の推移を支援する方法を用いる．

手順2：STR手法で記述されたサービス仕様からサービス合成動作（状態推移動作の集合）を生成する．

手順3：矛盾状態知識を部分集合に持つ状態を検出する．

[定義 5. 3] 矛盾状態

矛盾状態知識を部分集合にもつ状態を「矛盾状態」と呼ぶ． □

手順4：初期状態から矛盾状態へ至る状態推移を設計者に提示する．

[定義 5. 4] 初期状態

初期状態とは，端末状態がアイドル状態記述要素「idle(P)」だけの状態をいう． □

5. 4 矛盾状態知識の設定

5. 4. 1 矛盾状態知識

矛盾状態知識は，状態記述要素の組み合わせで表現する．

[定義 5. 5] 矛盾状態知識表現

P_i を状態記述要素とすると，矛盾状態知識は，以下のように状態記述要素の集合で表現できる．

矛盾状態知識 = $\{P_0, \dots, P_n\}$ □

矛盾状態知識の例とその意味を以下に示す.

[矛盾状態知識の具体例]

例 1) m-cfv(A, B), ringing(A, C)

端末 A が端末 B に着信転送登録をしていることを示す状態記述要素 (m-cfv(A, B)) と端末 A に端末 C から呼びだしがかかっていることを示す状態記述要素 (ringing(A, C)) が同時に端末 A の状態に含まれることは矛盾状態である.

例 2) idle(A), path(A, B)

端末 A が初期状態を示す状態記述要素 (idle(A)) と端末 A が端末 B と通話中を示す状態記述要素 (path(A, B)) が同時に端末 A の状態に含まれることは矛盾状態である.

例 3) idle(A), path(B, A)

端末 A が初期状態を示す状態記述要素 (idle(A)) と, 端末 B が端末 A と通話中を示す状態記述要素 (path(B, A)) が, 端末 A, 端末 B の状態に同時に含まれることは矛盾状態である.

例 4) path(A, B), path-passive(A, B)

端末 A が端末 B と通話中を示す状態記述要素 (path(A, B)) と, 端末 A が端末 B を保留していることを示す状態記述要素 (path-passive(A, B)) が, 同時に端末 A の状態に含まれることは矛盾状態である.

[矛盾状態知識の具体例終]

5. 4. 2 矛盾状態知識の獲得支援

意味的に矛盾する状態知識の獲得支援として, 1) 状態記述要素の属性の整理を前提として, 基本的な基準を与えることで矛盾状態を列挙する方法, 2) サービス合成により生成される未定義な状態および推移を列挙することで, 矛盾状態知識の獲得を

支援する方法、3) サービス干渉の可能性のある状態および推移を列挙し、矛盾状態知識の獲得を支援する方法を以下に示す。

5. 4. 2. 1 状態記述要素の属性に基づく知識獲得支援

状態記述要素の属性を整理しておけば、基本的な基準を与えることで矛盾する状態記述要素の組み合わせを機械的に列挙することができる。

〔具体例〕状態記述要素の属性の整理例を以下に示す。

例1) アイドルとは、空状態を意味し、アイドルを示す状態記述要素 (`idle(A)`) のみで構成されるか、アイドルを示す状態記述要素 (`idle(A)`) とサービス契約等を示すメモリ関連の状態記述要素 (「m-」で始まる状態記述要素) で構成される。前者を初期状態と呼んで区別する。

例2) 音を表示する状態記述要素は、`ringback(A,B)`, `ringing(A, B)`, 等である。

例3) 通話に関する状態記述要素は、`path(A, B)`, `path-passive(A, B)`である。

上記の整理を前提とすると、以下の基準に基づき、機械的な矛盾状態の列挙が可能となる。

基準1) 同一端末において、アイドルとアイドル以外の組み合わせは矛盾状態である。

この基準により、`{idle(A), path(A, B)}` 等が列挙できる。

基準2) 同一端末において、同一状態記述要素の重複は矛盾状態である。

この基準により、`{idle(A), idle(A)}` 等が列挙できる。

基準3) 同一端末において、音に関する状態記述要素の重複は矛盾状態である。

この基準により、`{ringback(A, B), ringing(A, B)}` 等が列挙できる。

基準4) 同一端末間において、同一方向の通話に関する状態記述要素の重複は矛盾状態である。

この基準により, $\{\text{path}(A, B), \text{path-passive}(A, B)\}$ 等が列挙できる.

[具体例終]

このように, 状態記述要素の属性を整理しておき, 新規サービス登録時にも, 必ずこの手順を踏むことを前提とすれば, 設計者の熟練度に関係なく, 通信サービスとして基本的に矛盾する状態記述要素の組み合わせを機械的に列挙することができる.

5. 4. 2. 2 未定義状態・推移からの知識獲得支援

サービス合成により生じた, 新規サービスにも既存サービスにも存在しない状態および状態推移の列挙を行い, 設計者の矛盾状態の検出を促す. 既存サービスの状態集合を $S\text{-exist}$, 新規サービスの状態集合を $S\text{-new}$, サービスを合成することにより新たに生じる (未定義な) 状態の集合を $S\text{-added}$ とすると, 未定義な推移は, $S\text{-new}$ から $S\text{-added}$ への推移, $S\text{-added}$ から $S\text{-added}$ への推移, $S\text{-added}$ から $S\text{-exist}$ への推移を検出することにより網羅できる. 機械的に検出した未定義な状態, および推移から, 矛盾状態知識の獲得支援を行う例を以下に示す.

[具体例] 着信スクリーニングサービス (TCS) と着信転送サービス (CFV) の組み合わせにおいて発生する, 意味的に矛盾する例を以下に示す. TCS は, ダイヤルを受けたとき, スクリーニングリストを調べ, あらかじめ登録している端末の着信をブロックするサービスである. CFV は着信を転送するサービスである (詳細は 5. 2 節の例の説明を参照). 端末 A がスクリーニングリストに端末 B を登録していることを示す状態記述要素を「 $m\text{-tcs}(A, B)$ 」と表す. このサービスを以下のように規則記述した場合を想定する.

$tcs\text{-}1) \text{ idle}(A), m\text{-tcs}(A, B), \text{dial-tone}(B) \quad \text{dial}(B, A):$

$\text{idle}(A), m\text{-tcs}(A, B), \text{busy-dial}(A, B).$

この規則はダイヤルを行った端末 B が TCS により, ブロックされて, ビジー状態にな

ることを示す.

TCSとCFVの組み合わせとして以下の状況を想定する.

端末P: $\{m\text{-}tcs(P, Q), idle(P)\}$

TCSとしてスクリーニングリストに端末Qを登録しており, アイドル状態である.

端末R: $\{m\text{-}cfv(R, P), idle(R)\}$

CFVとして, 着信転送先に端末Pを設定しているおり ($m\text{-}cfv(R, P)$), アイドル状態である.

この状況で端末Qが端末Rにダイヤルを行うと, 着信転送が起動され, 端末QからPへの呼び出し状態となる. 端末Pが応答すると, 通話状態となる. 端末Pからみると, 端末Qからの呼び出しをブロックしようとしているにもかかわらず, CFVにより, 着信してしまい, 意味的に矛盾する状態となる.

端末Pがスクリーニングリストに端末Qを登録しているにもかかわらず, Qから呼び出されている以下の状態は, CFVとTCSを合成して生じる状態であり, 未定義状態として機械的に検出できる.

端末P+Q: $\{ringing(P, Q), m\text{-}tcs(P, Q), ringback(Q, P)\}$

未定義状態を調べることにより, 上記の状態が意味的に矛盾する状態と判定することができ, CFV以外のサービスとの合成でも現われる可能性があるので, 上記の状態を矛盾状態知識として設定する. [具体例終]

設計者は未定義な状態および推移の妥当性を確認する必要がある, 検証中に検出した意味的に矛盾する未定義な状態を, 意味的に矛盾する知識として設定すればよい.

図2-(D)の端末Qの矛盾した状態は, この方法により獲得した矛盾状態知識を用いることで検出が可能となる.

5. 4. 2. 3 サービス干渉の可能性からの知識獲得支援

サービス干渉において、個別サービスの受ける影響としては、1) 現われるべき推移がでなくなる、2) 推移の順番が変わる、等がある。これらについては、以下の方法を用いて検証支援、あるいは、矛盾知識の獲得支援を行う。

未定義な状態に対して、どのサービスで生成された状態かを識別する情報を付加し、状態に付加されているサービスの規則が適用されないか、あるいは、その状態に対して適用可能な規則が適用されていない（全く、規則が適用されない場合はデッドロックである）、等の状態および推移を列挙し、設計者の検討を促す。

サービス干渉の可能性検出からの知識獲得支援について以下に例を示す。

[具体例] いずれの端末も基本電話サービス (POTS) に加入しているので、前述の未定義な状態推移の例で用いた TCS と CFV の組み合わせを端末ごとに示すと、端末P=POTS+TCS、端末Q=POTS、端末R=POTS+CFVである。端末RのCFVの干渉による端末Pの状態変化と端末Pのサービスの規則適用との関係で、端末Pのサービス推移が影響を受けることがわかる。

CFV適用前の端末Pの状態 (POTS+TCS) :

端末P = {idle(P), m-tcs(P, Q)}

CFV適用後の端末Pの状態 (POTS+TCS+CFV) :

端末P = {ringing(P, Q), m-tcs(P, Q)}

CFV適用前の端末Pの状態には、以下の2つの規則が適用可能性がある。

tcs-1) idle(A), m-tcs(A, B), dial-tone(B) dial(B, A):

idle(A), m-tcs(A, B), busy-dial(A, B).

pots-1) idle(A) offhook(A): dial-tone(A).

ところが、CFV適用後の端末Pの状態 (未定義な状態) には、上記の2規則は、適用

可能性はなくなり、他のPOTSの規則、つまり、応答すると、通話中になる規則が適用可能となる。この状況は、1) ある状態 ($\text{idle}(P), \text{m-tcs}(P, Q)$) で適用されるべき規則 ($\text{pots-1}, \text{tcs-1}$) が、他サービス(CFV)により干渉を受けた結果、適用されなくなり干渉を受けたことがわかる。また、2) たとえ、干渉を受けた結果の状態 ($\text{ringing}(P, Q), \text{m-tcs}(P, Q)$) に対して、干渉前に適用可能であったサービス (POTS, TCS) の他の規則が適用される可能性があったとしても、それらのサービスの推移として、ある状態から干渉を受けた結果の状態に直接推移することではなく (端末P = $\{\text{idle}(P), \text{m-tcs}(P, Q)\}$ に適用可能な規則 $\text{pots-1}, \text{tcs-1}$ を適用しても、端末P = $\{\text{ringing}(P, Q), \text{m-tcs}(P, Q)\}$ になる可能性はない) , 元のサービスの動作としては、想定していない推移になっていることがわかる。

状態の生成時に、1) 適用された規則のサービスの識別子 (pots-1 の規則では、「pots」に該当する) を付加し、2) その状態に適用可能な規則を付加し、3) 他サービスの適用により推移した状態に対して、推移前の状態に適用可能な規則が適用できなくなった場合を検出すれば、サービス干渉の可能性のある状態を列挙できる。設計者は、検出した状態および推移を検査し、意味的に矛盾する状態を知識として獲得する。 [具体例終]

設計者は、機械的に提示した検討範囲を分析し、サービスとして矛盾する状態を検出する。その後、あるサービスから見た、矛盾状態を知識として設定することで、他サービスとの組み合わせにより生じる、矛盾状態の検出ができ、知識の再利用ができる。

意味的に矛盾する状態知識の獲得支援として、1) 状態記述要素の属性の整理を前提として、基本的な基準を与えることで矛盾状態を列挙する方法、2) サービス合成により生成される未定義な状態および推移を列挙することで、矛盾状態知識の獲得を支援する方法、3) サービス干渉の可能性のある状態および推移を列挙し、矛盾状態

知識の獲得を支援する方法を示した。合成状態として、個別サービス動作が保証されたとしても、複数サービスの適用により生成された状態は、状態記述要素の組み合わせとして、妥当かどうかの問題は残り、上記の組み合わせにより意味的な矛盾状態知識を獲得する必要がある。

以下、サービスの合成状態推移の生成と獲得した矛盾状態知識を用いた矛盾動作検出について述べる。

5. 5 合成状態推移生成

状態推移生成の方法は、第3章において詳細に示したので、本章では割愛する。以下では、状態推移および状態管理情報が生成されているものとして、各処理を説明する。

5. 6 矛盾状態知識検査

規則を適用した結果の次状態の中に、矛盾状態知識に該当する記述要素が在る場合は、[定義5. 3]により、次状態を「矛盾状態」と判定する。

例えば、CCBSとCFVとの合成状態推移として、表5-1に示す状態推移表が生成された場合、次状態の「 $m\text{-cfv}(P, Q), \text{ringing}(P, R)$ 」は、 $P = A$ 、 $Q = B$ 、 $R = C$ とみなすことにより矛盾状態知識の例1：「 $m\text{-cfv}(A, B), \text{ringing}(A, C)$ 」に該当し、この動作は矛盾状態と判定される。表5-1のイベント欄の「 $\text{idle}(P)$ 」は、状態をイベントとみなす疑似イベントと呼ばれるものである。この場合、状態を監視してアイドル状態「 $\text{idle}(P)$ 」になったとき、イベントと同様に規則を起動させる契機となる。

表 5-1 状態推移表

状態	他端末状態	イベント	規則	次状態
idle(P) m-cfv(P,Q) m-CCBSed(P,R)	accept-CCBS(R,P) m-CCBS(R,P)	[idle(P)]	ccbs-2)	m-cfv(P,Q) ringing(P,R) ringback(R,P)

5. 7 意味的矛盾動作表示

状態管理情報（第 3 章の [定義 3. 1] 参照）を用いて，初期状態から矛盾状態に至る状態推移を表示する．表示する手順を以下に示す．

手順 1：矛盾状態に推移するときの端末 P の状態をキーとして，状態管理情報を捉える．

手順 2：状態管理情報のひとつ前の状態番号をキーとして，状態管理情報を検索する．初期状態が見つかるまで，手順 2 の操作を繰り返す．

手順 3：探索した状態管理情報をもとに，初期状態から矛盾状態に至る状態推移を合成して表示する．

上記手順 1 ～ 3 の説明を具体例を用いて以下に示す。矛盾状態を検出したときの状態推移を図 5 - 3 に示す。「 S_{i+1} 」は端末 P の状態を示す。「 E_x 」はイベント、「 $T S_x$ 」は他端末状態、「 R_x 」は適用規則を示す。このとき、手順 1 により、端末 P の状態「 S_{i+1} 」をキーとして、状態管理情報を探索する。状態管理情報と探索経路を図 5 - 4 に示す。手順 1 で、状態管理情報「 K_{i+1} 」が検出される。次に、手順 2 で、ひとつ前の状態管理番号（ $P N = i$ ）をキーに状態管理情報「 K_i 」が検出される。同様に初期状態が検出されるまで手順 2 を繰り返す。手順 3 として、初期状態から検出した管理情報を辿って矛盾状態に至る状態推移を合成し表示する。図 5 - 3、図 5 - 4 の例から合成された状態推移を図 5 - 5 に示す。

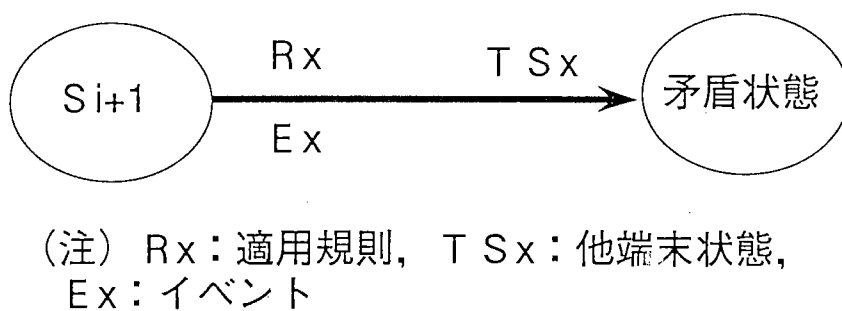
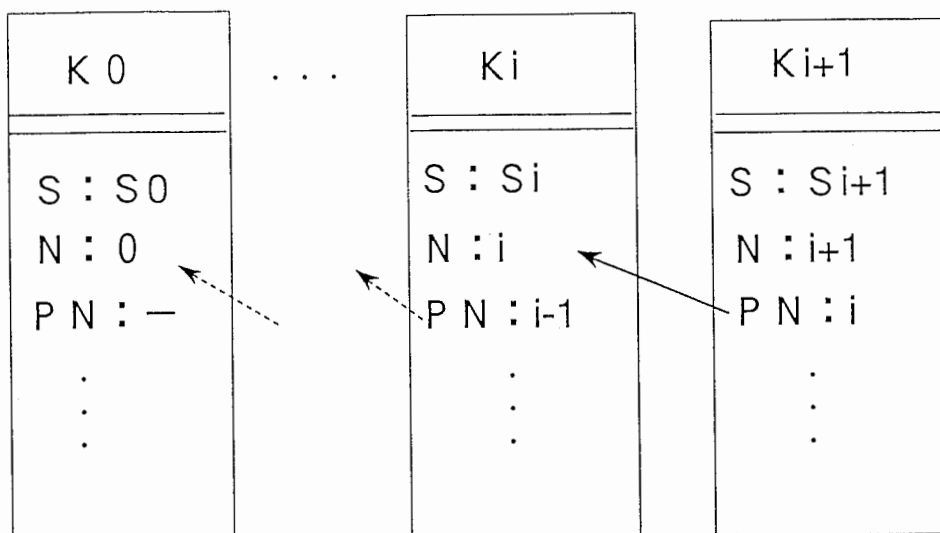


図 5 - 3 矛盾状態検出時の状態推移



(注) K_i : 状態管理情報, S : 状態, N : 状態番号,
 PN : ひとつ前の状態番号

図 5-4 状態管理情報と探索経路

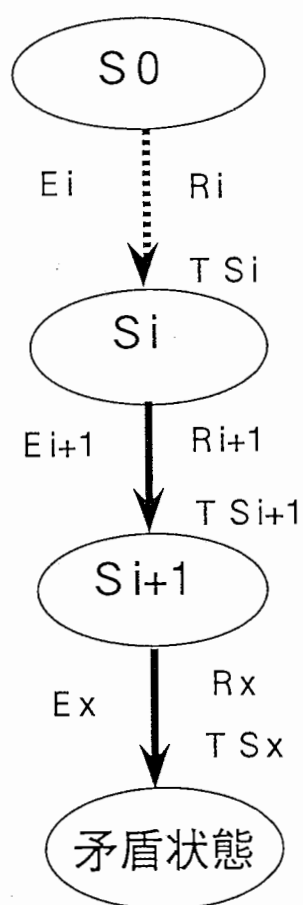


図 5-5 意味的矛盾動作表示例

5. 8 考察

本章で提案した矛盾動作検出方式を，Symbolics 3650 (Lisp) を用いて試作した．この試作システムを用いて，サービス相互作用に含まれる矛盾動作解析実験を行なった．通信サービスは，POTS(Plain Old Telephone Service)，話中着信(CW)，着信転送(CFV)，3者通話(3WC)，CCBS(Completion of Call to Busy Subscriber)の5サービスを対象に記述した．それらのサービスを組み合わせて，矛盾動作検出実験を行った．

(1) 矛盾状態検出

意味的矛盾知識（例：「m-cfv(A, B), ringing(A, C)」）に該当した矛盾状態の数と，表 5-1 で示すような状態推移表の次状態数を，サービス組み合わせごとに表 5-2 に示す．後者は，状態の妥当性確認を手作業で行う場合の状態数を示し，前者は機械的に検出した矛盾状態数を示す．後者に対する前者の比率は平均 12 % 以下（合計値比較： $(26 / 232) \times 100$ ）となっている．この数値は，検証対象の範囲が手作業に比べて 12 % 以下に絞り込まれ，かつ，限定されたことを示し，検証支援として有効であることを示す．

(2) 意味的矛盾動作検出

矛盾状態をみて，どういう状態推移で矛盾状態に至ったのか推測することは難しい．そのため，初期状態から矛盾状態に至る推移を機械的に合成し，設計者に提示し，原因解析を促進する．

初期状態から意味的矛盾知識（例：「m-cfv(A, B), ringing(A, C)」）に該当した矛盾状態への状態推移の例を図 5-6 に示す．図 5-6 の状態の左に付加している番号は状態番号を示す．状態 0) が初期状態，状態 5) が矛盾状態である．状態番号 0) か

表 5-2 意味の矛盾状態数

	意味の矛盾 状態検出数	状態推移表 次状態数
POTS+CFV+CCBS	8	80
POTS+CFV+ CW	10	72
POTS+CFV+3WC	8	82
合計	26	232

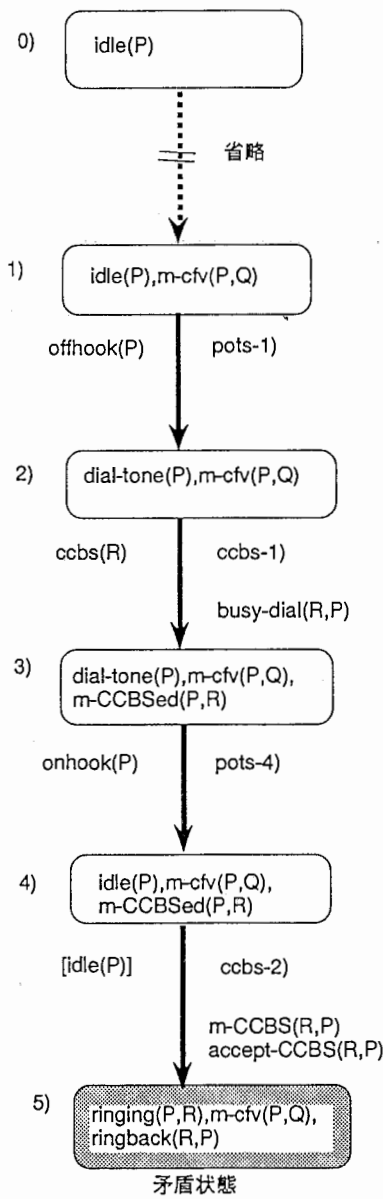


図 5-6 意味の矛盾動作検出例

ら 1) の推移は着信転送先を登録する動作であるが、説明上、省略する。

このように、状態推移は機械的に規則適用されて生成されるため、どの規則適用が矛盾状態を引き起こす真の原因であるかを解析するための支援に有効である。

(3) 意味的矛盾動作解消

状態は一つ前の状態を情報としてもっており、矛盾状態から初期状態へ遡ることが可能であり、修正対象範囲は限定できる。但し、推移のどこを修正するかは、設計者に委ねざるをえない。修正の方法は、規則の優先適用（定義 2. 8 参照）を前提とし、優先する規則に対して、イベントが等しく、かつ始状態を包含するようにし、次状態は、目的に応じて規定する。図 5-6 の状態をみると、状態 3) に本質的な問題がある。つまり、着信転送を登録している端末に C C B S 要求が設定されている。この状態に推移する規則の修正を行えば、意味的矛盾動作の解消が可能となる。S T R 手法では、規則適用規約 2 として規則優先適用を採用している。規則適用規約 2 を前提とすれば、既に C C B S の動作規定として図 5-7 の規則があるとき、図 5-8 の規則を追加すれば、図 5-6 の状態 3) の状態への推移は解消できる。図 5-7 の規則 ccbs-1) は、ダイヤル後のビジー音表示状態のとき C C B S 要求を行うと、ダイヤル先に無条件に C C B S 要求が設定される仕様であり、ccbs-2) は C C B S 要求後、要求先がアイドル状態になったとき（イベント：[idle(B)]）呼び出し状態に推移する仕様である。図 5-8 の追加規則では、ダイヤル後のビジー音表示状態のとき C C B S 要求を行うと、C C B S 要求先が着信転送を登録しているときは、C C B S 要求を着信転送登録先へ対して行う仕様である。この規則を追加することで、図 5-6 の状態 2) に対しては、追加規則が図 5-7 の ccbs-1) よりも優先的に適用され、状態 3) は現われなくなり、結果として、状態 5) の矛盾状態は解消される。

TCS サービスの特徴が CFV により損なわれる場合の（5. 4. 2. 2 節の例を参

ccbs-1) busy-dial(A,B)
 ccbs(A):
 accept-CCBS(A,B),m-CCBS(A,B),m-CCBSed(B,A).

ccbs-2) accept-CCBS(A,B),m-CCBS(A,B),m-CCBSed(B,A)
 [idle(B)]:
 ringback(A,B),ringing(B,A).

図 5 - 7 C C B S の動作規定

ccbs-3) busy-dial(A,B) ,m-cfv(B,C)
 ccbs(A):
 accpt-CCBS(A,C),m-CCBS(A,C),m-CCBSed(C,A).

図 5 - 8 C C B S の追加規則

照) 対応について以下に示す。まず、TCS、CFVの関連する規則を示し、次に解消規則を示す。

tcs-1) idle(A), m-tcs(A, B), dial-tone(B) dial(B, A):

idle(A), m-tcs(A, B), busy-dial(A, B).

cfv-1) idle(A), m-cfv(B, A), idle(B), dial-tone(C) dial(C, A):

ringing(A, C), m-cfv(B, A), pingring(B), ringback(C, A).

TCSの意図は、ダイヤル相手が、スクリーニングリストに登録されているときは、着信をブロックしたいということであるから、上記の着信転送の規則 (cfv-1) を元に、着信転送先がTCSに登録している場合には、発信端末をビジーにする以下の規則を新たに追加する。

twc+cfv-1)

idle(A), m-tcs(A, C), m-cfv(B, A), idle(B), dial-tone(C)

dial(C, A):

idle(A), m-tcs(A, C), m-cfv(B, A), idle(B), busy-dial(C, A).

この規則を追加することにより、規則「cfv-1」より優先適用され、5. 4. 2. 2節で述べた以下の状況において、端末QがRにダイヤルした場合は、端末Qはビジー状態となり、CFVによる端末Pの状態へのサービス干渉は解消される。

端末P : { m-tcs(P, Q), idle(P) }

端末Q : { dial-tone(Q) }

端末R : { m-cfv(R, P), idle(R) }

（４）矛盾状態知識の獲得支援

矛盾状態知識を設定できれば、実験データで示したとおり、知識の再利用により、検討範囲の絞り込み効果が大きく、検証支援に有効である。問題は、矛盾状態知識の獲得と設定をどのように支援できるかにかかっている。本章では、意味的に矛盾する状態知識の獲得支援として、１）状態記述要素の属性の整理を前提として、基本的な基準を与えることで矛盾状態を列挙する方法、２）サービス合成により生成される未定義な状態および推移を列挙することで、矛盾状態知識の獲得を支援する方法、３）サービス干渉の可能性のある状態および推移を列挙し、矛盾状態知識の獲得を支援する方法を示した（５．４．２節参照）。提案した方法では、設計者は機械的に列挙された状態の中から、状態の妥当性を検査した上で矛盾状態を選択すればよく、矛盾状態知識設定の労力を削減することができる。矛盾状態知識は、サービス追加の都度、逐次、追加していくものとする。既存サービスの視点からの矛盾状態は設定されており、新規サービス追加時においては、既存サービスの視点からの矛盾状態は検出できる。新規サービスの視点からの矛盾状態知識の獲得は、既存サービス全体と新規サービスの合成を行う前に、以下の手順を踏み段階的に獲得する。既存サービス中のひとつずつのサービスと新規サービスおよび基本サービス（POTS）の組み合わせにより、サービス合成を行い、サービス間の矛盾検出をあらかじめ行い、矛盾の解消をしつつ、矛盾状態知識の獲得を行う。この手順により、新規サービスの視点から、既存サービス中の任意のサービスとの間における意味的な矛盾状態知識は網羅的に把握できる。基本的な状態記述要素間の矛盾状態知識と既存サービスの視点から蓄積された矛盾状態知識、および新規サービスの視点から獲得した矛盾状態知識と合わせることで矛盾状態知識が設定できる。その後、既存サービス全体と新規サービスの合成を行い、矛盾状態知識を用いた矛盾動作の検出、サービス干渉の可能性の検出、および未定義な状態・推移検出を行う。合成サービスとして矛盾状態知識に該当する状態も

なく，サービス干渉の可能性もなければ，合成サービス動作中の個別サービスの動作は満足されていると判定できる．

5.9 第5章のまとめ

通信ソフトウェア開発全体の効率化を図る為には，設計の上流工程である，通信サービス設計段階で，サービス相互作用を解消したうえで，次工程（例えば，従来のSDL [9,10] 作成）へ進むことが重要である．本章では，STR手法 [2,3] で記述された動作規定を前提とし，通信サービス設計段階における相互作用設計支援の一環として，意味的矛盾動作検出方式を提案した [1] ．

提案手法のシステムの試作を行い，実際の5つのサービスを対象に実験を行った．その結果より，矛盾状態の検討範囲としての絞り込み効果があり，検証作業の効率化が図れることを示した．また，初期状態から矛盾状態に至る状態推移を示すことで，意味的矛盾動作の解析が効率的に行えることを示した．

意味的に矛盾する状態知識の獲得支援として，1) 状態記述要素の属性の整理を前提として，基本的な基準を与えることで矛盾状態を列挙する方法，2) サービス合成により生成される未定義な状態および推移を列挙することで，矛盾状態知識の獲得を支援する方法，3) サービス干渉の可能性のある状態および推移を列挙し，矛盾状態知識の獲得を支援する方法を示した．

今後の課題として以下の項目を検討している。

解釈実行システムとの連動：解釈実行システムとは，規則を解釈実行し，端末の動作をアニメーションで設計者に提示するシステムである．このシステムは，イベントの順番をファイルとして入力すると自動実行する機能を持つ．初期状態から矛盾状態へ到達する状態推移のイベント順をファイル化し，解釈実行システムへ渡すことで，視覚的な解析支援が可能となる．

第5章の参考文献

- [1] 原田良雄, 高見一正, 太田理, 寺島信義: 通信サービス相互作用に含まれる意味的矛盾動作検出方式, 情処論文誌, Vol. 35, NO. 8, pp. 1602-1613 (1994).
- [2] 原田良雄, 平川豊, 竹中豊文, 門田充弘: サービス仕様の自動生成に関する考察, 情処第39回大会, 5S-5 (1989).
- [3] Hirakawa, Y., Harada, Y. and Takenaka T.: 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 (1990).
- [4] Duran, M. J. and Visser, J.: International Standards for Intelligent Networks, IEEE Communication Magazine, Vol. 30, No. 2, pp. 34-42 (1992).
- [5] Collet, P. and Kung, R.: The Intelligent Network in France, IEEE Communication Magazine, Vol. 30, No. 2, pp. 82-89 (1992).
- [6] Bowen, T. F., et al., : The Feature Interaction Problem in Telecommunications Systems, Proc. of the 7th IEE Int. Conf. on Soft. Eng. for Telecom. Systems, pp. 59-62 (1989).
- [7] Cameron, E. J. and Velthuijsen, H.: Feature Interactions in Telecommunications Systems, IEEE Communication Magazine, Vol. 31, No. 8, pp. 18-23 (1993).
- [8] 長谷川晴朗: ペトリネットを用いた通信サービス仕様設計支援, 信学論文誌, Vol. J74-B-I, No. 6, pp. 445-455 (1991).
- [9] CCITT Recommendations Z. 100 to Z. 104 (1985).
- [10] 若原 恭: SDL言語の特質と処理系の現状と動向, 情報処理, Vol. 31, No. 1, pp. 23-34 (1990).

第6章 通信サービス課金仕様競合検出方式

6.1 第6章の概要

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・インターラクション）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。相互作用解消の問題は、1991年12月、CCITTによってインテリジェントネットワークのための技術的勧告として提示されたC S 1 (IN Capability Set 1) においても、今後の重要な課題のひとつに挙げられている [1-2]。通信サービスには、着信転送サービス、話中着信サービス等の接続動作に関するサービスだけではなく、フリーダイヤルサービス等の課金に関するサービスもあり、課金に関するサービスが追加されるとき、既存の課金仕様との矛盾解析が必要となる。

課金仕様の競合を機械的に検出するためには、課金仕様を形式的な記述方式で規定する必要があるが、従来、課金仕様を形式的に記述する方式の提案はなされていない。S T R手法 [3-4] は、サービスの接続動作に関する仕様を規則の集合として規定するものであり、課金仕様は記述対象の範囲外であり、S T R手法とは別に課金仕様を形式的に記述する手法が必要となる。本章では、まず、課金仕様を形式的な規則形式で記述する方式を提案する。次に、課金仕様の同時適用可能性を判定し、課金仕様要素間の競合検出を行う方式を提案する。サービスの接続動作に関する仕様はS T R手法 [3-4] を用いて規定するものとし、接続仕様と課金仕様はそれぞれ別々に規定し、両者の設計後の仕様を用いて課金仕様の同時適用性を判定する。ひとつの呼に対して複数の状態推移が発生し一意に決められない状態推移の非決定性検出方式は提

案している [5] . 課金仕様競合検出方式では, その基本的なアルゴリズムを応用し課金仕様の同時適用性判定に用いた.

実際に 6 つのサービスに対する課金仕様を記述し, 課金仕様の競合検出の机上実験結果を示す. また, 課金仕様記述方式の特徴として, 既存仕様の再利用による記述量の削減化が図られること, 競合解消においては既存仕様の修正の手戻りが少なく競合解消の効率化ができることを示す.

6. 2 課金仕様競合検出の課題

サービス相互作用検出方式の例として課金競合の例を取り扱っている論文がある [6] . この方式では, サービス相互作用は, 基本的に複数のサービスが同一あるいは同種類の要素を共有する場合に発生すると想定して, サービスの要素を分類し, サービスの組み合わせ毎に共通要素を洗いだし, 共通要素に対してロジックとよばれる処理を行う. その結果に対して, あらかじめ準備したルール (要素に対する規定をしたもの) が適用できるか否かによってサービス相互作用の検出を行う方法である. 簡易な方法であるが, サービス相互作用のひとつである状態推移の非決定的動作は必ずしも同一要素を共有する場合においてだけ生じるわけではなく, 非決定的動作の検出に対しては, この手法では充分に対応できない. また, サービスが増えた場合, サービス要素の組み合わせの設定自体難しくなる上に, 問題のないサービスの組み合わせに対してもサービス要素の組み合わせを行う作業が必要となる. ひとつの呼に対して複数サービスが同時適用可能なときに課金仕様競合の問題が生じるのであり, 同時適用の判定を追加して, 対象を絞り込んだ上でのサービス要素間の矛盾判定を行うといったアプローチが必要である.

6. 3 課金仕様競合

課金仕様競合として、課金端末競合と課金レート競合に着目し、課金仕様記述と課金仕様競合検出方式を提案する。本節では、課金端末競合と課金レート競合の定義とそれぞれの課金仕様競合例を示す。

[定義 6. 1]

サービスによって2つの端末間の課金対象端末が違い、一意に決まらないことを「課金端末競合」と呼ぶ。 □

[定義 6. 2]

サービスによって課金レートが違う場合、単独サービス仕様を組み合わせただけでは課金レートが違うため、どちらの課金レートにすべきか一意に決まらない。課金レートが違い一意に決まらないことを「課金レート競合」と呼ぶ。 □

[課金仕様競合例]

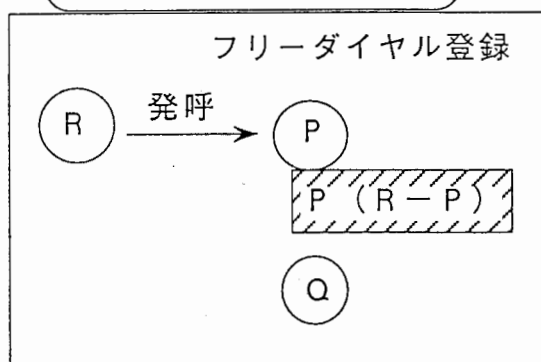
(1) 課金端末競合

着信転送サービスとフリーダイヤルサービスの課金仕様が以下のように規定されているとき、2つのサービスの組み合わせによる課金端末競合例を図6-1に示す。

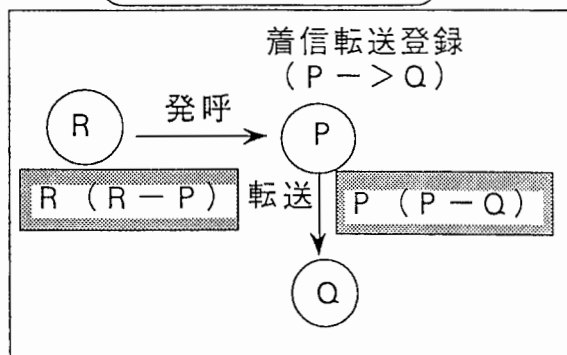
フリーダイヤルサービス：フリーダイヤルサービスの設定を行っている端末に着信があり、応答し通話を行うと、課金はフリーダイヤル設定端末に対して行われる。

着信転送サービス：着信転送サービスは、着信転送登録を行っている端末へ着信があると、その着信を着信転送登録先端末へ転送するサービスである。着信転送サービスの課金仕様では、発呼端末と着信転送登録端末間は発呼端末に課金され、着信転送登録端末と着信転送登録先端末間は着信転送登録端末に課金される。

(A) フリーダイヤルサービス

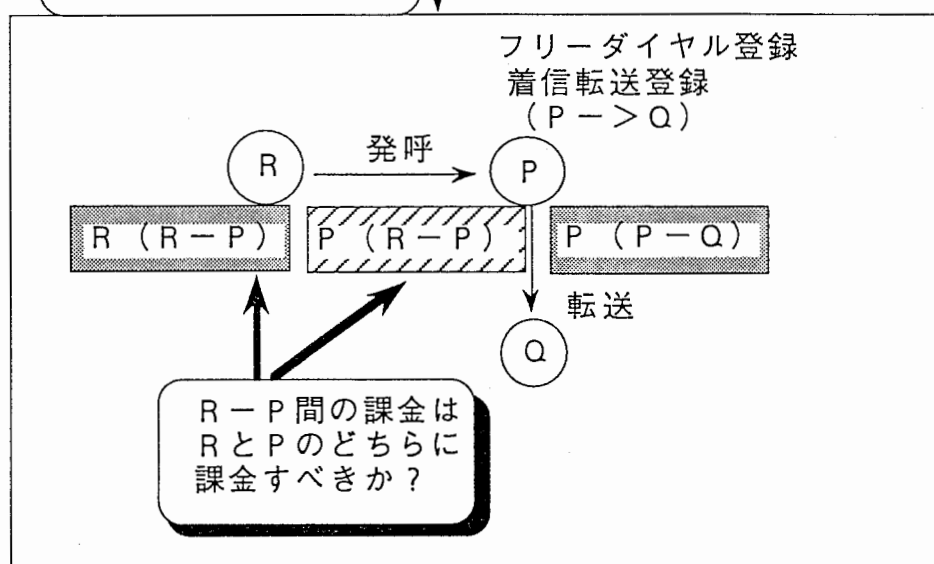


(B) 着信転送サービス

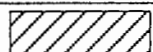


端末Pがフリーダイヤルと
着信転送を登録

(C) 課金端末競合発生状況



(注) $X (X-Y)$: $X-Y$ 間の課金端末は X であることを示す.

 : フリーダイヤルサービスの課金端末を示す.

 : 着信転送サービスの課金端末を示す.

P, Q, Rは端末を示す.

図6-1 課金端末競合例

図 6-1 (A)はフリーダイヤルサービスの課金端末を示し、発呼端末 R とフリーダイヤル設定端末 P 間は端末 P が課金対象となる。一方、図 6-1 (B)は着信転送サービスの課金端末を示し、発呼端末 R と着信転送登録端末 P 間は端末 R が課金対象となり、着信転送登録端末 P と着信転送登録先端末 Q 間は端末 P が課金対象となる。図 6-1 (C)はある端末 P が端末 Q に着信転送登録をしておき、かつ、フリーダイヤルを設定しており端末 R がダイヤル可能状態のとき、端末 R から端末 P にダイヤルを行うと、着信は転送登録先に転送され、転送登録先端末に呼び出しがかかる。このとき、端末 R、P 間で 2 つのサービスの課金対象が違い、どちらの仕様に従うべきか決まらない。

(2) 課金レート競合

携帯電話サービスの課金レートは「r1」とする。また着信転送サービスの課金レートは「r2」とする。携帯電話を用いて発呼したとき、ダイヤル先が着信転送を登録していた場合、着信呼は転送される。このとき、課金は何れのサービスの課金レートをもとにすべきか一意に決まらない。

[課金仕様競合例終]

6. 4 課金仕様記述方式

6. 4. 1 仕様記述要素

課金仕様として記述しなければならない要素として、課金端末、課金レート、課金開始・終了条件がある。これらを規定する為には、表 6-1 に示すような様々な条件を考慮する必要がある。本章では、表 6-1 の要素の中で、課金端末と課金レートに着目した記述法を述べる。なお、課金レートは、表 6-1 に示すように様々な要因を考えて、システムごとに課金レートを決定する必要があるが、本章では、単位料金を想定して記述する。

表 6 - 1 課金仕様記述要素

要素	主な内容
課金端末	発端末，着端末，発着端末分割又は特定の第 3 者
課金レート	通話時間，距離，情報量，サービスグレード， 時間帯，曜日，伝送媒体（有線，無線）， 中継網，転送速度
課金開始・終了	通話開始・終了，サービス登録・削除， 無（非課金時）

6. 4. 2 課金規則記述

課金仕様は I F-T H E N 形の規則形式で記述する。I F 部（条件部）と T H E N 部（動作部）はセミコロンの (;) で区切る。課金規則記述方式の定義を以下に示す。

[定義 6. 3]

C ; T, R.

C : 課金規則が適用されるための、規則条件を記述する。規則条件は、

S T R 記述手法の状態記述要素の集合とイベントを用いて記述する。

例 : C = S i (A, B) E i (A, B)

S i (A, B) は状態記述要素を表し、

E i (A, B) はイベントを表す。

T : 課金対象端末と課金区間の指定。S T R 手法の端末識別子を用いて記述する。

以下の例は、A-B間の課金はA端末に、B-C間の課金はB端末に行うことを示す。

例 : A (A, B) B (B, C)

R : 課金レートとして単位料金 (r i) を記述する。

例 : R = r i

□

[事例を用いた課金仕様記述の説明]

基本電話サービス、着信転送サービス、フリーダイヤルサービスの課金規則記述例を図 6-2 に示す。

図 6-2 の基本電話サービスの課金規則は、端末 A から端末 B にダイヤルして (dial(A, B)) , 端末が応答したとき、端末 A, B 間では課金端末は A, 課金レートは「r1」に設定されることを示す。着信転送サービスの課金規則は、端末 B が C に着信

基本電話)

dial(A,B); A(A,B), r1.

着信転送)

m-cfv(B,C) dial(A,B); A(A,B) B(B,C), r1.

フリーダイヤル)

m-frd(B) dial(A,B); B(A,B), r1.

図 6 - 2 課金規則記述例

転送を登録しているとき、端末Aからダイヤルがあり、転送を行い、端末Cが応答をした場合は、端末A、B間では課金端末はA、端末B、C間では課金端末はB、課金レートは「r1」に設定されることを示す。フリーダイヤルサービスの課金規則は、フリーダイヤル設定端末B (m-frd(B)) に端末Aからダイヤルがあり、呼びだしになり、応答すると、端末A、B間では課金端末はB、課金レートは「r1」に設定されることを示す。 [事例終]

6. 5 課金規則適用

課金規則適用については、以下の2つの課金規則適用規約を用いる。

[定義 6. 4]

課金規則適用規約 1: 状態推移時に課金規則の条件部 (C) が満たされるとき、課金規則は適用可能となり、課金対象端末 (T) と課金レート (R) を設定する。

課金規則適用規約 2: 複数の規則が適用可能なとき、規則の条件部を比較し、イベントが同じで、かつ、状態に完全包含の関係がある場合、包含している規則を優先して適用する。 □

[事例を用いた課金規則適用の説明]

課金規則適用規約 1: 図 6-2 のフリーダイヤルの課金規則を例にして図 6-3 の課金規則適用図を用いて説明する。図 6-3 (A) の状態から (B) の状態への状態推移は、フリーダイヤルの S T R 規則が適用されて起きる状態推移である。このとき、課金規則の端末識別子 A = 端末 P, 端末識別子 B = 端末 Q とすると、課金規則の条件部は満足されて、課金規則は適用可能となり、課金対象端末 (端末 P-Q 間は端末 Q) と課金レート (r1) を設定する。

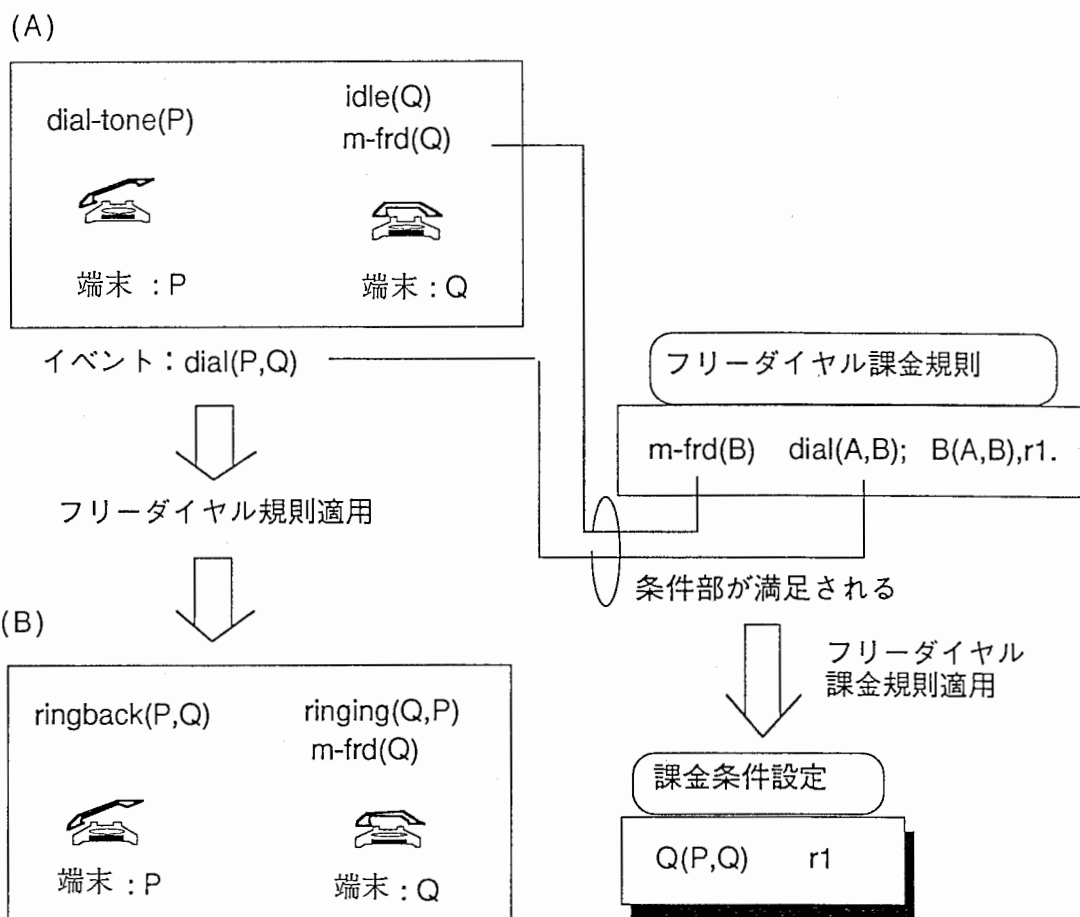


図 6 - 3 課金規則適用例

課金規則適用規約 2: 図 6-2 の基本電話規則とフリーダイヤル規則の条件部を比較すると、イベントは等しく、かつ、状態は後者が「m-frd(B)」で前者は指定がない。つまり、後者の状態が前者の状態を完全に包含している。図 6-3 の状態推移が起きるとき両方の課金規則が同時に適用可能であるが、この場合、後者を優先的に適用する。

[事例終]

6. 6 課金仕様競合検出方式

6. 6. 1 課金仕様と接続仕様の関係

サービスの接続動作に関する仕様はSTR手法 [3-4] を用いて規定するものとし、接続仕様と課金仕様はそれぞれ別々に規定し、両者の設計後の仕様を用いて課金仕様の同時適用性を判定する。課金仕様の同時適用性とは、接続仕様が規定している動作中において、課金規則の条件部（状態、イベント）が同時に満足される動作が存在することをいう。

STR手法を用いた接続仕様から、表 6-2 に示すような状態推移表が生成されていることを前提とし、状態推移表を参照し課金仕様競合の同時適用性を判定する。状態推移表は、任意の端末のとりうる状態推移を表現したものである。状態推移表を STT (State Transition Table) とすると、状態推移表は以下のように定義される。

[定義 6. 5] (状態推移表)

$$STT = \{T_0, \dots, T_n\}$$

$$T_i = \{PS, OT, E, NS\}$$

$$PS, OT, NS = \{SE_0, \dots, SE_m\}$$

SE₀ = 初期状態

T : 状態推移を表す。

PS：任意の端末をPとして，端末Pの状態を表す．

OT：端末P以外の端末（順次Q，R，Sを用いる）の状態を表す．

端末Pの状態をPS_i，規則R_jの始状態をIS_jとしたとき，

「PS_i∩IS_j」の関係にあるときは，他端末状態は

「IS_j－(PS_i∩IS_j)」とする．

E：イベントを表す．

NS：規則適用後の状態を表す．

SE：状態記述要素を表す．

状態推移表（STT）は状態推移の集合から構成され，状態推移は端末Pの状態，端末P以外の端末状態，イベント，適用規則，次状態から構成される．端末Pの状態，端末P以外の端末状態，次状態は状態記述要素の集合から構成される． □

状態推移表は端末Pに初期状態「idle(P)」を設定し，初期状態から適用可能な規則を順次適用して生成したものである．状態推移表の最左欄の端末Pの状態は規則を適用し実際に起こりうる状態であるので，「実状態」とよびその集合を「実状態集合」と呼ぶ．

表 6-2 状態推移表

状態 (実状態集合)	他端末状態	イベント	次状態
dial-tone(P)	idle(Q)	dial(P, Q)	ringback(P, Q) ringing(Q, P)
idle(P) m-cfv(P, Q) m-frd(P)	dial-tone(R) idle(Q)	dial(R, P)	pingring(P, R) m-cfv(P, Q) m-frd(P) ringback(R, Q) ringing(Q, R)

6. 6. 2 課金仕様競合検出方式の概要

課金規則の2つずつを比較して、2つの課金規則の組合わせが、1) 同時適用可能、かつ、2) 同時適用した場合に課金端末、課金レートに競合が生じる場合を検出する。課金競合検出方法の概要を図6-4に示す課金規則例を用いて説明する。

2 課金規則同時適用可能検査：

図6-4の2つの課金規則が同時に適用可能なためには、以下の4つの条件を満足する必要があり、条件を満足したとき、2つの課金規則は同時適用の可能性があると判定する。

条件1：イベントが等しいこと ($E_i(A, B) = E_j(A, B)$)。

条件2：2つの課金規則の状態（端末A = $S_i(A, B)$ ，端末B = $S_j(B, C)$ ）を満足する実際のシステム状態が存在していること。これは、STR規則から作成した状態推移表の実状態集合に個々の端末状態を満足する状態が存在しているかどうかで判定する。

条件3：状態とイベントの組み合わせが、STR規則から生成された状態推移表に実在していること。図6-4の例では、端末Aの状態「 $S_i(A, B)$ 」を満足する実状態集合からイベント「 $E_i(A, B)$ 」が生起していること。端末Bの状態「 $S_j(B, C)$ 」を満足する実状態集合はイベント「 $E_i(A, B)$ 」を受ける動作が実在していること。

条件4：2つの課金規則には規則適用の優先関係がないこと（定義6.4の課金規則適用規約2参照）。

上記の4つの条件が満足されるとき、図6-4の2つの課金規則は図6-5に示す状況で、同時適用可能と判定される。

rule-1) $S_i(A,B) \quad E_i(A,B); \quad B(A,B), r1.$
rule-2) $S_j(B,C) \quad E_j(A,B); \quad A(A,B) \quad B(B,C), r2.$

図 6 - 4 課金規則例

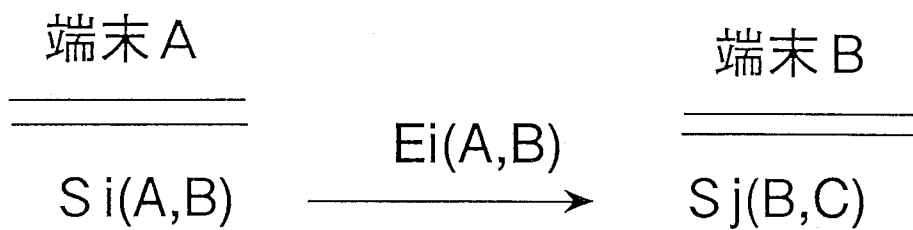


図 6 - 5 課金規則同時適用可能状況

課金仕様競合検査：

2つの課金規則が同時適用可能なとき、課金端末と課金レートに競合があるかどうかを検査する。

① 課金端末検査：同一の課金区間に対して、異なる課金端末となっているかどうか調べる。異なっている場合は、課金端末競合と判定する。図 6-4 の2つの課金規則では、図 6-6 に示すとおり、課金区間 A-B に対してそれぞれ異なる課金端末となっているので、課金端末競合と判定する。

② 課金レート競合：2つの課金規則の課金レートが異なっている場合、課金レート競合と判定する。図 6-4 の2つの課金規則では、図 6-7 に示すとおり、異なる課金レートとなっているので課金レート競合と判定する。

次節では、詳細な課金仕様競合検出アルゴリズムについて述べる。

	rule-1)	rule-2)
A－B 間	B	A
B－C 間		B

図 6-6 課金端末競合検出例

	rule-1)	rule-2)
課金レート	r1	r2

図 6-7 課金レート競合検出例

6. 6. 3 課金仕様競合検出アルゴリズム

課金仕様競合検出方式の処理の流れ図を図 6-8 に示す. この図に添って, 各手順の説明を行う.

手順 1 : 前処理 (状態の付加)

課金規則が適用可能な状態を調べるために, イベントを生起させるときのデフォルト状態を課金規則の条件部に付加する. ダイヤルができる状態はダイヤル可能状態「dial-tone(A)」なので, この状態を課金規則の状態に付加する.

手順 2 : 課金規則抽出

2 つの課金規則の組み合わせを抽出し, その組み合わせに対して, 手順 3 以降の手順を行う.

手順 3 : 2 規則同時適用検査

2 つの課金規則が同時に適用可能かどうかの検査を行う.

- ① イベント等価検査: 比較する 2 つの課金規則の条件部のイベントが等しいことを検査する. 違う場合は, 同時適用にはならないので, 手順 2 に戻る.
- ② 規則優先検査: 定義 6.4 の課金規則適用規約 2 に該当するかどうか検査する. 該当する場合は同時適用しないと判定し, 手順 2 に戻る.
- ③ 同時適用可能な状態候補作成: 2 つの課金規則の同時適用可能な状態の組み合わせを作成する. イベントの端末識別子の等価関係を, 状態の端末識別子の制約に付加し, 各端末の状態の和集合を構成する.
- ④ 端末状態の存在検査: 同時適用可能状態候補の各端末が表 6-2 の状態推移表の実状態集合に存在 (含まれているか, 等しい) しているかどうか検査する. 全ての端末状態が実状態集合に存在しない場合は, 2 つの課金規則の状態を満足する状況は存在しないので, 同時適用しないと判定し, 手順 2 へ戻る.
- ⑤ 端末状態とイベントの組み合わせ検査: 課金規則の条件部は状態とイベントの組み

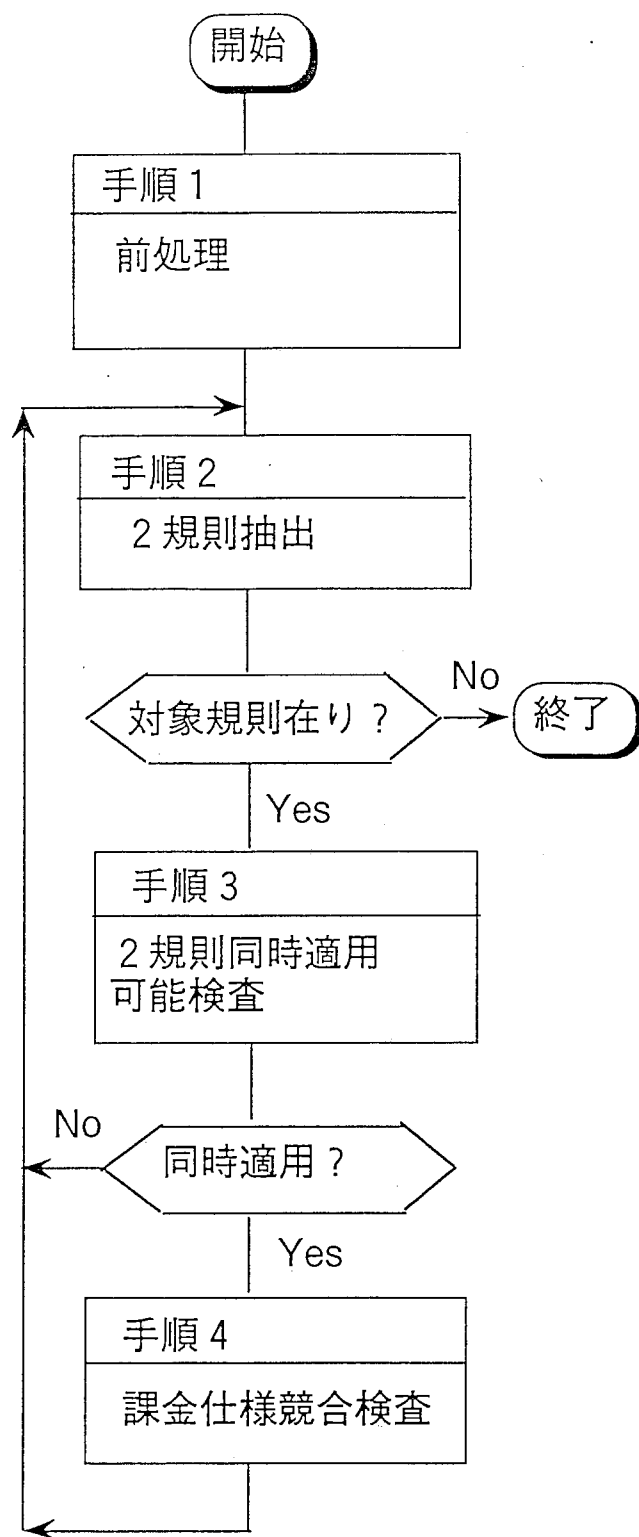


図 6 - 8 課金仕様競合検出方式の処理の流れ図

合わせで構成されている。課金規則が適用可能なためには、状態とイベントの組み合わせが同時に表 6-2 の状態推移表上に実在していることが必要である。状態とイベントの組み合わせが存在しない場合は、同時適用しないと判定し、手順 2 へ戻る。

手順 4：課金仕様競合検査

同時に適用可能な場合、課金端末、課金レートに競合が生じるかどうかの検査を行う。（詳細は、6.6.2 節の課金仕様競合検査参照）

〔事例による課金仕様競合の説明〕

手順 1：図 6-4 の各課金規則にダイヤル可能状態が付加されて図 8-9 のようになる。

手順 2：図 6-9 の課金規則から着信転送とフリーダイヤルの課金規則を抽出したとする。

手順 3：① 2つの課金規則のイベントは等しく、② 規則適用規約 2 にも該当しない。③課金規則の状態から端末状態の和集合をとると、端末 A、端末 B の各状態は、図 6-10 のように構成される。④状態候補の端末 A の状態「dial-tone(A)」は、端末識別子 $A = P$ とみなすことで、表 6-2 の 1 番目の実状態と等しいことが分かる。端末 B の状態「m-cfv(B, C), m-frd(B)」は、端末識別子 $B = P$ 、 $C = Q$ とみなすことで表 6-2 の 2 番目の実状態に含まれていることが分かる。⑤着信転送とフリーダイヤルの 2つの課金規則では、以下の組み合わせが実在しているかどうかを調べる。

1) 端末 A の状態「dial-tone(A)」と

イベント「dial(A,B)」の組み合わせ

2) 端末 B の状態「m-cfv(B,C), m-frd(B)」と

イベント「dial(A,B)」の組み合わせ

両者の状態を満足する実状態は表 6-2 に存在し、それぞれのイベントも存在して

基本電話)

dial-tone(A) dial(A,B); A(A,B), r1.

着信転送)

dial-tone(A), m-cfv(B,C) dial(A,B); A(A,B) B(B,C), r1.

フリーダイヤル)

dial-tone(A), m-frd(B) dial(A,B); B(A,B), r1.

図 6-9 手順 1 後の課金規則

端末A={dial-tone(A)}

端末B={m-cfv(B,C), m-frd(B)}

図 6-1 0 同時適用可能状態候補

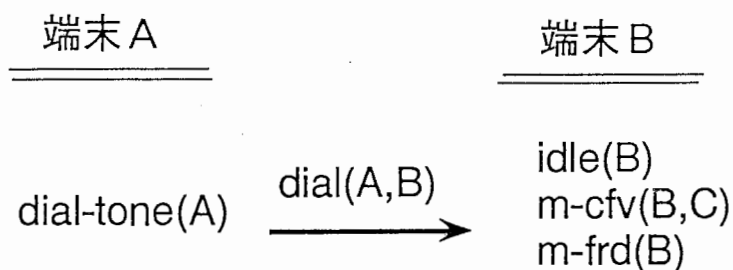


図 6-1 1 2 課金規則同時適用可能状況

携帯電話)

m-portable(A) dial(A,B); A(A,B), r2.

図 6-1 2 携帯電話サービスの課金規則

いることが分かる。

これらの操作で、着信転送とフリーダイヤルの2つの課金規則は図6-11に示す状況において同時に適用可能と判定される。

手順4：それぞれの課金区間に対する課金端末を調べると、以下のようになり、課金区間A-Bで課金端末競合が生じていると判定される。

課金区間A-Bの課金端末：

着信転送： 端末A

フリーダイヤル： 端末B

課金レートは競合していないと判定される。

課金レート競合が発生する例を以下に示す。例えば、携帯電話サービスの課金規則が図6-12のように設定されているとき、図6-4の着信転送サービスの課金規則とは同時適用と判定される。手順4の課金端末競合検査では、課金端末には矛盾はないと判定される。ところが、課金レートが違うため、課金レート競合と判定される。□

6. 7 考察

実際の 6 サービス（基本電話 (Plain Old Telephone Service), 話中着信 (Call-Waiting), 着信転送 (Call-Forward Variable), 3 者通話 (Three-Way Call), フリーダイヤル (Free-Dial), 携帯電話 (Portable-Telephone)）に対して、提案した手法を適用し有効性の評価を行う。

(1) 課金規則記述量の削減化

話中着信サービス, 3 者通話サービスの課金仕様は発呼端末と着信端末の 2 端末間を考慮すればよく, これらの 2 つのサービスは発呼端末が課金対象端末である。課金対象端末の規定は基本電話サービスの課金規定と同様であり, 課金レートも基本電話サービスの課金規定と同様であるので, 上記 2 つのサービスについては既存の基本電話サービスの課金規則を適用すればよく, 新たに課金規則を追加する必要はない。つまり, 上記 6 サービスに対する課金規則は基本電話, 着信転送, フリーダイヤル, 携帯電話の 4 個でよい。このように, 課金規則記述方式では, 課金規則の再利用により, 全てのサービスに対して課金規則を記述する必要がない分, 課金規則記述の削減化が可能となる。

(2) 課金仕様競合検出効果

上記 6 サービスに対する 4 個の課金条件規則を対象とし, 2 つの課金条件規則の組み合わせについて, 課金競合が生じるかどうかの判定を行った結果を表 6-3 に示す。2 つの課金規則の組み合わせ数は 6 組 (4 個から 2 個を組み合わせる場合の数) であり, この中から, 着信転送とフリーダイヤルの組み合わせは課金端末競合が生じると判定され, 着信転送と携帯電話の組み合わせは課金レート競合が生じると判定される。携帯電話とフリーダイヤルの組み合わせは課金端末競合と課金レート競合が生じると判定される。6 組の 2 課金規則の組み合わせ中, 3 組の課金規則の組み合わせ

表 6 - 3 課金仕様競合検出結果

	課金端末競合	課金レート競合
着信転送＋フリーダイヤル	検出	
着信転送＋携帯電話		検出
携帯電話＋フリーダイヤル	検出	検出

において課金仕様競合が生じていると判定された。

通信サービス数は増加する傾向にあり、人手による新規サービスと既存サービス間に生じる課金仕様競合の検出は、設計者の大きな負担となる。本手法を用いれば、機械的に課金仕様競合を検出することが可能となり設計支援として有効である。

(3) 課金仕様競合解消

課金規則適用規約2により、課金規則の状態の包含関係により課金規則の優先関係が決定される。この規則適用規約に従えば、既存の課金規則の修正が不要な場合の課金仕様競合解消は、課金規則の条件部の状態は競合する両者の課金規則の状態を包含する状態とし、イベントは変更せず、動作部（課金端末、課金レート）は変更規定（一方を優先させるのか、両方の規定を協調させるのか）に従った課金規則を新たに追加すればよい。

課金端末競合をおこすフリーダイヤルサービスと着信転送サービスに対する解消例として、端末A-B間は端末Bに課金するように規定した場合の解消例を図6-13に示す。

このように、提案する課金仕様記述方式を用いると、既存の課金規則の修正を行う手戻りがすくなく、競合解消効率がよい。

(4) 設計の効率化

従来のSDL[7]では、課金処理は状態推移手続き中のタスクの中に含まれる。この段階で課金仕様競合を検出し、修正を加えるのでは、修正作業の手戻りが大きい。この問題に対する解決策として、設計の上流工程である通信サービス設計段階で、サービス間に生じる課金仕様競合を検出する方式を提案した。つまり、提案した方式を用いると、状態推移動作仕様と課金仕様をそれぞれ設計し、機械的に統合することにより課金仕様競合検出が可能となる。次工程（SDL仕様作成）に進む前に課金仕様競合の解消が可能となり、従来に比べ、設計の手戻りによる工数が不要になる分、設計

着信転送)

m-cfv(B,C) dial(A,B); A(A,B) B(B,C), r1.

フリーダイヤル)

m-frd(B) dial(A,B); B(A,B), r1.



A-B間は端末Bに課金する

着信転送＋フリーダイヤル)

m-frd(B),m-cfv(B,C) dial(A,B); B(A,B) B(B,C), r1.

図 6 - 1 3 課金端末競合解消例

の効率化が可能となる。

(5) 検討サービスの一般性

通話料金を決定する要素は表 6-1 に示しているとおりの多様であり、課金を主眼としたサービスとしては、それらの要素を組み合わせたサービスが将来的に提供される可能性がある。一般的には、料金算出に係わる要素と要素間の関係を定義する必要がある。また、料金の分担を通話時に設定するようなサービスも考えられ、動的に課金仕様要素を変更可能にできる枠組みの検討も必要である。検討した 6 つのサービスは、将来にわたって多様化する全ての課金サービスに対して、一般性を持つものではない。

課金端末については、発側課金か、着側課金か、発側と着側の分担課金か、第 3 者への課金（無料も含む）か、に分類される。一般的には、課金は通話に係わる端末に対して課金を行うことが原則であり、発側課金、着側課金、発側・着側分担課金に係わるサービスを検討対象とした。この観点から 6 つの検討サービスを分類すると表 6-4 のようになり、課金端末に関する現在のサービス検討対象としての要件は満足されている。また、料金は基本的には単位料金（時間帯毎）と距離と通話時間により決定される。距離と通話時間はシステムの測定により自動的に求められるため、単位料金が決まれば料金は算出できる。検討した 6 つのサービスは、課金レート（単位料金）以外の要素は共通であると想定しても問題はなく、基本的な料金算定の検討対象としての要件は満足されている。

表 6-4 検討サービスの分類

課金端末	サービス
発信課金	基本電話，話中着信， 3 者通話，携帯電話
着信課金	フリーダイヤル
発信・着信 分割課金	着信転送

6. 8 第6章のまとめ

通信サービス設計の上流工程におけるサービスインタラクション設計に必要な課金仕様競合検出方式の提案を行った [8] . まず, 課金仕様に必要な条件として, 課金対象端末, 課金レートに着目し, それらを規則形式で記述する方式を提案した. 次に, この課金規則が同時適用可能なときに生じる課金仕様競合検出方式 (課金端末競合検出方式, 課金レート競合検出方式) を提案した. また, 実際の6個のサービスを対象に課金仕様を記述し, その課金仕様間で生じる課金仕様競合検出の机上実験結果を示すとともに, 課金仕様競合解消の方法を示し, 提案した課金仕様記述方式と課金競合検出方式の有効性を示した.

今後の課題として以下の項目を検討している.

- (1) システム試作: 提案方式のシステム化を行い, 有効性を実証する.
- (2) 課金サービスの記述範囲の拡張: 課金サービスの種類は, 多様化する傾向にあり, それらの仕様記述を可能にするよう記述方式の拡張を検討する.

第 6 章の参考文献

- [1] Duran, J. M. and Visser, J. : International Standards for Intelligent Networks,
IEEE Communication Magazine, Vol. 30, No. 2, pp. 34-42 (1992).
- [2] Collet, P. and Kung, R. : The Intelligent Network in France,
IEEE Communication Magazine, Vol. 30, No. 2, pp. 82-89 (1992).
- [3] 原田良雄, 平川豊, 竹中豊文, 門田充弘 : サービス仕様の自動生成に関する
考察, 情処第 39 回大会, 5S-5 (1989).
- [4] Hirakawa, Y., Harada, Y. and Takenaka, T. : 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 (1990).
- [5] Harada, Y., Hirakawa, Y., Takenaka, T., Terashima, N.: A Conflict Detection
Support Method for Telecommunication Service Descriptions,
信学論文誌, Vol. E-75-B, No. 10, pp. 986-997 (1992).
- [6] 菊田弘之, 八木輝, 酒井清一郎, 藤岡雅宣, 若原恭 : サービス・インターラク
ション検出方法の提案, 1992信学春季大会, B-527 (1992).
- [7] CCITT Recommendations Z. 100 to Z. 104 (1985).
- [8] 原田良雄, 高見一正, 太田理, 寺島信義 : 通信サービス課金仕様競合検出方式,
情処論文誌, Vol. 34, No. 5, pp. 1064-1073 (1993).

第7章 結 論

通信サービスの接続動作を利用者の視点から形式的な規則形式で記述する S T R 手法を提案し、その手法を用いて通信サービスは記述されることを前提とした、新規サービス設計支援方法およびサービス合成時における相互作用設計支援方式を提案した。また、サービスには、接続動作だけではなく課金を主眼とするサービスもあり、課金を主眼とするサービスに対しては課金仕様記述方式を提案し、課金仕様の競合検出方式を提案した。提案内容とその特徴を以下に示す。

(1) 通信サービス記述手法 (S T R 手法) [1-2]

利用者の視点から通信サービス動作を、形式的な規則形式の集合で記述する S T R 手法を提案した。S T R 手法の規則記述と規則適用方法を示した。S T R 手法の特徴は、1) 規則の部分適用 (イベントが生起したとき、実状態に規則の始状態が含まれていれば規則は適用可能)、2) 状態の書き換え (規則の始状態を次状態に書き換える)、3) 規則の優先適用 (始状態の包含している方の規則を優先する)、である。S T R 手法を前提とした場合の新規サービス設計時において、既存規則の再利用による記述量の削減化とサービスの特徴的な記述のみ (差分記述) で通信サービスの記述が可能になり効率的な通信サービス設計ができることを示した。基本電話サービス規定を再利用することを前提とした話中着信サービスの記述実験では、本来記述すべき話中着信サービス規定の 40 % 程度の記述量であることを示した。P B X サービスを対象にした記述実験から、記述の網羅性は 94 % であったことを示した。

(2) 通信サービス (相互作用) 設計支援 [3-5]

新規サービス設計支援とサービス合成時における相互作用設計支援の方法を示し

た.

新規サービス設計支援としては、1) 視覚的にサービス動作を表示できる解釈実行システムを用いて、サービス規定が設計者の意図どおりであるかどうかの確認支援を行う方法を示した。また、2) 新規サービスの規則集合を入力として、任意の端末に着目して、状態推移を生成し、設計者に提示し、網羅的な動作の確認支援を行う方法を示した。

通信サービス相互作用設計支援としては、1) 検証項目の分類として状態推移モデル上の問題かどうかを基準として示し、前者を論理誤り、後者を意味誤りとし、2) 各項目に基本的な検出方法を示した。また、修正における、影響波及範囲の検出の基本的な方法も示した。

新規サービスおよび相互作用検証に必要な状態推移を、任意の端末に着目して生成する方式を示した。

サービスを追加する際に既存サービスとの間で生じる矛盾の解消は重要な課題であるが、相互作用数は既存サービスが増加するに従い組み合わせ的に増加し、その検討範囲の限定化は人手で行うことのできる範囲を超えたものとなっている。提案手法により、相互作用の検討範囲、修正影響範囲の絞り込みが可能になり、サービス設計の効率的な支援が可能となる。

(3) 通信サービス競合検出方式 [6-8]

通信サービスの非決定的動作の検出の問題は、STR手法を用いることによって規則競合の問題として扱うことができる。本論文では、イベントの同じ2規則の組み合わせを比較して、2規則が同時に適用される可能性のある状況を検出し、設計者に提示する方法を提案した。実験結果から、提案した競合検出手法は、機械的に検討対象を絞り込む効果が大きく（POTS+CW+CFV+3WC+CCBSでは1%以下）設計支援に

有効であることを示した。また、検出結果は、競合する可能性のある2規則の組み合わせを網羅していることを示した。

(4) 通信サービス相互作用に含まれる意味的矛盾動作検出方式 [9]

通信サービス設計段階における相互作用設計支援の一環として、相互作用に含まれる意味的矛盾動作検出方式を提案した。提案手法の特徴は、意味的に矛盾する状態（矛盾状態知識）を設定し、サービス合成時に矛盾状態知識を含む状態を検出し、初期状態から矛盾状態に至る状態推移を提示し、設計者の原因解析を促す手法である。意味的に矛盾する状態（矛盾状態知識）の設定に際しては、3種類の矛盾状態知識の獲得支援の方法を示した。また、STR手法の規則優先適用を前提とした矛盾の解消方法（規則の修正方法）も示した。提案手法のシステム試作を行い、実際の5つのサービスを対象に実験を行った。その結果より、矛盾状態の検討範囲として12%以下の絞り込み効果があり、検証作業の効率化が図れることを示した。また、初期状態から矛盾状態に至る状態推移を示すことで、意味的矛盾動作の解析が効率的に行うことができることを示した。

(5) 通信サービス課金仕様競合検出方式 [10]

通信サービス設計の上流工程におけるサービスインタラクション設計に必要な課金仕様競合検出方式の提案を行った。まず、課金仕様に必要な条件として、課金対象端末、課金レートに着目し、それらを規則形式で記述する方式を提案した。次に、この課金規則が同時適用可能なときに生じる課金仕様競合検出方式（課金端末競合検出方式、課金レート競合検出方式）を提案した。実際の6個のサービスを対象に課金仕様を記述し、その課金仕様間で生じる課金仕様競合検出の机上実験結果を示すととも

に，課金仕様競合解消の方法を示し，提案した課金仕様記述方式と課金仕様競合検出方式の有効性を示した．

今後の課題として，以下の項目を検討している．

(1) STR手法の機能拡張：

繰り返し操作に関するサービスや，グループ内の端末操作，等が記述できるように機能拡張する必要がある．SDL [11] は，状態，信号，タスクを記述する枠組みは準備しており，記述する内容は設計者に委ねている．コメントも欄外に自由に記述できる．記述手法としては，全てのサービスが記述可能であるが，反面，サービス相互作用に含まれる矛盾については，設計者があらかじめ発見し，かつ，解消することを前提としており，問題解決には向かない記述方式である．STR手法を用いてサービスを形式的に規則で記述すれば，本論文で示したとおり，解析支援が可能となる．通信サービスは，多様化，複雑化する傾向にあり，それらが形式的に記述できるかどうか，解析を容易にできるかどうかの鍵となる．

(2) 通信サービス相互作用設計支援手法の拡張：

現在，通信サービス相互作用として検証すべき項目として，何を行えばよいか自体が，今だに明確にはなっていない状況である．検証項目としては，設計段階，プログラム作成段階，実行段階において，今後の検討が必要である．本論文では，通信サービスの設計段階における，利用者の観点からみたサービスの振る舞いに着目した，サービス相互作用の解析手法を提案した．基本的には，提案した状態推移生成の情報をもとに様々な検出項目への対応が可能と考えている．今後，さらに，具体的な問題点に対して検出アルゴリズムを検討し，効率の良い設計・検証システムの構築を行う必要がある．

(3) 課金仕様記述範囲の拡張：

課金サービスの種類は、多様化する傾向にあり、それらの仕様記述を可能にするよう記述方式の拡張を検討する。

第7章の参考文献

- [1] 原田良雄, 平川豊, 竹中豊文, 門田充弘: サービス仕様の自動生成に関する考察, 情処第39回大会, 5S-5 (1989).
- [2] Hirakawa, Y., Harada, Y. and Takenaka T.: 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 (1990).
- [3] Harada, Y., Hirakawa, Y. and Takenaka T.: A Design Support Method for Telecommunication Service Interactions, GLOBECOM'91, pp. 1661-1666 (1991).
- [4] 原田良雄, 平川豊, 竹中豊文: 通信サービス追加設計支援方法の一考察, 信学技法, Vol. 90, No. 454, pp. 13-18 (1991).
- [5] Takami, K., Harada, Y., Ohta, T., Terashima, N.: A Visual Design System for Telecommunication Service Descriptions, IPCCC, pp. 593-599 (1993).
- [6] Harada, Y., Hirakawa, Y., Takenaka, T., Terashima, N.: A Conflict Detection Support Method for Telecommunication Service Descriptions, 信学論文誌, Vol. E75-B, No. 10, pp. 986-997 (1992).
- [7] 原田良雄, 平川豊, 竹中豊文: 通信サービス動作競合検出方法, 情処研報, Vol. 91, No. 83, pp. 87-92 (1991).
- [8] 原田良雄, 平川豊, 竹中豊文: 通信サービス競合検出システムの試作と評価, 信学技法, Vol. 92, No. 75, pp. 13-18 (1992).
- [9] 原田良雄, 高見一正, 太田理, 寺島信義: 通信サービス相互作用に含まれる意味的矛盾動作検出方式, 情処論文誌, Vol. 35, No. 8, pp. 1602-1613 (1994).
- [10] 原田良雄, 高見一正, 太田理, 寺島信義: 通信サービス課金仕様競合検出方式, 情処論文誌, Vol. 34, No. 5, pp. 1064-1073 (1993).
- [11] CCITT Recommendations Z. 100 to Z. 104 (1985).

謝 辞

本論文をまとめるにあたり、丁寧な御指導を賜りました京都大学工学部 松本吉弘教授に厚く感謝の意をささげる次第です。

本研究の機会を与えられ、種々の御指導と御鞭撻を賜った、株式会社ATR通信システム研究所 代表取締役社長 寺島信義博士、通信ソフトウェア研究室元室長 竹中豊文博士（現NTT広帯域ネットワーク研究所）、現室長 太田 理博士、平川 豊博士（現NTTソフトウェア研究所）ならびに高見一正氏（現NTTネットワークシステム研究所）に衷心より感謝の意をささげます。

また、ATR通信システム研究所への出向の機会を与えて頂きました、株式会社東洋情報システム 田中良信取締役に感謝の意を表します。さらに、出向中も種々の御指導と御鞭撻を賜りました 同社専務取締役 兼 株式会社東洋情報システム技術研究所 所長 伊藤 周博士に心からの感謝の意を表します。

研究業績

種別	題名 及び 著者	誌名, 巻号 (年), NO, 等
論文誌	A Conflict Detection Support Method for Telecommunication Service Descriptions <u>Harada, Y.</u> Hirakawa, Y. Takenaka, T. Terashima, N.	電子情報通信学会 論文誌 通信ソフトウェア小特集号 vol. E75-B, No. 10, pp. 986-997 (平成4年10月号)
	通信サービス課金仕様競合検出方式 <u>原田良雄</u> 高見一正 太田理 寺島信義	情報処理学会 論文誌 Vol. 34, No. 5, pp. 1064-1073 (平成5年5月号)
	通信サービス相互作用に含まれる 意味的矛盾動作検出方式 <u>原田良雄</u> 高見一正 太田理 寺島信義	情報処理学会 論文誌 Vol. 35, No. 8, pp. 1602-1613 (平成6年8月号)
	A Method of Automatic State Acquisition for Conflict Detection of Telecommunication Service Descriptions <u>Harada, Y.</u> Ohta, T. Terashima, N.	International Journal on EAAI (Engineering Application of AI) (採録決定)
	A Detection Support Method for Dynamic Telecommunication Service Interactions <u>Harada, Y.</u> Takami, K. Ohta, T. Terashima, N.	IEEE Tran. on Software Engineering (投稿中)
国際 会議	A Design Support Method for Telecommunication Service Interactions <u>Harada, Y.</u> Hirakawa, Y. Takenaka, T.	IEEE Global Communication Conference (GLOBECOM) pp. 1661-1669 (Dec. 1991)
	Behavior Description for a System Which Consists of an Infinite Number of Processes Hirakawa, Y. <u>Harada, Y.</u> Takenaka, T.	BILKENT Int. Conf. on New Trends in Communications, Control, and Digital Processing pp. 59-68 (July 1990)
	A Visual Design Support System for Telecommunication Services Takami, K. <u>Harada, Y.</u> Ohta, T. Terashima, N.	International Phoenix Conference on Computers and Communications (IPCCC) pp. 593-599 (March 1993)

種別	題名 及び 著者	誌名, 巻号 (年), NO, 等
研究会 発表	通信サービス追加設計支援方法の一考察 <u>原田良雄</u> 平川 豊 竹中 豊文	電子情報通信学会 信学技法 (交換システム研究会) Vol. 90, No. 454, pp. 13-18 (平成3年3月)
	通信サービス動作競合検出方法 <u>原田良雄</u> 平川 豊 竹中 豊文	情報処理学会 情処研報 (マルチメディアと分散処理 研究会) Vol. 91, No. 83, pp. 87-92 (平成3年9月)
	通信サービス競合検出システムの試作と評価 <u>原田良雄</u> 平川 豊 竹中 豊文	電子情報通信学会 信学技法 (交換システム研究会) Vol. 92, No. 75, pp. 13-18 (平成4年5月)
	通信サービス課金処理競合検出方式の検討 <u>原田良雄</u> 高見 一正 太田 理	電子情報通信学会 信学技法 (人工知能と知識処理研究会) Vol. 92, No. 237, pp. 41-48 (平成4年9月)
	状態遷移表生成の精度向上手法 ——機械的知識獲得による複数端末状態の 同期性判定法—— <u>原田良雄</u> 高見 一正 太田 理	電子情報通信学会 信学技法 (交換システム研究会) Vol. 92, No. 509, pp. 55-60 (平成5年3月)
	通信サービス動作の修正影響解析支援方法 <u>原田良雄</u> 高見 一正 太田 理	電子情報通信学会 信学技法 (交換システム研究会) Vol. 93, No. 257, pp. 93-98 (平成5年9月)
	通信サービス相互作用の検出支援の一方式 <u>原田良雄</u> 太田 理	電子情報通信学会 ネットワーキングアーキテクチャ ワークショップ予稿集 pp. B-4: (1)-(8) (平成6年8月)

種別	題名 及び 著者	誌名, 巻号 (年), NO, 等
全国 大会 発表	サービス仕様の自動生成に関する考察 <u>原田良雄</u> 平川 豊 竹中 豊文 門田 充弘	情報処理学会 第39回大会 5S-5 (平成元年10月)
	サービス仕様の自動生成に関する考察 ——自動生成機構に関する考察—— <u>原田良雄</u> 平川 豊 竹中 豊文	情報処理学会 第40回大会 3R-2 (平成2年3月)
	サービス記述検証支援システムの構想 <u>原田良雄</u> 平川 豊 竹中 豊文	情報処理学会 第41回大会 4H-9 (平成2年9月)
	通信サービス追加時のサービス絡み動作 設計支援方法 <u>原田良雄</u> 平川 豊 竹中 豊文	電子情報通信学会 1991信学春季大会 B-543 (平成3年3月)
	通信サービス動作競合検出手法 <u>原田良雄</u> 平川 豊 竹中 豊文	電子情報通信学会 1991信学秋季大会 B-339 (平成3年9月)
	通信サービス相互作用に含まれる意味的矛盾動 作検出支援方法 <u>原田良雄</u> 高見 一正 太田 理	電子情報通信学会 1992信学秋季大会 B-385 (平成4年9月)
	機械的な知識獲得によるサービス競合検出方法 の精度向上 <u>原田良雄</u> 高見 一正 太田 理	電子情報通信学会 1993信学春季大会 B-577 (平成5年3月)
	通信サービス仕様合成における検証項目の分類 の一検討 <u>原田良雄</u> 高見 一正 太田 理	電子情報通信学会 1994信学春季大会 B-619 (平成6年3月)

種別	名称	出願日等
特許出願	競合検出方式（特願平 4-033593）	平成4年2月20日