OUKA
The University of Osaka
Institutional Knowledge Archive

Title	通信サービス要求仕様化支援技術に関する研究
Author(s)	高見, 一正
Citation	大阪大学, 1997, 博士論文
Version Type	VoR
URL	https://doi.org/10.11501/3129338
rights	
Note	

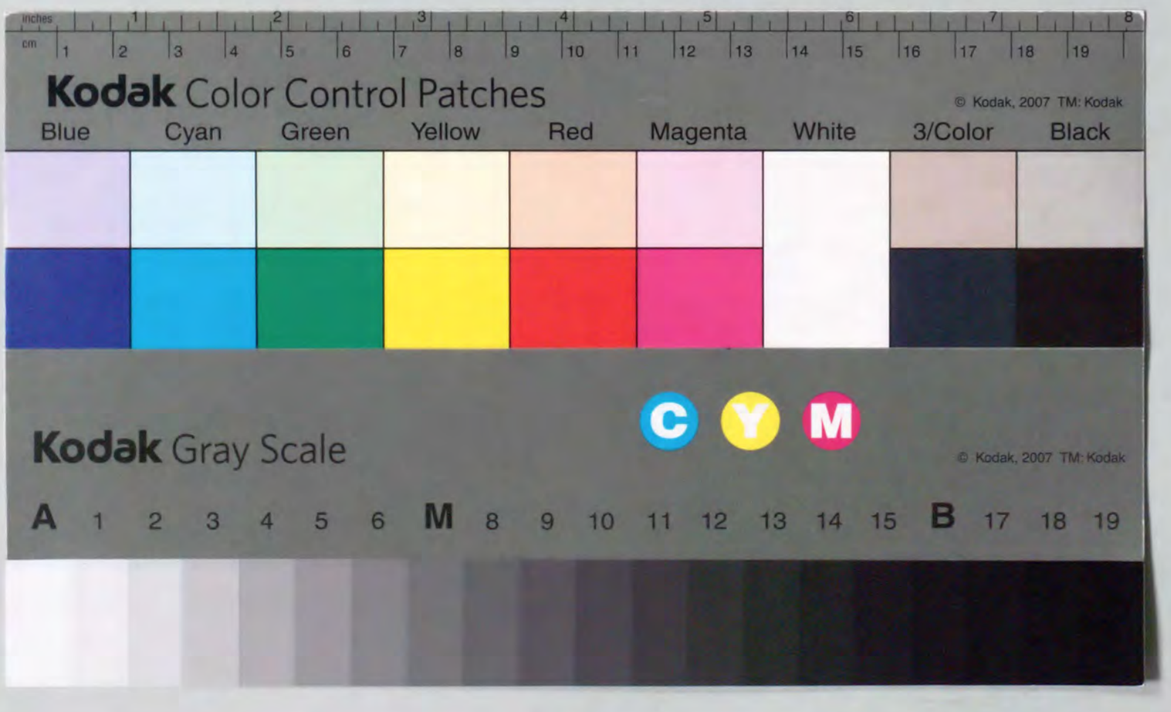
The University of Osaka Institutional Knowledge Archive : OUKA

https://ir.library.osaka-u.ac.jp/

The University of Osaka

通信サービス要求仕様化
交換技術に関する研究

青見一正



通信サービス要求仕様化支援技術に関する研究

1997年1月

高見 一正

論文内容の要旨

博士論文題名：通信サービス要求仕様化支援技術に関する研究

学位申請者：高　見　一　正

通信サービスの多様化に伴い、通信サービスを実現するソフトウェア開発の効率化が望まれている。本論文は非専門家でも通信サービスに対する要求を正確に反映した仕様を形式的記述として獲得でき、ソフトウェアを作成できるための仕様化支援技術に関する研究成果をまとめたものであり、以下の8章より構成した。

1章では本研究の背景と目的、要求条件と課題、従来研究との関係を述べた。

2章では非専門家のための仕様入力と仕様矛盾検証結果確認を支援する視覚的手法を提案した。ユーザの視点からの図形による仕様入力支援における仕様記述モデルとして端末の状態とイベントに基づく状態遷移モデルを採用し、端末状態とイベントの図形シンボルによる仕様記述法、通信サービス仕様の動的な振る舞いをアニメーションにより表現した仕様確認法、及び新規サービス仕様と既存サービス仕様間で検出された状態遷移の非決定性に起因するサービス競合を解消するための解消仕様の生成アルゴリズムと生成仕様の図形表示による支援技術を実現した。

3章ではサービス仕様競合検証支援技術として状態遷移の意味的矛盾に着目した検出法を提案した。意味的矛盾として2つのサービスを同時に提供した場合に生成される異常な状態（単独サービス仕様には規定されていない状態）への遷移の定義と機械的な検出法を示した。

4章では記述言語の異なる記述仕様との合成を支援するための仕様記述変換技術を提案した。メッセージシーケンス図（ＭＳＣ）の手続き型の仕様記述法で記述されたサービス仕様の合成と仕様検証能力を向上させることを目的として、ＭＳＣで記述された仕様のif-then 型のルール集合による宣言型記述仕様への変換法を示した。

5章ではＭＳＣによるサービス仕様記述支援を目的として、メッセージの入出力間の関係を3つのタイプに分類したif-then 型の信号規則を定義し、その信号規則の適用による提案型の設計支援法と仕様補完技術を提案した。

6章ではサービスを実現するための高度インテリジェントネットワーク（ＩＮ）に基づく網機能配備をパーソナル通信サービスの観点から述べ、詳細化すべきサービスソフトウェア仕様のネットワークでの位置づけを明確化した。

7章ではサービス要求仕様から高度ＩＮに基づくサービスソフトウェア仕様への詳細化支援技術を提案した。

8章では本論文で得られた結論を総括した。

目　次

謝辞 …… 1

第1章　序論 …… 2
1.1　背景と目的 …… 2
1.2　要求条件と課題 …… 4
1.3　従来研究と本論文の関係 …… 7
1.3.1　仕様入力との関係 …… 7
1.3.2　仕様統合化との関係 …… 7
1.3.3　仕様完全化との関係 …… 9
1.3.4　仕様詳細化との関係 …… 10
1.4　論文の構成と各章の概要 …… 11

第2章　通信サービスの視覚的な要求仕様記述支援法 …… 15
2.1　緒言 …… 15
2.2　図形による仕様記述 …… 15
2.2.1　サービス要求仕様 …… 15
2.2.2　図形による仕様記述法 …… 19
2.3　アニメーションによる仕様確認 …… 22
2.3.1　端末状態の絵シンボル …… 22
2.3.2　アニメーション表現アルゴリズム …… 24
2.4　サービス競合解消 …… 26
2.5　仕様化支援環境 …… 27
2.6　評価及び考察 …… 29
2.6.1　記述実験 …… 29
2.6.2　図形による仕様記述 …… 29
2.6.3　アニメーションによる仕様確認 …… 31

2.6.4　サービス競合解消 …… 33
2.6.5　提案支援法の有用性 …… 35
2.7　結言 …… 37

第3章　通信サービスの要求仕様検証支援法 …… 38
3.1　緒言 …… 38
3.2　異常な状態への遷移の定義 …… 38
3.2.1　サービス仕様競合検証支援と前提条件 …… 38
3.2.2　異常な状態への遷移 …… 39
3.3　仕様記述言語 …… 47
3.4　異常な状態への遷移検出法 …… 48
3.5　異常な状態への遷移の検出例 …… 50
3.6　評価及び考察 …… 53
3.6.1　検出実験 …… 53
3.6.2　検出精度 …… 53
3.6.3　人手による検出削減効果 …… 55
3.6.4　実現上の課題 …… 55
3.6.5　ルール集合の管理検証に関する考察 …… 56
3.7　結言 …… 57

第4章　通信サービスにおけるメッセージシーケンス図記述から
ルール記述への仕様変換法 …… 59
4.1　緒言 …… 59
4.2　ＭＳＣによる仕様記述 …… 59
4.3　ＭＳＣからルールへの変換法 …… 60
4.3.1　前提条件 …… 60
4.3.2　ＭＳＣ記述と端末状態とイベントの対応 …… 63
4.3.3　状態とイベントへの変換 …… 65
4.3.4　変換アルゴリズム …… 69
4.4　適用例と考察 …… 70

4.4.1　具体的事例への適用と変換法の正当性 …… 72
4.4.2　サービス仕様確認への適用 …… 72
4.4.3　サービス仕様競合検証への適用 …… 74
4.5　結言 …… 76

第5章　信号規則による通信サービス仕様設計支援法 …… 77
5.1　緒言 …… 77
5.2　通信サービス設計 …… 77
5.3　ＭＳＲ記述法 …… 79
5.3.1　メッセージの抽象化 …… 80
5.3.2　メッセージ間の関係モデルとその形式化 …… 82
5.3.3　ルール適用法 …… 85
5.4　ＭＳＲによる仕様設計支援法 …… 87
5.4.1　代表的なサービス動作の支援 …… 87
5.4.2　付随するサービス動作の支援 …… 89
5.5　具体的な設計支援例 …… 89
5.5.1　着信転送サービスの代表的動作の設計 …… 91
5.5.2　例外メッセージ系列の生成 …… 92
5.6　設計支援環境の試作 …… 94
5.7　評価及び考察 …… 97
5.7.1　ルールの再利用性 …… 97
5.7.2　設計支援法の有効性 …… 97
5.8　結言 …… 100

第6章　高度ＩＮによるパーソナル通信サービスのための網機能配備法 …… 102
6.1　緒言 …… 102
6.2　パーソナル通信サービスと要求条件 …… 102
6.2.1　コンセプトと要求条件 …… 102
6.2.2　通信網の所要機能 …… 105
6.3　パーソナル通信網アーキテクチャ …… 108

6.3.1　高度ＩＮによるアーキテクチャ……108
6.3.2　ＳＬＰ構成法……110
6.3.3　パーソナルモビィリティ制御法……112
6.3.4　サービスのパーソナル化とポータビィリティ……114
6.4　考察……114
6.4.1　サービス展開法……114
6.4.2　ＮＡＰアーキテクチャ……116
6.4.3　ローミング法……116
6.5　結言……119

第７章　知識を用いた通信サービス仕様の段階的な詳細化支援法……120
7.1　緒言……120
7.2　仕様詳細化支援法……120
7.2.1　概要……120
7.2.2　詳細化支援工程……121
7.3　仕様定義法……124
7.3.1　詳細仕様定義……124
7.3.2　ＳＬＰソフトウェア仕様……124
7.4　機能仕様からＳＬＰソフトウェア仕様への変換……126
7.4.1　コンセプトと課題……126
7.4.2　仕様詳細化知識……127
7.4.3　準正常手順補完……135
7.4.4　変換アルゴリズム……135
7.4.5　具体的事例への適用……139
7.5　ＳＬＰ生成ツール……139
7.6　評価及び考察……141
7.6.1　記述実験と変換アルゴリズムの正当性……141
7.6.2　ＳＬＰ開発工程消化時間の削減度……141
7.6.3　生成されるＳＬＰの品質……146
7.6.4　高度ＩＮの能力追加・変更に対する柔軟性……146

7.7　結言……147

第８章　結論……148

参考文献……152

付録……164

発表論文一覧……176

謝　辞

著者が日本電信電話株式会社（ＮＴＴ）交換システム研究所配属以来、終始御指導、御支援を頂き、本論文をまとめるに際しても、懇切なる御指導ならびに御助言、御尽力を賜った大阪大学工学部教授・池田博昌博士に心からの感謝の意を表します。

本論文をまとめる過程に際し、丁重なる御指導、御教示を賜った大阪大学産業科学研究所教授・元田浩博士に厚く感謝の意を表します。

また、本論文に対して有益なる御討論、御助言を頂いた大阪大学工学部教授・倉薗貞夫博士、同教授・森永規彦博士、同教授・小牧省三博士、同教授・前田肇博士、同教授・児玉裕治博士、同教授・長谷川晃博士に深謝致します。

本研究は主に著者が国際電気通信基礎技術研究所（ＡＴＲ）における通信サービスソフトウェア自動作成方式の研究とＮＴＴにおける高度インテリジェントネットワーク（ＩＮ）のサービス生成方式の研究開発の一環として担当したものである。この間御指導と御鞭撻をいただいたＮＴＴエレクトロニクステクノロジ株式会社顧問・葉原耕平博士（前ＡＴＲ通信システム研究所会長）、早稲田大学教授・寺島信義博士（前同研究所社長）、創価大学教授・太田理博士（前同・通信ソフトウェア研究室室長）、ＮＴＴマルチメディアネットワーク研究所・ネットワークインテグリティ研究部・竹中豊文グループリーダ（元同室長）、さらにＮＴＴネットワークサービスシステム研究所（ＮＳ研）・鈴木滋彦所長、ＮＴＴ通信網総合研究所・研究企画部・弓場英明部長（前ＮＳ研・高機能処理研究部（Ｎ処部）部長）、ＮＳ研・Ｎ処部・花澤隆部長、同サービス制御方式研究グループ（ネ式Ｇ）・関口英生グループリーダ及び新津善弘主幹研究員に深く感謝致します。

状態遷移規則による仕様記述法の研究を共にし、多くの討論と助力をいただいたＮＴＴマルチメディア総合研究所・研究企画部・平川豊担当部長（元ＡＴＲ通信システム研究所・通信ソフトウェア研究室主任研究員）に感謝致します。また、第２章～第５章までは通信ソフトウェアの自動生成の研究を共にした元ＡＴＲ通信システム研究所・通信ソフトウェア研究室の元治景朝氏、原田良雄氏、井上泰彰氏をはじめとする皆様に、第６章～第７章については、高度ＩＮ方式とサービス生成法の研究開発を共に進めたＮＳ研・Ｎ処部・ネ式Ｇの杉山武志研究主任、坂井博氏、前同研究グループ・本多麻理子研究主任をはじめとする多くの皆様に助力をいただいており、ここに感謝致します。

第１章　序　論

1.1　背景と目的

近年の情報社会の進展は通信サービスに高度化とパーソナル化及びマルチメディア化を要求している。このような通信サービスの多様化に伴い、通信サービスを実現するソフトウェア開発の効率化が望まれ、ソフトウェア開発の上流工程において要求仕様を早期に確定することの重要性が認識されている。ソフトウェア開発の上流工程において、要求仕様の誤りを早期に検出・解消することにより、下流工程での試験が効率的に実施でき、手戻りの少ない開発により生産性の向上が期待できる。このような目的から、ソフトウェアの要求仕様を形式的仕様記述言語で定義することにより仕様化の機械的な支援を目指す技術が期待されている。

通信サービスは図1.1に示ように主に各種の機能を持った複数のノードからなるネットワークにより提供され、サービスの柔軟性は特に各ノードのプラットフォーム上で動作するアプリケーションソフトウェアによって提供される。これまでの設計支援では交換機の仕様やOSIのプロトコルを規定するための仕様記述言語をベースとして、高い技術力を持ったソフトウェアシステム開発者が要求を分析し、ソフトウェア仕様を形式的に記述することを前提とした図1.2に位置付けるような設計支援技術が中心に検討されてきた。このような支援技術としてCCITTやISOにおいて開発された形式的仕様記述言語であるＳＤＬ，Estelle,LOTOSに基づく設計支援システムが報告されている[(1)~(3)]。これらの支援システムの多くは、形式的仕様記述言語をベースとしており、言語に関する知識と通信システムに関する専門的な知識を必要とする設計者がソフトウェア仕様設計段階において主に適用することを前提に実現したものといえる。しかしながら、近年研究が盛んな交換システムにおけるIntelligent Network(ＩＮ)では、サービス機能配備の階層化によるサービス追加の柔軟性の向上を図るという観点から、物理的・論理的サービス制御アーキテクチャ、信号プロトコル（フォーマット、コーディング、手順）、モジュール化された機能とその機能配備法等の通信システムに関する専門的な知識を必要としないサービス開発者（非専門家）が通信サービスを仕様化できるような環境に重点がおかれているのが特徴である。このため、そのようなサービス開発者のための新しい仕様化支援技術が必要となる。

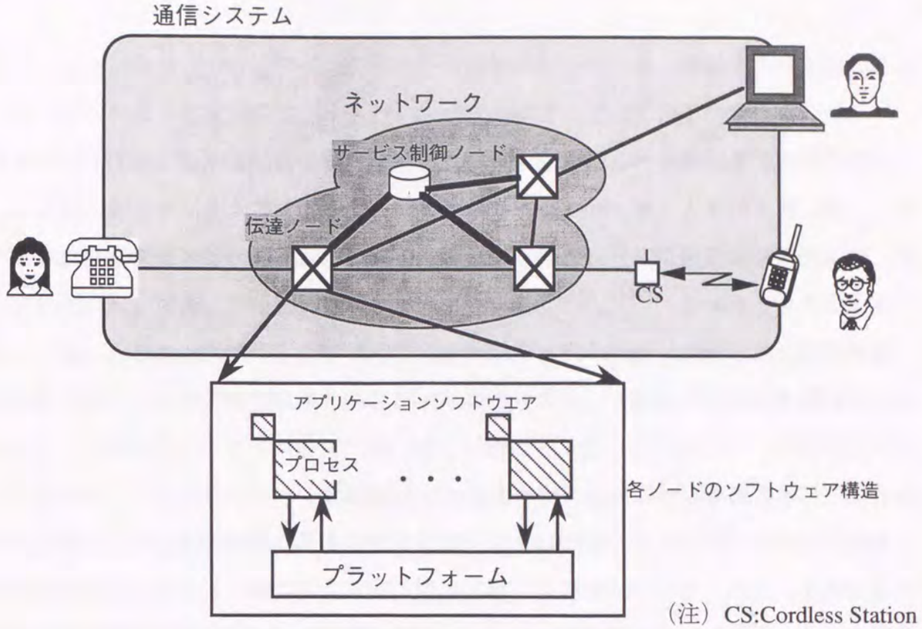


図1.1　通信サービスシステムモデル

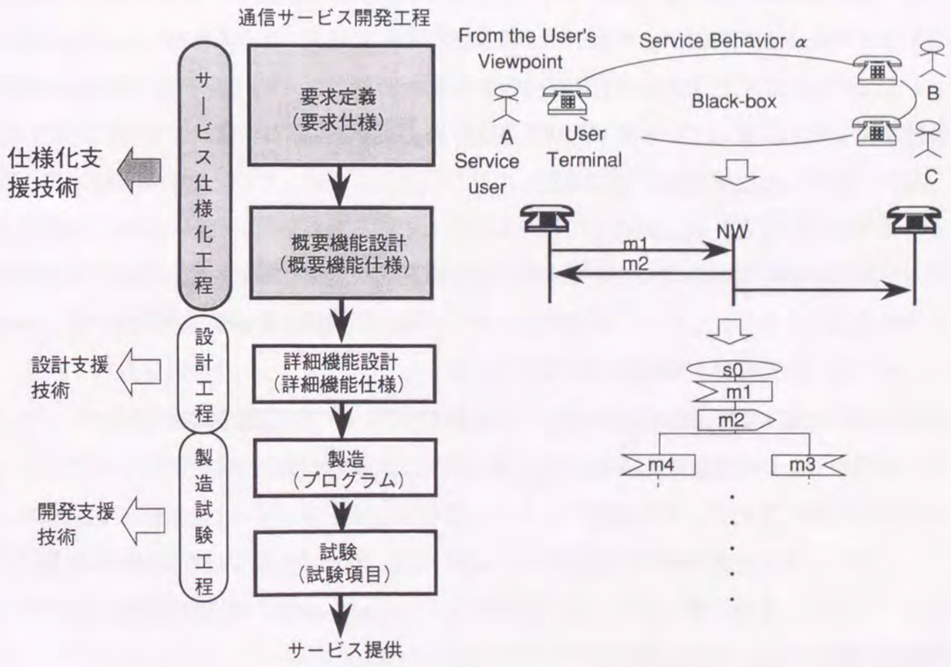


図1.2　位置付け

1.2　要求条件と課題

非専門家による通信サービス要求の仕様化を可能とするための必要な条件について述べる。まず、非専門家としてどのようなスキルを持った人を対象にしているかを述べる。次に、非専門家による通信サービスの要求仕様の設計のあり方について述べ、仕様化支援への要求条件をまとめる[4][5]。

非専門家とは、新しい通信サービスの提案と開発ができる設計者である。しかし、そのような設計者は既存の通信サービス仕様に関する詳細な知識、通信システムの内部構成とその内部動作、プロトコルに関する詳細な知識、更に通信サービスを実現するソフトウェアの開発に関する専門的な知識を必ずしも持つ必要はない。

通信サービスに対する要求仕様は、サービスを利用するユーザの側にたって考えられるべきである。また、その記述仕様は、機械処理を実現するため、形式的な記述言語とする必要がある。しかし、非専門家に対しては、記述の形式性に縛られた違和感を緩和する必要がある。さらに、設計の最初からサービスの正確なイメージを持ち、記述できるとは限らず、設計者の頭の中にあるぼんやりとした断片的な要求を書き下しては、それらを繋ぎ合わせていくような設計スタイルが殆どであるため、記述サービスのイメージが浮かび易く、且つ直感的に記述できるべきである。さらに、曖昧で断片的な設計者の要求記述を完全化するために、抜けている要求記述を補完する必要がある。

また、記述した仕様が設計者の意図した通りの動作を規定しているかの確認が必要となる。専門家（技術者）は、形式的な言語に熟知しているため、記述仕様を目視で辿りながらサービスの動的な振る舞いを頭の中で描き、意図した要求仕様と一致しているかの確認を行うことができる。しかし、非専門家に対しては、形式的な言語を非専門家が直接解釈しなくても、記述仕様の確認ができなければならない。

新しいサービスの記述が完了すると、その新しいサービスの要求仕様が既存サービス仕様と矛盾がないかの検証が必要となる。専門家（技術者）は数多くの既存サービス仕様の詳細まで理解しており、また通信システムに関する知識を持っているため、それらのサービスが同時に提供された状況を想定することができ、サービス競合の発生の有無を検出できる。しかし、非専門家に対しては、既存サービス仕様を詳細に理解しなくても、サービス競合を検出でき、解消仕様を設計できなければらない。

最後に、設計したサービスを規定された網機能配備に基づき通信システムで実行するためのサービス制御ソフトウェアへの詳細化を支援する必要がある。

上記で述べたような設計を支援するための要求条件としては、以下が考えられる。

（1）通信システムの内部動作を隠蔽して、ユーザの視点から仕様が記述できること。

（2）また、記述サービスのイメージが浮かび易く、且つ頭に浮かんだサービス動作が直感的に記述でき、曖昧で断片な記述を許容し、書きたい仕様の何処からでも記述（宣言的記述）ができ、不足部分を補完できること。

（3）一方、記述仕様の機械処理を実現するため、形式的な記述言語に変換できること。

（4）記述内容の確認を支援するため、記述仕様の動的な振る舞いを視覚的に表現できること。

（5）新サービス仕様を既存サービス仕様に追加した場合に、設計者が既存サービス仕様について詳細な知識が無くてもサービス競合を検出でき、且つ解消仕様を生成できること。

（6）サービスを提供するための網機能配備に基づき実行できる通信ソフトウェアに変換できること。

非専門家でも通信サービスに対する要求を正確に反映した仕様を形式的記述として獲得するためには、上記の（1）～（6）の要求条件を達成する必要があり、以下に示す4つの仕様化支援技術に関する課題を検討しなければならない[5]～[7]。図1.3に課題の概要を示す。

課題1：仕様入力

記述の形式性を意識することなく、且つサービス利用者の視点から仕様定義と入力仕様の確認を支援する技術

課題2：仕様統合化

サービス要求記述仕様の合成し易い言語への変換支援技術と独立に記述された複数のサービス仕様間の矛盾を検出し、解消するための検証支援技術

課題3：仕様完全化

設計者が気付かなった仕様を補完し完全化するための支援技術

課題4：仕様詳細化

システムが持っている網機能配備に関する知識を利用しながら獲得した要求仕様からソフトウェア仕様に逐次変換する過程を支援するための技術

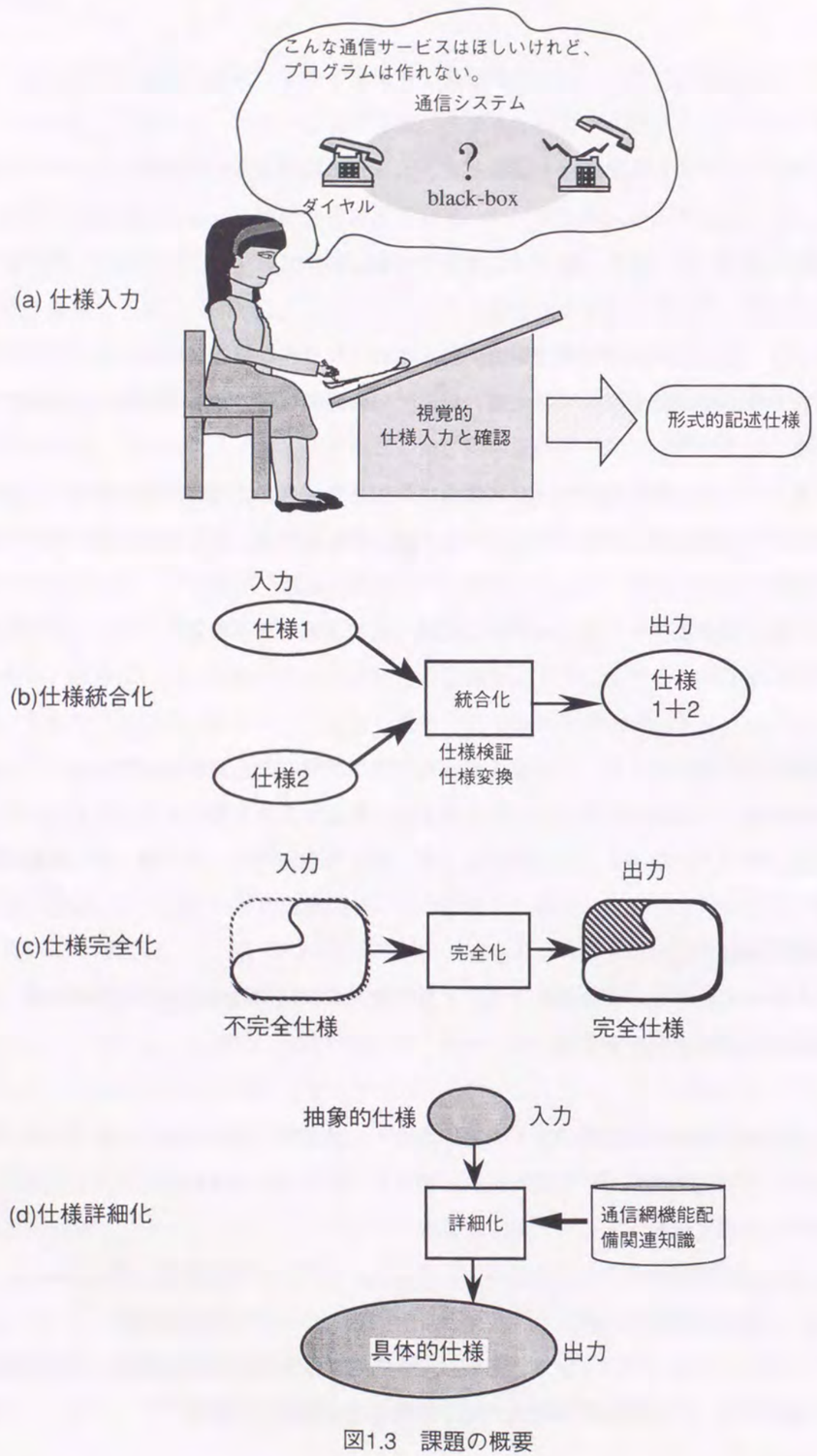


図1.3　課題の概要

1. 3　従来研究と本論文の関係

本節では、本論文で扱う課題に対して、従来の研究との関係を示す。

1. 3. 1　仕様入力との関係

これまでにCCITTやISOにおいて開発された形式的仕様記述言語に基づく仕様入力支援技術の代表的なものとして文献１～３、８に示す手法が提案されている。文献１はSDL/GRで記述した仕様をＣ言語に変換して仮想的に実行させ、その結果を信号シーケンス図等で表示し確認しながら仕様化するシステムである。また、文献２はEstelleに基づくシステムであり、通信システムをOSI参照モデルで定義して仕様を記述し、テキストに変換してシミュレーションにより確認する。文献３は、LOTOSの言語的難しさ（例えば、記述のしにくさや理解の難しさなど）を支援するために、テキストと図形シンボルを相互変換するGLOTOSに関する研究も進められている。文献８では、Petri-Netsに基づく記述と仮想実行のための設計支援環境が提案されている。これらの支援システムの多くは、形式的仕様記述言語をベースとしており、言語に関する知識と通信システムに関する専門的な知識を必要とする設計者がソフトウェア仕様設計段階において主に適用することを前提に実現したものであり、非専門家を対象とした仕様入力支援技術としては、前記文献１～３、８は使えない。

本論文では、非専門家でも通信サービスの要求を仕様化できる支援法として、ユーザの視点からの端末の状態とイベントに基づく状態遷移モデルによる記述法を採用し、仕様記述の図形アイコンによる支援法、記述仕様のアニメーションによる確認法を実現した。

1. 3. 2　仕様統合化との関係

仕様統合化支援技術は仕様検証と仕様変換の２つの技術によって構成され、それぞれについて関係を述べる。

（１）仕様検証との関係

これまでに、サービス仕様レベルでの競合検出法として文献９～１１が提案されている。文献９は、サービスを実現する通信システムの内部処理、プロトコル及び既存サービス仕

様に関する詳細な知識を必要とする方法であり、専門家を対象とした検出支援法である。文献１０では通信システムをブラックボックスとして利用者の視点からサービス仕様をLOTOSで記述する。LOTOSのインタプリタを用いてサービスを設計者がシミュレーションしながら競合を検出する手法であり、機械的には検出できない。文献１１は、サービス仕様をif-then型のルール集合で記述し、ルール競合として状態遷移の非決定性が機械的に検出できる手法である。また、知識処理分野においてルール集合間の矛盾回避、無矛盾保持のためのシステムとしてＴＭＳ、ＡＴＭＳが提案されている[12][13]。更に、ルール集合間の制約をチェックする手法として述語論理等による手法がデータベースの分野で提案されている[13]。しかし、これらはルール集合間の論理的一貫性保持及び論理的矛盾検出のための手法であり、サービス仕様間の意味的矛盾による競合検出法としてはまだ提案されていない。一方、文献１４において、意味的矛盾を検出する手法が提案されている。しかし、本手法は検出するための意味的矛盾を定義した知識を事前に獲得し、設定しておく必要がある。

本論文では、サービス仕様は状態遷移モデルで定義され、既存サービス仕様に新しいサービス仕様を追加した場合に発生する意味的矛盾に着目した機械的なサービス競合検出支援法を提案する。具体的には、意味的矛盾として、既存サービス仕様と新規サービス仕様には定義されていない状態への遷移（以下、「異常な状態への遷移」と呼ぶ）を定義し、その検出法を実現した。また、状態遷移の非決定性に起因するサービス競合の解消支援法を実現した。

（２）仕様変換との関係

通信サービスの仕様記述においては、メッセージシーケンス図（以下ＭＳＣと略す）、ＳＤＬで代表されるような手続き型の仕様記述法[15]~[17]とif-then 型のルール集合による宣言型の仕様記述法[18]が提案されている。しかし、それぞれの仕様記述法は検証能力と機械的支援のし易さの点で差別化されている。

手続き型の仕様記述法であるＭＳＣ、ＳＤＬによる仕様検証では、現状では自動プログラミングの結果を用いた実行可能プログラムに変換したシミュレーションによる動作確認が中心になっており、仕様確認の作業効率の向上が課題である[15]~[17]。これに対して、従来の研究では、個々の仕様のデッドロックやライブロック、到達可能性[17]、マクロ化シーケンス検証、遷移手順検証[19]といったvalidation指向の検証法の提案が中心である。

一方、宣言型の仕様記述法であるif-then 型のルールによる仕様検証では、デッドロック、到達可能性等のvalidation機能の他に、複数のサービス仕様間の矛盾（サービス競合）を調べるverification機能が提案されている[20]~[22]。即ち、新しいサービス仕様を既存の交換システムに追加する場合に必ず検証しなければならないサービス競合検証（既存サービス仕様と新しいサービス仕様を同時に提供した場合に矛盾が発生しないか）を調べることが可能となる。また、if-then 型のルール集合による仕様記述は、実行可能なプログラムに変換しなくても、ルールを解釈実行するインタプリタを用いて逐次ルールを実行させることができる。ルールの記述要素を図形シンボルに変換してその実行状況を表示することによりアニメーションによる仕様確認が容易となる[23]。

このように宣言型の仕様記述法の長所を利用するためには仕様変換が必要となるが、これまでは提案されていない。本論文では、ＭＳＣの手続き型の仕様記述法の長所を損なうことなく仕様検証能力を向上させることを目的として、ＭＳＣで記述された仕様のif-then型のルール集合による仕様への変換法を提案する。

1.3.3　仕様完全化との関係

通信サービス設計を支援するために、仕様化技術、設計法、設計支援環境が研究されている。例えば、文献２４はＭＳＣを使ったオペレーション仕様化法を提案している。ＭＳＣはviews の集合として構成されており、また実行可能で理解し易くなっている。文献２５は様々な仕様表現のためのエディタと検証のめたのプロトタイピィングツールからなるESCORT設計環境を提案した。文献１５はＭＳＣのための仕様化システムＩＥＤを提案した。ＩＥＤは設計者がメッセージを再利用できるためのメッセージ定義データベースを備えている。文献２６は様々な設計エキスパートシステムからなるＡＩに基づく設計システムＫＤＳＳを提案した。これらのエキスパートシステムは既存メッセージ構造を再利用し、仕様を確認するためのプロダクションルールを備えている。文献２７と２８に示すような技術を使うことによって、上記に加え、サービス仕様が完全化されていれば、ＳＤＬ記述を通してプログラムに変換される。しかし、設計者が通信サービスの設計においてまず最初に直面する困難は、手抜かりなく可能なサービス動作を仕様化することである。

これまでには、プロトコル仕様を対象として、補うべき仕様を導出するために、仕様の一般的な正当性概念あるいは正当性を規定する付加的な情報を利用して完全化する手法が提案されている[6]。しかし、設計者がＭＳＣで記述した代表的なサービス動作に対して、

設計者が気付かなかった多数の付随するサービス動作をメッセージ間の関係に着目して仕様を完全化する支援法は研究されていない。本論文では、この問題を解決するために、設計知識として信号規則（ＭＳＲ:Message Sequence Rule）を用いた設計支援法を提案する。

1.3.4　仕様詳細化との関係

仕様詳細化支援技術は仕様から生成されたソフトウェアが実行する通信システムの機能配備法に関する技術とその機能配備に適合したソフトウェアに仕様を詳細化する技術により構成され、それぞれについて関係を述べる。

（1）網機能配備との関係

多種多様を通信サービスを経済的に提供するためには、ＩＮシステムのスケーラビィリティの向上と複数サービスにおいてＩＮシステムを共用できることが要求される。また、アクセス網形態（無線系と有線系アクセス）、サービス種別に依存しない標準インタフェースによるサービス制御が必要なる。しかしながら、これまでのＩＮの網機能配備は、サービスに依存したものであり、カスタマイズ化機能が制限されていた。また、通信のパーソナル化のための網機能は自動車電話、携帯電話、PHS(Personal Handy Phone System)等の移動通信システム[29]の発展に伴い、ネットワークの基本技術として確立されつつあるが、現在提供されているこれらの移動通信サービスはサービスに依存した古い網機能配備に基づくものであり、サービス非依存なアーキテクチャが求められている。また、移動通信サービスのための新しいネットワークアーキテクチャとしてFPLMTS(Future Public Land Mobile Telecommunication Systems）が提案されているが、移動体通信網を主体としたものであり[30]、固定電話網を基盤とした経済的なモビリティ機能を提供するためのアーキテクチャについては検討されていない。

一方、ネットワークのサービス依存性をプラットフォームとして隠蔽し、サービス開発の即応化とサービスのカスタマイズ化等を狙ったネットワークのインテリジェント化の研究も着実に進められ、高度ＩＮシステム[31]として構築されつつある。本論文では、PSTN, ISDN による固定電話網を中心としたパーソナル通信サービスを実現するための高度ＩＮに基づく新しいネットワークアーキテクチャを提案する。

（2）仕様詳細化との関係

これまでは、文献6に示すようにサービス仕様にあらわれるアクションを実行すべきプロトコルエンティティを指定する情報を与えてプロトコル仕様を詳細化する研究が多く行われている。また、文献１５と１６においてサービス仕様からＩＮのプラットフォームで実行できるソフトウェア仕様を詳細化する研究も行われている。この文献では、サービス仕様の記述要素に対応した処理仕様を部品としてデータベースに登録し、変換時に部品のリンクを行う方式であり、部品が多くなると、再利用性を考慮した部品間の関係の把握が課題であった。さらに、本文献でのＩＮのアーキテクチャはネットワークの仮想化をノードと端末の接続パターン(legの組み合わせ）に着目してモデル化し[32]、サービスを制御する手法であり、ITU-T で標準化されているコールセグメントに着目したモデルによるサービス制御とは異なっている。

本論文では、ITU-T の標準に基づく高度ＩＮのサービス制御ポイント（SCP: Service Control Point）上で動作するサービスソフトウェア（ＳＬＰ：Service Logic Processing program）を自動的に生成するために、グローバル状態遷移知識を定義し、その知識を用いたアルゴリズムによる詳細化支援法を提案する。

1.4　論文の構成と各章の概要

本論文は、1.2節で述べた通信サービスの要求仕様化支援技術に関する研究成果をまとめたものであり、以下の8章より構成され、課題と論文の構成との関係を図1.4に示す。

2章は課題1と課題2に対応し、非専門家のための仕様入力と仕様矛盾検証結果確認を支援する視覚的手法を提案する。ユーザの視点からの図形による仕様入力支援における仕様記述モデルとして端末の状態とイベントに基づく状態遷移モデルを採用し、端末状態とイベントの図形シンボルによる仕様記述法、通信サービス仕様の動的な振る舞いをアニメーションにより表現した仕様確認法、及び新規サービス仕様と既存サービス仕様間で検出された状態遷移の非決定性に起因するサービス競合を解消するための解消仕様の生成アルゴリズムと生成仕様の図形表示による支援法を実現した。さらに、本手法に基づく複数サービスの仕様化支援実験結果についても述べる。

3章と4章は課題2に対応しており、3章ではサービス仕様競合の検出支援法を明らかにする。サービス仕様競合として状態遷移の意味的矛盾に着目した検出法を提案する。意味的矛盾として2つのサービスを同時に提供した場合に生成される異常な状態（単独サー

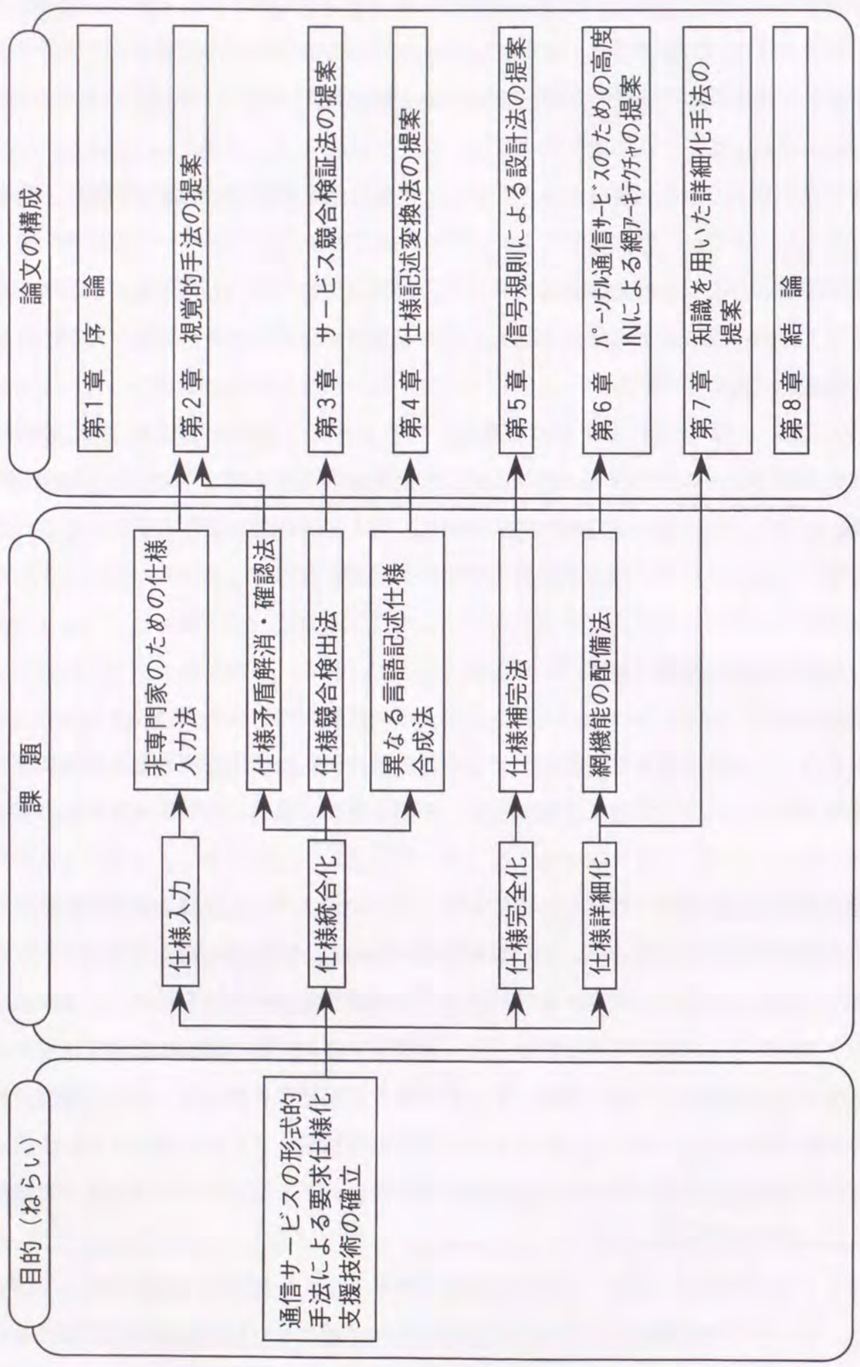


図1.4 課題と論文の構成との関係

ビス仕様には規定されていない状態）への遷移の定義と機械的な検出手法を示す。更に、本手法に基づくサービス仕様競合の机上検出実験結果についても述べる。

また、4章では記述言語の異なる記述仕様との合成を支援するための仕様記述変換法を提案する。本章では、ＭＳＣの手続き型の仕様記述法で記述されたサービス仕様の合成と仕様検証能力を向上させることを目的として、ＭＳＣで記述された仕様のif-then 型のルール集合による宣言型記述仕様への変換法を示す。具体的には、まず、端末と通信網の間のメッセージの時系列として記述されたサービス仕様と端末の状態と端末から入力するイベントに基づくルール集合によるサービス仕様を定義する。次に、2つの仕様間の記述要素間の対応関係を示し、変換を可能とする方法の提案を行う。また、提案した方法を具体的なサービス仕様に適用して評価・考察している。

5章は課題3に対応し、信号規則による仕様補完設計法を提案する。本章では、ＭＳＣによるサービス仕様記述支援を目的として、知識として信号規則を定義し、その信号規則の適用による提案型の設計支援法と仕様の補完法を提案する。メッセージの入出力間の関係を3つのタイプに分類してif-then 型の規則として定義する手法とその規則の適用法を示す。また、信号規則を用いた記述支援法と設計者が気付かなった不足仕様を完全化するためのアルゴリズムを提案する。また、具体的事例により有効性を示すと共に、規則の再利用性について考察する。

6章と7章は課題4に対応しており、6章では、サービスを実現するための網機能配備法をパーソナル通信サービスの観点から述べ、詳細化すべきサービスソフトウェア仕様のネットワークでの位置づけを明確化する。具体的には、有線アクセス系と無線アクセス系を共通にサービス制御できる高度ＩＮによるアーキテクチャを提案する。また、パーソナル通信サービスのためのターミナルモビリティとパーソナルモビリティ、サービスパーソナル化とポータビリティを実現するための高度ＩＮによる制御法を示す。さらに、既存網や他のパーソナル通信網へユーザが移動した場合でも提案アーキテクチャが有効であることを考察する。

7章ではサービス要求仕様から高度ＩＮアーキテクチャに基づくサービスソフトウェア仕様への詳細化手法を提案する。本章では、通信システムをブラックボックスとしてモデル化し、ユーザの視点から定義した通信サービスの要求仕様から、通信サービスを実現する通信システムを端末とネットワーク間のメッセージシーケンスに基づくＭＳＣとして仕様変換し、さらに端末とネットワーク間のＭＳＣによるサービス仕様から高度ＩＮアーキ

テクチャに基づくＳＬＰサービスソフトウェア仕様への変換アルゴリズムと仕様詳細化知識構成法を提案する。

第８章では、本論文で得られた結論を総括する。

第２章
通信サービスの
視覚的な要求仕様記述支援法

2.1　緒言

通信サービスの多様化に伴い、通信サービスを実現するソフトウェア開発の効率化が望まれ、ソフトウェア開発の上流工程において要求仕様を早期に確定することの重要性が認識されている。通信サービスに対する要求を正確に反映した要求仕様を獲得するためには、サービス開発者が要求を直接記述できることが必要となる。

本章では、非専門家でも通信サービスの要求仕様化が図れる支援法について提案する[1)~(18)]。ユーザの視点からの仕様記述モデルとして端末の状態とイベントに基づく状態遷移モデルによる記述法を採用し、仕様記述の図形による支援法、サービス仕様の動的な振る舞いをアニメーションにより表現した仕様確認法及び状態遷移の非決定性に起因するサービス競合の解消支援法を実現した。提案手法に基づく要求仕様化支援環境を試作し、記述実験による評価と考察を述べる。

2.2　図形による仕様記述

2.2.1　サービス要求仕様

まず、本論文におけるサービス要求仕様の記述モデルについて定義する。サービス要求仕様は、1.2節で示した要求条件（1）を達成するため、通信サービス動作を、通信システムの内部動作には立ち入らないで、ユーザの視点からの端末状態とイベントからなる状態遷移モデルにより記述する。これは、通信システムに関する詳細な知識のないサービス開発者でもサービスが記述できる環境及び通信サービスソフトウェアの設計の初期段階において、ユーザ要求を正確に反映した通信サービス仕様として厳密に記述することを目的としている。図2.1は、基本電話サービスにおいてユーザの視点に立ってサービスを捉え

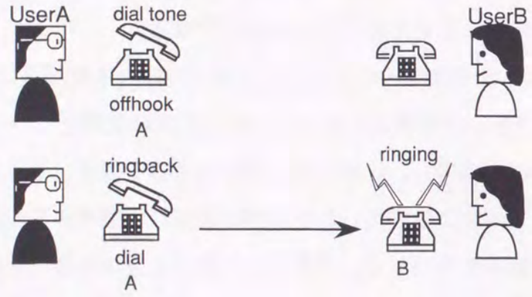


図2.1　ユーザの視点からの端末状態の例

た場合の一部の例を示している。ユーザAがユーザBと通信したい場合、ユーザAはまず最初に受話器あげる。受話器を上げるとダイヤルトーンが聞こえてくる。次にユーザAはユーザBに対してダイヤルすると、ユーザBには呼出音が鳴り、ユーザAには呼び返し音が聞こえる状態となる。このようにユーザが実行する動作を“イベント“と定義し、電話機の状態を“端末状態“と定義して、サービスに関係する全ての端末の状態を考慮した状態遷移を基本として記述する。このような記述法においてサービス要求仕様は、以下のように定義される。

定義１：サービス要求仕様

RQ-SPEC ＝＜T, S, E, Φ, s_0, f_0＞

ここで、

a）Tは端末の集合である。要素をt_iで表す。

b）Sはサービスの状態s_iの集合であり、s_iは端末の状態の集合として表す。端末tの状態を$\tau(t)$で表し、サービスの状態s_iに含まれる端末数をnとすると、$s_i = \{\tau(t_1), \tau(t_2), \cdots\cdots, \tau(t_n)\}$ となる。また、端末状態$\tau(t)$は、より細分化された状態である状態記述要素pの集合として定義される。端末の状態は、以下に示す2種類の状態記述要素pを組み合わせて表す。

・$p(t_i)$:端末t_iの状態は他の端末と関係が無いことを示す。

・$p(t_i, t_m)$:端末t_iの状態は端末t_mと関係があることを示す。

c）Eは端末から入力されるイベントe_iの集合である。e_iは、以下の2種類がある。

・$e(t_i)$:端末t_iから入力されたイベントを示す。

・$e(t_i, t_m)$:端末t_iからt_mに対して入力されたイベントを示す。

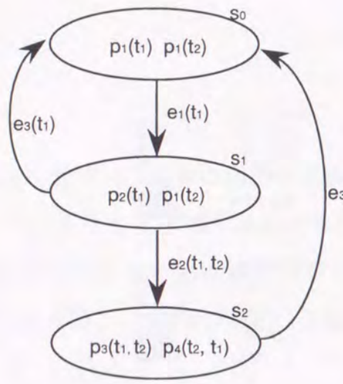
d）Φはサービス状態間の遷移関数である。

$s_j = \Phi(s_i, e_i)$

e）s_0は初期状態である。　$s_0 \in S$

d）f_0は最終状態である。　$f_0 \in S$　□

図2.1の定義1に基づく状態遷移の例を図2.2に状態遷移図で示す。サービスの状態は2つの端末t_1、t_2の状態から構成され、状態s_1とs_0は端末間の関係が存在しない端末状態から構成されており、状態s_2は端末t_1とt_2がそれぞれ相互に関係している状態から構成されている。



（注）s_1とs_2が図2.1に対応し、その意味は以下のとおり

p_1 = idle　　e_1 = offhook
p_2 = dial-tone　　e_2 = dial
p_3 = ringback　　e_3 = onhook
p_4 = ringing

t_1 = A
t_2 = B

図2.2　端末の状態とイベントに基づく状態遷移モデルによる記述例

2.2.2　図形による仕様記述法

1.2節で示した要求条件（2）を満たすためには、初期状態から遷移に沿って順次、正確に記述することが必要な手続き的な記述よりも、断片的な要求をそのまま宣言的、且つ頭に浮かんだ状況に出来るだけ近い形式で直感的に記述できる手法が望ましい。そこで、通信サービスの宣言的で直感的な仕様記述を支援するため、図形による記述法を提案する。

（1）端末状態とイベントの図形表記法

5つの既存電話サービス（POTS, CW, CFV, TWC, CCBS：各サービスの簡単な説明を付録2.1に示す）を分析し、図形シンボルとして表2.1に示す4つの種別を定義した。種別1と3は状態を表すシンボルであり、種別1はその端末の状態が他の端末と関係のない単独の状態であることを意味している。また、種別3はその端末の状態が矢印で示す他の端末と関係があることを意味している。種別2と4はイベントを表すシンボルであり、種別2はその端末から入力されるイベントを示す。種別4はその端末から矢印で示す端末に対して入力するイベントを示す。

（2）図形による状態遷移記述

仕様記述は、宣言的な記述を支援するため、図2.3に示すような1つの状態遷移を単位として行う。図形による状態遷移記述は、表2.1で定義した図形シンボルを用いて、現状態とイベントを記述する画面と次状態を記述する画面の2つに分割して行う。表2.2の項番1は2つの端末の場合の記述例であり、現状態記述画面には、端末aとbがあり、それらの状態が、それぞれp_1とp_2であり、その時に端末aからbにイベントe_1が入力された状況を記述している。また、次状態は遷移により端末aとbの状態がそれぞれp_3とp_4になり、それぞれの状態は相互に関連があることを示している。項番2は端末数がm個の場合の1つの記述例である。

（3）テキストへの変換

1.2節で示した要求条件（3）に対して、仕様検証、仕様詳細化等の他の支援系への入出力を可能し、支援機能のシステム化を容易とするため、図形記述とテキスト記述の相互変換を可能とする。

変換するテキスト形式は、1つの状態遷移を単位としたif-then型のルールとする。ルールの条件部は現状態csとイベントeとし、動作部は次状態nsとする。以下にルールのフォーマットを示す。

表2.1　図形シンボル

class	graphical symbol	meaning	converted text
1	a p	端末aの状態が他の端末と関係のない状態記述要素pで表わされる。	p(*a*)
2	e: a	端末aからイベントeが入力された。	e(*a*) :
3	a　p　b	端末aの状態が他端末と関係のある状態記述要素pで表わされる。	p(*a*,*b*)
4	a　e:　b	端末aから他端末bに対してイベントeが入力された。	e(*a*,*b*) :

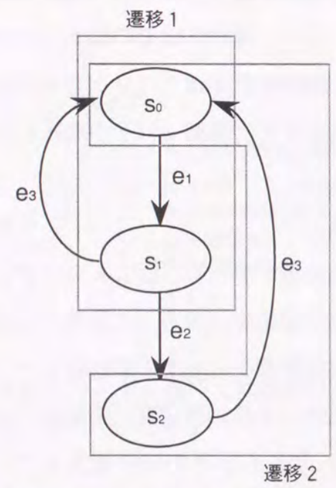


図2.3　状態遷移単位の宣言的記述

表2.2　図形による状態遷移記述例

case		graphical description		converted text
		current state	next state	
1	2 端末	a　e1:　b p1　p2	a　p3　b a　p4　b	p1(*a*), p2(*b*) e1(*a*,*b*): p3(*a*,*b*), p4(*b*,*a*)
2	m端末	a　p1　b p2 e2: m pn	a　p3　b pl pk m	p1(*a*,*b*), p2(*b*,*a*) ---, pn(*m*) e2(*m*,*a*): p3(*a*,*b*), ---, pl(*m*,*a*), pk(*a*,*m*)

r）cs　e：　ns.

ルールの条件部と動作部はコロン（：）で区切り、csとnsは状態記述要素ｐを用いて記述する。表2.1の図形シンボルは、その表2.1にconverted textとして示すように、端末識別子をパラメータとしたテキストに変換する。そして、図形により記述された1つの状態遷移は、表2.2のconverted text欄に示すようなテキストに変換される。

このようなルールと定義1で示した遷移関数Φとの関係は、以下のように定義することができる。

定義2：ルールによる状態遷移記述

サービス要求仕様RQ-SPEC の遷移関数Φは以下のように定義することができる。

Φ＝＜R，Σ＞

ここで、

a）Rはルールの集合を示す。要素をｒで表す。ｒはif-then型のルールであり、条件部と動作部をコロン（：）で区切り、以下のように記述する。

ｒ＝cs　　e：　ns.

cs：現状態を表し、端末の状態記述要素ｐを用いて記述する。

ns：次状態を表し、端末の状態記述要素ｐを用いて記述する。

ｅ：遷移の起因となるイベントを示す。ｅ∈Ｅ

b）Σはルールの適用関数であり、以下の2つの関数から構成される。ここでs_iとs_jも状態記述要素ｐを用いて表現される。

ｒ＝Ｍ（s_i，ｅ）：

関数Ｍはルールのマッピング関数であり、イベントｅが選択された時にサービス状態s_iに対して、ルールｒの条件部csが$s_i \supseteq$ csを満足するルールを抽出する。尚、サービス仕様が確定した時点においては、ルール競合は解消され、唯一のルールが決定される。

s_j＝Ｗ（ｒ，s_i）：

関数Ｗはルールｒが適用された結果、s_iをそのルールの動作部に記述されている内容に従って書き換え、次状態s_jを生成する関数である。　□

2.3　アニメーションによる仕様確認

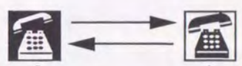
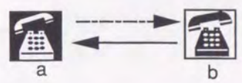
記述した仕様が設計者が意図した通りのサービス動作となっているかを確認する必要がある。図形により記述した状態遷移を繋ぎ合わせて、状態遷移図のように表現する事もできる。しかし、複雑なサービス仕様の場合には、状態を構成する端末数、遷移数が増加するため、複数の端末の状態によって構成されるサービスの状態の意味を即座に理解することは難しく、状態遷移図の全てを画面に表示して、目視により確認することも困難になると考えられる。そこで、1.2節で示した要求条件（4）に対応した通信サービス記述仕様の確認を支援するため、アニメーションによる視覚的な手法を提案する。サービス要求仕様のアニメーションは仕様記述において定義した端末状態を表す図形シンボルをよりリアル化したシンボル（以下ではこれを、記述の時に定義した図形シンボルと区別するため絵シンボルと呼ぶ）を規定し、それらを記述仕様の解釈実行により組み合わせて、逐次表示することより実現する。以下では、端末状態の絵シンボルとアニメーション表示アルゴリズムについて述べる。

2.3.1　端末状態の絵シンボル

アニメーションによる仕様確認が、様々なサービス仕様に対して柔軟に適用できるように、我々は既存の5つのサービス仕様（POTS、CW、TWC、CFV、CCBS）を分析し、端末状態の絵シンボルとして、端末の複数の状態を代表する6つの絵シンボルを定義した。その絵シンボルを表2.3に示す。絵シンボルは電話の状態を示した電話画素と端末間の関係を表示した矢印画素により構成されている。これらの絵シンボルを組み合わせて端末の状態をアニメーションで表現する。

絵シンボルは他の端末と関係のない端末状態と他の端末と関係のある端末状態に分けて構成した。他の端末と関係のない端末状態の絵シンボルは、電話画素でのみ構成され、受話器を置いている状態と上げている状態を区別するため、それぞれに対応した絵シンボルを（表2.3のts_1, ts_2）を定義した。また、他の端末と関係のある端末状態の絵シンボルは、電話画素と矢印画素から構成され、呼設定状態と通話状態が区別できる絵シンボルを定義した。更に、呼設定状態は電話機を使っていない空き状態での着信時の呼出状態と通話中での呼び出し状態が区別できる絵シンボル（表2.3のts_3, ts_4）を定義した。また、

表2.3　端末状態を代表する絵シンボル

symbol			meaning
他の端末と関係のない端末状態	ts_1	 a p	onhook state
	ts_2	 a p	offhook state
他の端末と関係のある端末状態	ts_3	 p1 p2 a b	call establishing state
	ts_4	 p1 p2 a b	alerting state during talking
	ts_5	 a b	talking state
	ts_6	 a b	holding state

（注）a,b：端末識別子、p：状態記述要素

通話中状態では、相互に通話している状態と通話を保留している状態が区別できる絵シンボル（表2.3のts_5, ts_6）を定義した。

2.3.2　アニメーション表現アルゴリズム

記述した仕様の解釈実行は、図形記述からテキスト形式に変換したルールを逐次適用することより実現する。アニメーション表現は、適用したルールの次状態記述に従って、絵シンボルを書き換えることにより行う。表現法の概要を図2.4に示す。

次に、サービス仕様のアニメーション表現アルゴリズムを説明する。

＜アニメーション表現アルゴリズム＞

ステップ1：初期状態

空白の画面が表示される。この画面に絵シンボルをサービス仕様確認に必要な数だけ、設計者が配置し、初期状態（s_0）を構成する。この時、ワークメモリ上にもテキスト形式での初期状態（s_0）が作成される。

ステップ2：ルール適用と次状態（s_1）生成

設計者がある絵シンボルの電話画素をマウスでクイックすると、図形による状態遷移記述から変換されたテキスト形式のルールの中から、ルールの現状態が初期状態に含まれるルールを選択し、その選択されたルールに含まれるイベント名を画面に表示する。設計者がその表示されたイベントの中から1つを選択するとそのイベントを含むルールが適用され、ワークメモリ上にそのルールの次状態に記述されている内容で、初期状態を書き換え、サービスの次状態（s_1）を生成する。

ステップ3：絵シンボルへの変換

表2.3の絵シンボルとその意味に対応した状態記述要素との対応表に基づき、サービスの次状態（s_1）を絵シンボルに変換する。

設計者がイベントを選択するごとに、ステップ2と3が繰り返し行われ、アニメーション表現される。設計者はイベントを逐次入力しながら記述した仕様を確認していく。また、入力するイベント系列を事前に記述することによりサービス動作の一連の流れを自動的に実行させ、確認することができる。

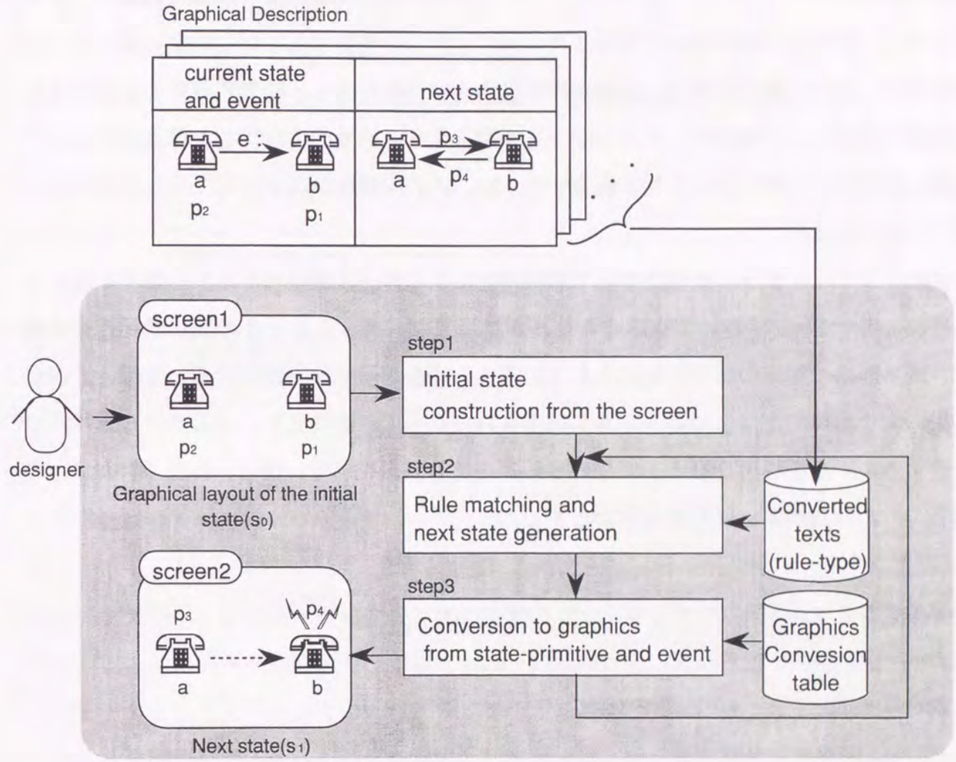


図2.4　サービス仕様のアニメーション表現法

2.4　サービス競合解消

新しいサービス仕様の設計が終了すると、既存サービス仕様との矛盾（サービス競合）を検出して、解消する必要がある。サービス競合の１つである状態遷移の非決定性をルールで記述した仕様を合成して、ルール適用の非決定性を検出することにより検出する手法が既に提案されている[19]。検出したサービス競合の解消法は重要な問題であるが、文献１９では、非決定な状態遷移の解消法については今後の課題として残されている。そこで、本稿では、検出された非決定な状態遷移の解消ルールを生成し、図形表示することにより、設計者に非決定な状態遷移のどちらか一方を選択させ、自動的に検出した状態遷移の非決定性を解消する支援手法について提案する。尚、この支援手法は1.２節で示した要求条件（５）に対応する。

文献１９では、ルール適用法として詳細優先によるルール適用法が用いられている。これは、ある状態s_iにおいて複数のルールが適用可能な時、各ルールの現状態記述を比較し、最も詳細に端末の状態が記述されているルールを適用する方法である。文献１９の検出法により、非決定な２つのルールが検出されたとして、詳細優先によるルール適用法に合致するルールを解消仕様として生成するアルゴリズムを示す。但し、検出された２つの競合ルールの現状態は同じではないとする。

＜解消アルゴリズム＞

入力：

競合ルールr_1とr_2が入力される。

r_1）cs_1　e：　ns_1.

r_2）cs_2　e：　ns_2.

ここで、cs_i，ns_i（ｉ＝１，２）は状態記述要素ｐで構成されている。また、$cs_1 \neq cs_2$とする。

出力：

r_1とr_2のどちらか一方を優先する解消ルールとして、$r_{1>2}$と$r_{2>1}$を出力する。

ステップ１：解消ルールの現状態の生成

r_1とr_2の現状態の和集合をとり、$r_{1>2}$と$r_{2>1}$の現状態cs_3を作成する。

$cs_3 = cs_1 \cup cs_2$

ステップ２：解消ルールの次状態の生成

ステップ１で生成した現状態から遷移し、r_1とr_2の次状態を満足するような$r_{1>2}$と$r_{2>1}$の次状態ns_{11}，ns_{21}を生成する。

$ns_{11} = ns_1 \cup (cs_2 - (cs_1 \cap cs_2))$

$ns_{21} = ns_2 \cup (cs_1 - (cs_1 \cap cs_2))$

ステップ３：解消ルール$r_{1>2}$と$r_{2>1}$の生成

ステップ１と２で生成した現状態、次状態より以下の解消ルールを構成する。

$r_{1>2}$）cs_3　e：　ns_{11}

$r_{2>1}$）cs_3　e：　ns_{21}

生成した解消ルールを絵シンボルと状態記述要素の対応表に基づいて変換し、表示する。図形により表示することにより競合状況と個々の競合するルールを優先させた時の次状態の状況が直感的に設計者は理解できる。

2.5　仕様化支援環境

ワークステーション上に要求仕様化支援環境のプロトタイプを試作した。解釈実行のための言語として我々が開発したＳＴＲ（state transition rule)言語[20]を、ユーザインタフェースとしてＸウインドのMotif ライブラリを使用しSUN4/SS2上に実現した。

図2.５に図形による仕様記述例を示す。記述画面は、現状態とイベントを記述するウインドと次状態を記述するウインドの２つに分割されている。図2.５はCWサービスの例であり、３端末が配置されている。現状態記述ウインドには、端末ＢとＣは通話中で端末Ａから端末Ｂにダイヤル（dial)した状況を記述している。端末状態を表す状態記述要素とイベントは、事前に登録し、それらをサブウインドからマウスで選択することにより記述する。次状態記述ウインドには、CWサービスの特徴である通話中でも呼び出し音により通話者に着信を通知することを示す状態が記述されている。つまり、端末Ｂは通話中（path）に端末Ａから呼び出され（cw-ringing）、端末Ａは呼び出し音（ringback）が鳴っている状態を記述している。

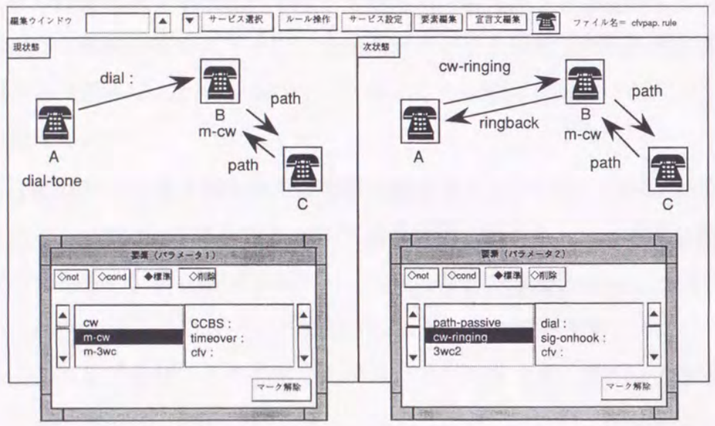


図2.5　図形によるサービス記述例（話中着信 (CW) サービス）

サービス仕様をアニメーション表示する画面例を図2.6に示す。本プロトタイプで定義した状態記述要素pから絵シンボルに変換するための対応表の例を付録2.2に示す。

2.6　評価及び考察

2.6.1　記述実験

ベルコア社のLSSGR[21]、CCITT[22]の付加サービスして規定されているサービスから28種類を選択し、プロトタイプを使用して記述実験を行った。記述した状態遷移数は全サービス合わせて約600遷移となった。図形仕様記述により個々のサービスを単独に記述し、アニメーションにより記述仕様の正当性とサービスの振る舞いの充足性を確認した。

2.6.2　図形による仕様記述

（1）直感的記述

プロトタイプを使用した主観的な評価では、システム設計段階で予想されたように、端末とその関係に基づき直感的に仕様を記述することができ、テキストによる仕様記述に比べ優れた利点として確認できた。

（2）サービス動作の網羅性

記述したサービスをCCITTの勧告[22]の基本サービスと付加サービスの規定を参照し、表2.4のように分類した。基本サービスは、2者間の通信を提供するものであり、POTSに代表される。番号識別サービスとしては、番号がサービスの多様性の要素となるものを分類し、発番号の着端末への通知や通知の禁止及びダイレクト接続等とした。呼提供サービスとしては、着信者が応答する前の呼の扱い方により分類されるものであり、着信転送、代表等が該当する。呼完了サービスは、着信呼の接続の仕方に関するサービスであり、コールウエイテングや話中時再呼出等がある。複数通話サービスは、複数の通話者により実現されるサービスであり、会議通話や3者通話がある。特定グループサービスは、通話者により1つのグループを構成するサービスであり、閉域等が該当する。情報提供サービスは、通信に関する詳細情報（接続期間、通話先等）を通知するサービスである。表2.4よ

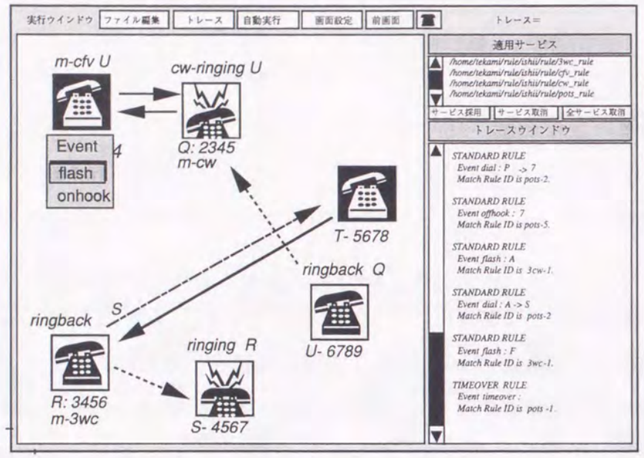


図2.6　アニメーション表示の画面例

表2.4　記述実験した電話サービスの分類

classification of service		number	example
Basic Service		1	POTS
Supplementary service	Number Identification	11	Calling Number Delivery
	Call Offering	6	Call-Forwarding Variable
	Call Completion	3	Call-Waiting
	Multi party	2	Three-Way Calling
	Community of interest	3	Closed User Group
	Information offering	2	Customer Premises Message Resisters
total		28	

り、既存サービスに含まれる特徴的な振る舞いが図形によるサービス仕様記述により充分記述できていることが分かった。

（3）図形シンボルの汎用性

表2.1で定義した4つの種別の図形シンボルを組み合わせて、28種類のサービス仕様は全て記述できた。これは、規定した図形シンボルが、端末の状態とイベントに基づくサービス仕様の記述ための汎用的な図形シンボルであることを示している。

（4）端末数制限

本実験でのサービス仕様記述に必要な端末数の分布を図2.7に示す。図2.7から、端末数は5端末以下で記述できることが分かり、端末数の増大による記述の煩雑さはなく、図形による仕様記述の有効性を損なわれない。

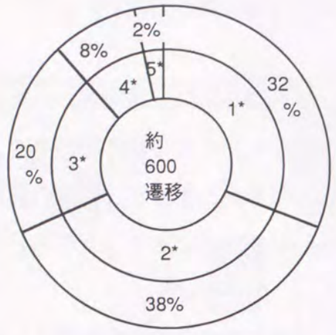
2.6.3　アニメーションによる仕様確認

（1）仕様の理解性

プロトタイプを使用して記述したサービス仕様を逐次実行させることにより、サービスの動作がアニメーションで表示され、仕様の理解性では、従来の手法に無い利点が得られた。例えば、CWサービス動作のアニメーションを図2.8に示す。図2.8(a)は端末QとRが通話中で、端末QがCWサービスを契約している時に、設計者が端末Pの電話アイコンをマウスでクリックしてdialイベントを選択し、端末Qに着信しようとしている状況を示している。dialイベントを選択すると図2.8(b)に画面が変わる。図2.8(b)は端末Qが通話中に呼び出されて、端末Qのflashイベントを設計者が選択すると図2.8(c)に画面が変わる。図2.8(c)は端末Qは端末Rとの通話を保留し、端末Pと通話している状況を示している。このように、図2.8の(a)〜(c)はCWサービスの特徴的な動作を示しており、付録2.3に示したテキスト形式の記述よりも明らかに容易にそのサービス仕様が理解できる。

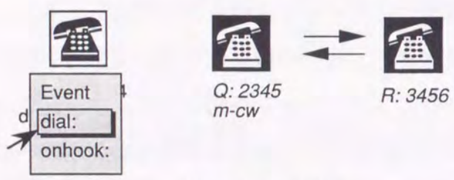
（2）仕様の検証性

サービス動作をアニメーションとして表示することにより、設計者が目視によりテキスト記述された仕様を検証する場合に比べ検証し易くなるという利点が得られた。例えば、転送（CFV）サービスと着信拒否（TCS）サービスを合成した場合の異常な遷移の検証について説明する。CFVサービスは、転送先端末を登録していると、その端末に着信があった場合、登録されている端末に呼びを転送するサービスである。TCSサービスは、登録し

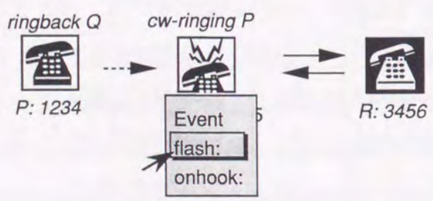


* : the number of terminals per state transition

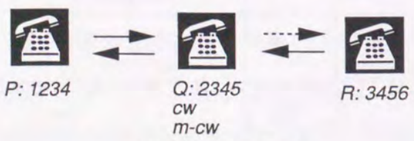
図2.7　図形によるサービス記述における端末数の分布



(a) 端末QとR通話中



(b) 端末Q通話中呼出し



(c) 端末PとQ通話中

図2.8　アニメーションによる仕様確認例

ている端末からの着信を拒否するサービスである。CFVとTCSサービスを合成した場合、アニメーションを逐次実行していくと、TCSサービスで禁止している動作（登録端末からの着信）が、CFVサービスの影響で可能となるような異常な状態に到達する。この異常な状態に到達した画面を図2.9に示す。図2.9は、端末Qが端末Rを転送先として登録しおり（m-cfv R）、端末Rは着信拒否端末として端末Pを登録している（m-tcs P）時に、端末Pが端末Qにダイヤルすると端末Rに転送され、着信を拒否している端末Pから端末Rが呼出されている状態へと遷移した状況を示している。このように、アニメーションで表示することによりこの異常な状態を容易に理解することができ、且つこの状態に到った遷移を把握することができ、仕様の検証にとって有効であることが示された。

（3）絵シンボルの汎用性

記述実験した全てのサービス仕様をプロトタイプを使用して実際に実行させて、表示される状況を確認した。アニメーション表現のために定義した表2.3の絵シンボルは、記述実験した全てのサービス仕様で適用でき、端末状態とイベントに基づく仕様記述の確認のための汎用的な絵シンボルであることが示された。

2.6.4　サービス競合解消

（1）図形表示の有効性

提案したアルゴリズムで生成された解消ルールを絵シンボルに変換して表示することにより、競合状況が直感的に理解でき、また生成された解消ルールの意味が理解し易くなり、従来にない利点が得られた。例えば、CWとCFV サービスを合成した時、付録2.4に示すSTR言語で記述した2つのルール（r_{CW}とr_{CFV}）が競合することが文献19において検出されている。競合状況と競合ルールについては付録2.4に詳細に示す。付録2.4に示すr_{CW}とr_{CFV}から提案したアルゴリズムにより競合を解消するルールを生成すると以下のようになる。

```
r_CW>CFV = dial-tone(A), not [idle(B)] ,
           path(B,C), path(C,B), m-cw(B),
           m-cfv(B,D), idle(D)
           dial(A,B):
           ringback(A,B), cw-ringing(B,A),
```

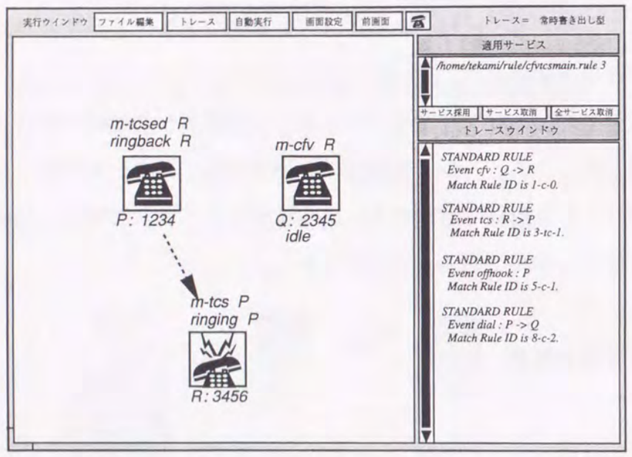


図2.9　異常な状態への遷移の表示例

r-path(A,B),m-cw(B),m-cfv(B,D),

path(B,C),path(C,B).

$r_{CFV>CW}$=dial-tone(A),not [idle(B)] ,

path(B,C),path(C,B),m-cw(B),

m-cfv(B,D),idle(D)

dial(A,B):

ringback(A,D),ringing(D,A),

r-path(A,D),m-cw(B),m-cfv(B,D),

path(B,C),path(C,B),not [idle(B)] .

生成された$r_{CW>CFV}$と$r_{CFV>CW}$を絵シンボルに変換して表示すると図2.10のようになる。設計者は、上記のテキスト形式による記述から競合解消ルールの意味を理解する場合に比べ、図形表示された解消ルールの方が、その意味を容易に理解できる。従って、何方かの選択が可能となり、状態遷移の非決定性によるサービス競合の解消支援として有効である。

（2）解消ルール生成アルゴリズムの正当性

文献7の競合検出実験によって検出された5サービス（POTS, CW, CFV, TWC, CCBS）を組み合わせた時の競合検出結果を付録2.5に示す。この結果に基づきルール競合の解消実験を提案したアルゴリズムにより行い、解消ルールが生成でき（付録2.6）、図形によりその状況が表示できたことを確認した。また、表示された解消候補から優先させたい遷移先を選択することにより競合を解消することができた。

2.6.5　提案支援法の有用性

提案した支援法は、電話交換サービスを対象としており、公衆電話交換システム及びPBXを中心とした私設交換システムのサービス開発支援として適用できる。また、このような交換システムのサービス開発は、これまでは専門的な知識をもった技術者によって主に行われてきたが、提案システムによってそのようなサービス開発の可能性を非専門家にまで拡大できる。但し、上記で述べたようなシステムとして実現するためには、獲得した要求仕様からソフトウェアを自動的に作成できる技術が必要となる。このようなソフトウェアの自動作成技術の検討は現在進めており、統合したシステムの提案は別途行う予定である。

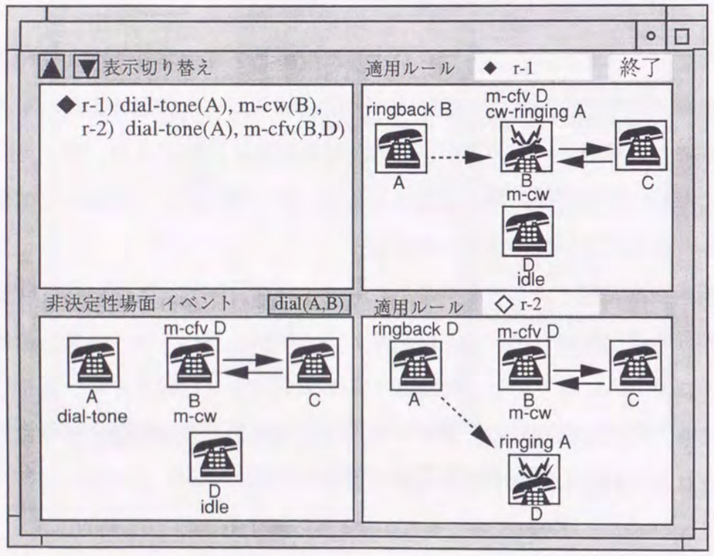


図2.10　サービス競合解消支援例

一方、端末の機能は、既存の電話サービスで使用されている機能を想定して、2.6.2節で述べたように基本サービスと6種類のカテゴリーからなる付加サービスの合わせて28種類の既存電話交換サービスで適用できることは確認した。しかし、近年の端末の種類の多様化により、端末のディスプレイへの電話番号や通話料金の表示、音声や音によるガイダンス等を利用したサービスが実現されている。このような多機能端末を利用したサービスにおいても、ディスプレイに電話料金が表示されている状態記述要素、通話料金が表示されている状態記述要素、等を新たに追加し、それらの状態記述要素を組み合わせて端末の状態を定義することにより、記述できる。しかし、そのような具体的なサービスの記述実験を行うことにより確認を行う必要がある。更に、計算機をネットワークに接続して音声以外のデータ通信サービスやマルチメディア通信サービスへの提案手法の適用については、今後の課題である。

2.7　結言

通信システムの内部動作を考慮しなくても、端末の状態とイベントだけにより、サービス要求を仕様化できる支援手法として、以下を提案し、試作により有効性を確認した。

（1）ユーザの視点からの直感的で宣言的な仕様記述を実現するためのに、端末の状態とイベントの図形シンボルの定義とそれらを用いた図形による仕様記述法。

（2）通信サービスの動的な振る舞いをアニメーションにより表現した仕様確認法。

（3）新規サービス仕様と既存サービス仕様間で検出された状態遷移の非決定性に起因するサービス競合の解消を図るための解消仕様の生成アルゴリズムと生成仕様の図形表示による支援法。

第３章
通信サービスの要求仕様検証支援法

3.1　緒言

情報社会の進展に伴い、多種多様な通信サービスが要求されており、サービスの迅速な開発と提供が求められている。通信サービスの開発は要求仕様を明確化し、その仕様を実現したソフトウェアにより提供される。新サービスの追加にあたっては、新しい通信サービスの仕様を規定し、その新サービスを通信システムに追加した場合の正常動作を検証する必要がある。この検証は開発工数を考慮すると、なるべく上流工程で行うことが望ましい。サービス数の増加に伴い新しいサービスと既存サービス間におけるサービス競合を人手によって検出し、解消することは益々困難となり、コンピュータの支援による機械的な手法を確立する必要がある[(1)(2)]。

本章では、サービス仕様は状態遷移モデルで定義され、既存サービス仕様に新しいサービス仕様を追加した場合に発生する意味的矛盾に着目したサービス競合検出支援法を提案する[(3)~(10)]。具体的には、意味的矛盾として、既存サービス仕様と新規サービス仕様には定義されていない状態への遷移（以下、「異常な状態への遷移」と呼ぶ）を定義し、その検出法を提案する。異常な状態への遷移を機械的に検出するための仕様記述言語として状態遷移をif-then型のルール集合を採用し、異常な状態への遷移の検出アルゴリズムを提案する。具体的なサービス仕様記述例に基づき、提案手法を説明し、提案した異常な状態への遷移の検出に基づくサービス競合検出の机上実験結果と考察を示す。

3.2　異常な状態への遷移の定義

本章では、異常な状態への遷移を定義するための前提条件について説明し、定義を与える。

3.2.1　サービス仕様競合検証支援と前提条件

異常な状態への遷移の検出法の概要を図3.1に示す。新規と既存の２つのサービス仕様を機械的に合成し、２つの被合成サービスと合成サービスのそれぞれの状態遷移を比較・判定しながら仕様競合を検出する。尚、本稿で用いる「合成」とは、２つのサービス仕様を機械的に組み合わせる過程までのことを意味する。この過程では、生成されたサービス仕様の矛盾は、まだ解消されていない。次に、図3.1に示すような検出法を実現するためのサービス仕様の前提条件を以下に示す。

（１）サービス仕様は状態遷移モデルに基づく仕様記述法で記述されている。

（２）サービスを合成する前の各サービス仕様は共に正常な状態遷移をすることが設計者により確認されている。例えば、個々のサービス仕様において状態遷移の非決定性、デッドロック、ループ等の論理的矛盾は解消されている。

3.2.2　異常な状態への遷移

異常な状態への遷移を検出するための前提となるサービス要求仕様は2.3節で示した定義１に基づき記述されるものとする。本節では、「異常な状態への遷移」を定義し、具体例により説明する。

２つの単独サービス仕様、Ｘサービス仕様とＹサービス仕様からＸとＹサービスの合成サービス仕様を設計する場合を考える。ＸサービスとＹサービスをそれぞれの仕様に基づき実行させ、図3.2に示すようなＸとＹサービスが重なり合うような状態s_iに遷移したと仮定する。このような状態において、ＸとＹの何方か一方のサービス、例えばＸサービスを実行させたとすると、新たに重なりあった次状態s_jが生成される。この次状態におけるＹサービスに関連した状態s_{yj}は、Ｙサービス仕様に規定されていない可能性がある。このＹサービスに関連した状態s_{yj}がＹサービス仕様に規定されていない場合、次状態においてＹサービスを実行できない。このように、２つのサービス仕様を合成して新たなサービス仕様を設計する場合、一方のサービス仕様の実行により、他方のサービスの実行が妨害されるような状態遷移を「異常な状態への遷移」とし、以下のように定義する。

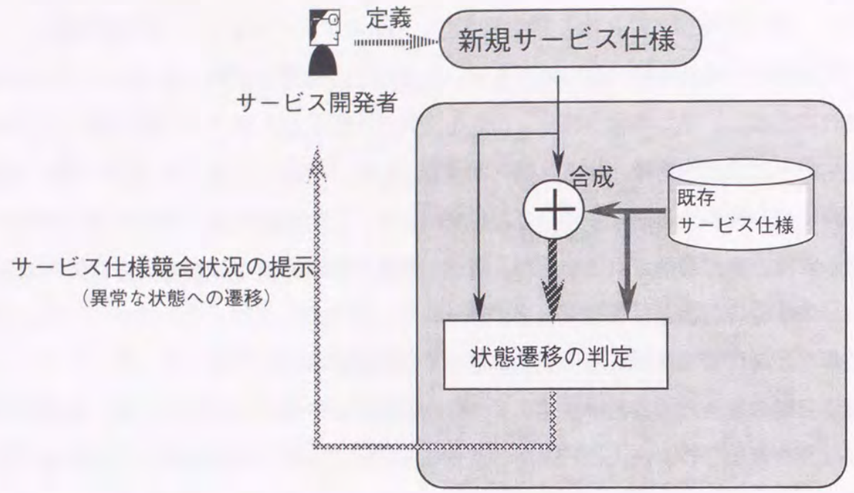


図3.1　サービス仕様競合検出の概要

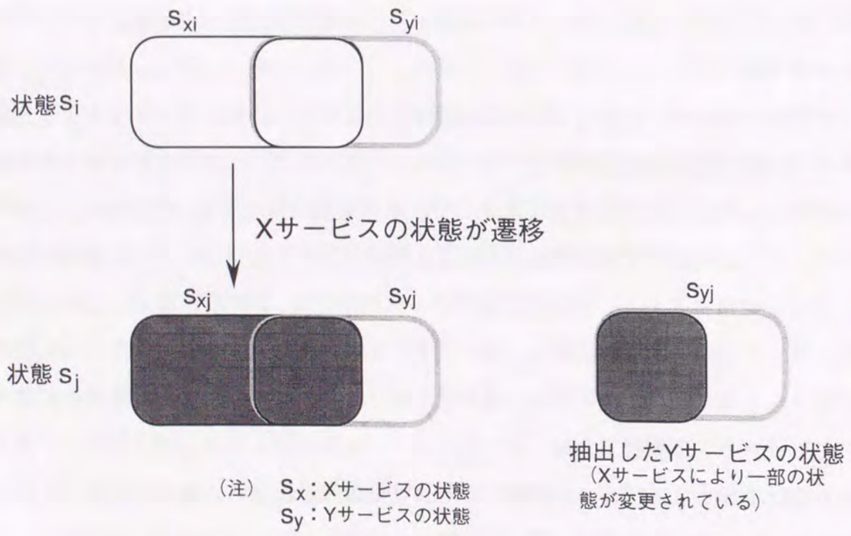


図3.2　合成サービスの状態遷移

定義１：異常な状態への遷移

ＸとＹサービスの仕様RQ-SPEC$_X$とRQ-SPEC$_Y$を以下のように定義する。

$$RQ\text{-}SPEC_X = < T_X, S_X, E_X, \Phi_X, s_{X0}, f_{X0} >$$

$$RQ\text{-}SPEC_Y = < T_Y, S_Y, E_Y, \Phi_Y, s_{Y0}, f_{Y0} >$$

ＸとＹサービスの状態s_{xi}とs_{yi}において、ＸとＹサービスが重なりあった状態$s_i = \{s_{xi} \cup s_{yi}\}$が生成されたとする。$s_{xi}$はｎ個の端末により、$s_{yi}$はｍ個の端末によりそれぞれ定義され、$\{s_{xi} \cap s_{yi}\}$の状態にｋ（$\leqq \min(n, m)$）個の端末が含まれているとする。

この状態s_iにおいて、Ｘサービスの仕様が実行され、Ｘサービスの次状態s_{xj}が生成されたとする。Ｘサービスの状態遷移によって、ＸとＹサービスが重なりあった状態に含まれるｋ個の端末のうちｑ（$\leqq$ｋ）個の端末の状態が変化したとする。変化しなかったｋ－ｑ個の端末の状態をε_iとする。新たに重なり合った状態s_jは以下のようになる。

$$s_{xj} = \Phi_x (s_{xi}, e_{xi})$$

$$s_j = \{s_{xj} \cup \{s_{yi} - \{s_{xi} \cap s_{yi}\}\} \cup \varepsilon_i\}$$

状態s_jにおいて、Ｙサービスに関する端末の状態をs_{yj}とすると、s_{yj}は以下のようになる。

$$s_{yj} = \{\{s_{yi} - \{s_{xi} \cap s_{yi}\}\} \cup \kappa_j\}$$

ここで、κ_jは状態s_jにおいて、ＸとＹサービスが重なりあっている状態に含まれるｋ個の端末の状態を表す。このs_{yj}が、$s_{yj} \subseteq S_Y$を満足しない時、異常な状態への遷移と定義する。

□

次に異常な状態への遷移の具体例を示す。着信転送サービス仕様と着信拒否サービス仕様を同時に提供したときに発生する異常な状態への遷移を示す。以下の説明で用いる着信転送サービスと着信拒否サービスの仕様の一部を表3.1に示す。表3.1の着信拒否サービスと着信転送サービスの各仕様を2.3節の定義１に基づく状態遷移モデルにより記述した例を図3.3と図3.4に示す。

着信転送サービスは、例えば別の相手と通話中で着信を受け付けられない場合に、事前に登録した転送先に着呼を転送するサービスである。図3.4は着信転送サービスの一部を端末状態の遷移により示したものである。本サービスは４端末で構成され、構成する端末

表3.1　着信転送サービスと着信拒否サービスを規定している状態遷移

	着信転送サービスの状態遷移	着信拒否サービスの状態遷移
イベント発生前の状態	t1端末においてダイヤルトーンがなっており、t2端末がt3端末と通話しており、かつt4端末に転送を登録しており、t4端末が空き状態となっている状態	t1端末においてダイヤルトーンがなっており、かつt1端末からt4端末への着信を禁止する機能が起動されており、t4端末が空き状態となっている状態
イベント	t1端末がt2端末に発呼すること	t1端末がt4端末に発呼すること
イベント発生後の状態	t1端末がt4端末を呼び出しており、t2端末がt3端末と通話しており、かつt4端末に転送を登録している状態	t1端末において発信ができなかったことを示す音がなっており、かつt1端末からt4端末への着信を禁止する機能が起動されており、t4端末が空き状態となっている状態

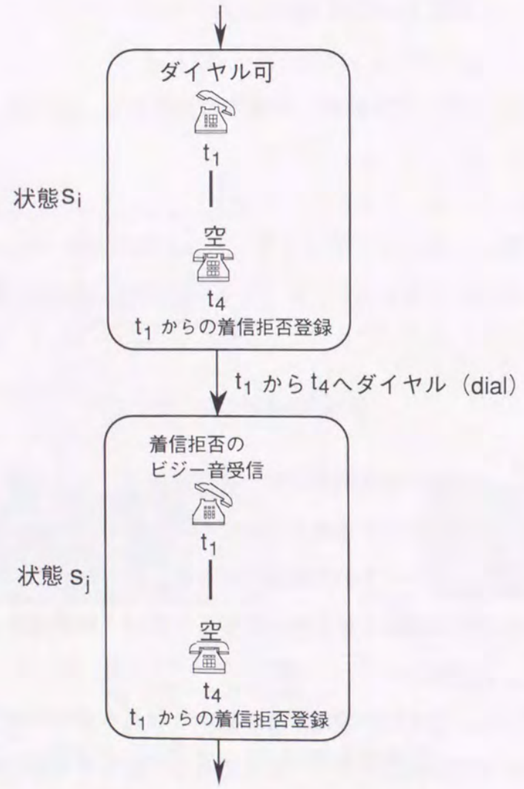


図3.3　着信拒否サービス仕様の状態遷移例

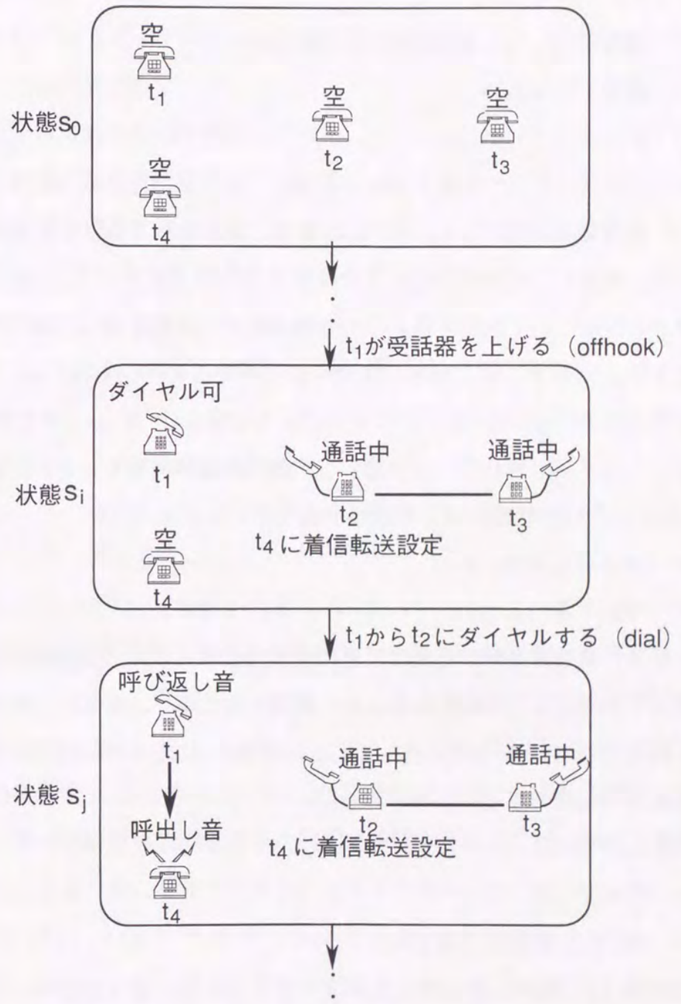


図3.4　着信転送サービス仕様の状態遷移例

の集合T＝ $\{t_1, t_2, t_3, t_4\}$ である。状態 s_i は4端末で構成され、端末t_2とt_3が通話中で、端末t_2からt_4に転送登録され、且つ端末t_1がt_2にダイヤルするために受話器を上げた状態を示している。即ち、$s_i = \{\tau(t_1), \tau(t_2), \tau(t_3), \tau(t_4)\}$ となる。更に、各端末の状態は状態記述要素を用いて、以下のように記述される。

$\tau(t_1)$＝ {ダイヤル可(t_1)}

$\tau(t_2)$＝ {通話中(t_2, t_1),着信転送設定(t_2, t_4)}

$\tau(t_3)$＝ {通話中(t_3, t_2)}

$\tau(t_4)$＝ {空(t_4)}

ダイヤル可(t_1)、空(t_4)が1つの端末の状態を表記する状態記述要素の例である。また、通話中(t_2, t_1) 、着信転送設定(t_2, t_4)等が他の端末と関係のある状態を表記する状態記述要素の例である。関連する端末間は図3.4では線又は矢印で結び示している。

次に、イベントの例について説明する。1つの端末だけの状態を変化されるイベントは、offhook(t1) であり、関連する他の端末の状態も変化させるイベントはdial(t1,t2)である。図3.4に示すようにdial(t1,t2) イベントによる状態 s_i から s_j （呼びが端末t_4に転送され、端末t_1から呼び出されている状態）への遷移が遷移関数Φにより定義される。状態 s_j では、端末t_1とt_4の状態が以下のように変化している。

$\tau(t_1)$＝ {呼び返し音(t_1, t_4)}

$\tau(t_4)$＝ {呼出し音(t_4, t_1)}

着信拒否サービス仕様を2.3節の定義1に基づく状態遷移モデルにより記述した例を図3.3に示す。図3.3の状態 s_i は端末t_4とt_1から構成されており、端末t_4に端末t_1からの着信拒否登録が設定されている状態を示している。状態 s_i は端末の状態を状態記述要素で表し、以下のように記述できる。

$s_i = \{\tau(t_1), \tau(t_4)\}$

$\tau(t_1)$＝ {ダイヤル可(t_1)}

$\tau(t_4)$＝ {空(t_4),着信拒否登録(t_4, t_1)}

また、同図の次状態 s_j は端末t_1が端末t_4にダイヤルしたため、着信拒否サービスが起動され、端末t_1が着信拒否を示すビジー音を受信している状態に変化したことを示している。

$s_j = \{\tau(t_1), \tau(t_4)\}$

$\tau(t_1)$＝ {着信拒否のビジー音(t_1, t_4)}

$\tau(t_4)$＝ {空(t_4),着信拒否登録(t_4, t_1)}

次に、図3.5に、着信転送サービス仕様（Xの添字で表す）と着信拒否サービス仕様（Yの添字で表す）を同時に提供した時に生成される2つのサービスの重なり合った状態とその後の遷移を示す。図3.5の前状態 s_i は、表3.1に記述されている2つの現状態の和集合をとった状態 $\{s_{xi} \cup s_{yi}\}$ を示している。この状態 s_i は以下に示す端末の状態を表している。

・端末t_1においてダイヤルトーンが鳴っている。

・端末t_2とt_3が通話している。

・端末t_2が端末t_4に転送を登録している。

・端末t_1からt_4への着信を禁止している。

・端末t_4が空き状態となっている。

また、状態 s_i を形式的に表現すると以下のようになる。

$s_i = \{\tau(t_1), \tau(t_2), \tau(t_3), \tau(t_4)\}$

$s_{xi} = \{\tau(t_1), \tau(t_2), \tau(t_3), \tau(t_4)\}$

$s_{yi} = \{\tau(t_1), \tau(t_4)\}$

$\tau(t_1)$＝ {ダイヤル可(t_1)}

$\tau(t_2)$＝ {通話中(t_2, t_1),着信転送設定(t_2, t_4)}

$\tau(t_3)$＝ {通話中(t_3, t_2)}

$\tau(t_4)$＝ {空(t_4),着信拒否登録(t_4, t_1)}

s_i を構成する着信転送サービスと着信拒否サービスの端末数はそれぞれn、mとすると、n＝4、m＝2である。状態 $\{s_{xi} \cap s_{yi}\}$ に含まれる端末はt_1とt_4の2端末であり、k＝2となる。

図3.5の次状態 s_j は、表3.1に示した着信転送サービス仕様を規定している状態遷移に従って、その状態遷移の現状態を次状態に遷移させたときに出現する状態を示している。端末t_1とt_4の状態が共に変化しており、q＝2である。

$s_j = \{\tau(t_1), \tau(t_2), \tau(t_3), \tau(t_4)\}$

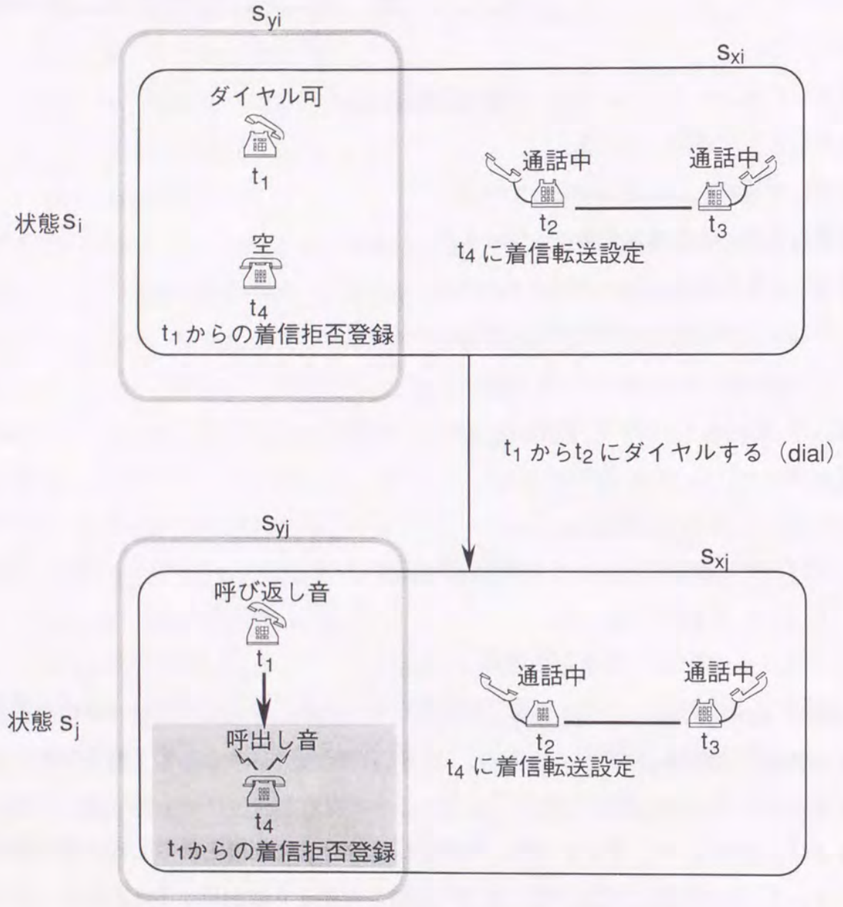
$s_{xj} = \{\tau(t_1), \tau(t_2), \tau(t_3), \tau(t_4)\}$

$s_{yj} = \{\ \{\phi\} \cup \{\tau(t_1), \tau(t_4)\}\ \}$

$= \{\tau(t_1), \tau(t_4)\}$

$\tau(t_1)$＝ {呼び返し音(t_1, t_4)}

$\tau(t_2)$＝ {通話中(t_2, t_1),着信転送設定(t_2, t_4)}



（注）S_{yj} は着信拒否サービスとしては矛盾した状態である。

図3.5　着信転送サービスと着信拒否サービスを合成した場合の異常な状態への遷移の例

$\tau(t_3)=$ {通話中(t_3, t_2)}

$\tau(t_4)=$ {呼出し音(t_4, t_1), 着信拒否登録(t_4, t_1)}

この状態遷移において、図3.5の次状態s_jにおける着信拒否サービスの状態s_{yj}に着目する。この状態s_{yj}は、端末t_4が端末t_1から呼び出されており、かつ端末t_4において端末t_1からの着信を禁止する機能が起動されている状態である。この端末t_4の状態｛呼出し音(t_4, t_1), 着信拒否登録(t_4, t_1)｝は、着信拒否サービス仕様には規定されていない状態（このサービスの本来の意味として禁止されねばならない状態）である。従って、図3.5の前状態s_iに示された状態から次状態s_jに示された状態への遷移は、異常な状態への遷移である。

3.3　仕様記述言語

本節では、通信サービス仕様から異常な状態への遷移を機械的に検出するために要求される仕様記述言語の機能について述べる。

複数の単独サービス仕様を同時に提供したときに発生する異常な状態への遷移を機械的に検出するためには、通信サービス仕様を記述する仕様記述言語は、少なくとも以下に示す2つの機能を有する必要がある。

（1）形式的に記述できること。

（2）単独サービス仕様の機械的な足し合わせ（和集合をとる）ができること。

上記（1）と（2）の機能は、2.3節の定義1で示したサービス仕様の遷移関数Φを2.3節の定義2で示した単一の状態遷移を規定したルールの集合とそれらのルールの適用関数を規定することにより実現できる。このルールは現状態、イベント、次状態から構成する。現状態と次状態は、2.3節の定義1で示した端末の状態を細分化した状態記述要素pの集合として表現される。ルールの集合と適用関数により状態遷移を規定することにより、2つのサービス仕様の足し合わせは、単独サービス仕様を規定しているルールの集合の和集合として実現できる。このようなルールに基づく状態遷移記述を図3.4の着信転送サービスの状態s_iからs_jへの遷移をルールr_1として記述した例を以下に示す。

$r_1 = cs_1 \quad e_1 : \quad ns_1.$

$cs_1=$ {ダイヤル可(t_1), 通話中(t_2, t_1), 着信転送設定(t_2, t_4),

通話中(t_3, t_2), 空(t_4)}

e_1=dial(t_1, t_2)

ns_1= {呼び返し音(t_1, t_4), 通話中(t_2, t_1), 着信転送設定(t_2, t_4),
通話中(t_3, t_2), 呼出し音(t_4, t_1)}

図3.4の例では、サービス状態s_iはr_1の現状態cs_1と等しいので、ルールの適用条件の1つである$s_i \supseteq cs_1$は満足している。また、イベントdial(t_1, t_4)が選択されているため、ルールr_1が適用でき（r_1=M（s_i, dial(t_1, t_2)）、s_iをルールの次状態記述ns_1で書き換え、次状態s_jが生成される（s_j=W（r_1, s_i））。

2.3節の定義2において、Σはルール適用を規定した関数であり、サービスに依存して変わることはない。従ってサービスの仕様はRにより規定される。2.3節の定義2に基づくサービス仕様記述言語としてif-then型のルール形式で記述するＳＴＲ(State Transition Rules)言語[20]を用いて既存電話サービス仕様の記述実験[20][21]を行い、上記（1）と（2）の要求条件への適合性を確認した。

3.4　異常な状態への遷移検出法

本節では、前節で示した2つの機能を有する形式的な仕様記述言語を用いて記述したサービス仕様を対象として、異常な状態への遷移を機械的に検出する手法を示す。最初に、異常な状態への遷移を検出するためのアルゴリズムに用いる4つの関数について定義し、これらの関数を用いたアルゴリズムを示す。

検出のための関数

（1）δT_i=terminal（s_i）：

あるサービスの状態s_iにおいて、このs_iを構成する端末の集合δT_iを求める関数である。

（2）Γ_i=relation（s_i）：

あるサービスの状態s_iにおいて、このs_iを構成する端末のそれらの間の関係の集合Γ_iを求める関数である。Γ_iは関連する2つの端末の組み合わせを1つの要素として構

成される。2つの端末t_iとt_jの組み合わせを示す要素は＜t_i, t_j＞と表す。

（3）s（δT_i）
=terminal-state（s_i, δT_i）：

あるサービスの状態s_iにおいて、このs_iに含まれる端末集合δT_iの端末自体の状態の集合s（δT_i）（他の端末とは関係のない端末の状態）だけを求める関数である。求めた端末の状態は状態記述要素pで表現される。

（4）s（Γ_i）=relation-state（s_i, Γ_i）：

あるサービスの状態s_iにおいて、このs_iに含まれる端末集合δT_iの他の端末と関係している端末の状態s（Γ_i）だけを求める関数である。求めた端末の状態は状態記述要素pで表現される。

検出アルゴリズム

入力：

2つの単独サービス仕様として、Xサービス仕様とYサービス仕様を規定したルールの集合であるR_xとR_yが入力される。

出力：

異常な状態への遷移を検出したXとYサービス仕様を合成した現状態と次状態、及びその遷移に関するルールr_xとr_yを出力する。

ステップ：

（1）合成した状態の生成

R_xとR_yから1つずつルールを選択した後、それらのルールの現状態の和集合を取り、合成サービスの前状態s_iを生成する。

$s_i = cs_{rx} \cup cs_{ry}$

（2）s_iが実際に出現するかどうかの判定

初期状態から、新たな状態が出現しなくなるまで、$R_x \cup R_y$に属するルールを順次適用し、その過程において出現した状態の中に状態s_iが存在する場合、状態s_iを実際に出現する状態であると判定する（文献２２、２３参照）。

（3）ルールの実行後の合成サービスの状態の導出

状態s_iにおいて、r_xを適用し、遷移後の状態s_jを導出する。

s_j=W（r_x, s_i）

（4）単独サービスの状態の抽出

状態s_jにおけるYサービスを構成する状態s_{yj}を抽出する。抽出式を以下に示す。

s_{yj}

$=$terminal-state（s_j, terminal（s_{yi}））

$\cup$relation-state（s_j, relation（s_{yi}））

（5）異常な状態への遷移の判定

ステップ（4）で抽出された状態s_{yj}が、Yサービスの状態として出現するかを判定する。ステップ（2）の手法を適用してYサービス仕様を規定したルール集合R_yを初期状態から順次適用し、全てのルールを適用し終わっても状態s_{yj}が生成されない時に異常な状態への遷移であると判定する。この結果、合成サービス状態s_i, s_j及び各サービスのルールr_xとr_yを出力する。

ステップ（1）からステップ（5）の処理を、XとYサービスの全てのルールの組み合わせに対して行なうことにより、合成した状態においてXサービス仕様を規定したルールを実行するときに発生する異常な状態への遷移を検出する。同様の手法を用いて、合成した状態においてYサービス仕様を規定したルールを実行するときに発生する異常な状態への遷移を検出する。

3.5　異常な状態への遷移の検出例

3.3節で説明したif-then型のルール形式に基づく形式的仕様記述言語として、ＳＴＲ言語[20]を採用し、その検出例を説明する。表3.2に、表3.1に示された2つの状態遷移をＳＴＲ言語を用いて記述した例を示す。本章では、このＳＴＲ言語を用いて記述したサービス仕様を例に取り、図3.5で示した異常な状態遷移の検出例について以下に示す。尚、以下の説明では、図3.5の状態sの添字はi=1, j=2 とする。

入力：

着信転送サービス仕様と着信拒否サービス仕様の一部を規定しているルールとして、表3.2に示した規則（r_{tcs}とr_{cfv}）を用いる。

表3.2　STR言語によるサービス記述例

	着信転送サービス	着信拒否サービス
規則	dial-tone(t1), path(t2, t3), path(t3, t2), m-cfv(t2, t4), idle(t4) dial(t1, t2) : ringback(t1, t4), ringing(t4, t1), path(t2, t3), path(t3, t2), m-cfv(t2, t4).	dial-tone(t1), m-tcs(t4, t1), idle(t4) dial(t1, t4) : busy-dial(t1), m-tcs(t4, t1), idle(t4).
意味	t1端末においてダイヤルトーンがなっており(dial-tone(t1))、t2端末がt3端末と通話しており(path(t2, t3), path(t3, t2))、かつt4端末に転送を登録しており(m-cfv(t2, t4))、t4端末が空き状態 (idle(t4))となっている状態において、t1端末がt2端末に発呼すると(dial(t1, t2))、t1端末がt4端末を呼び出している状態(ringback(t1, t4), ringing(t4, t1))に遷移する。	t1端末においてダイヤルトーンがなっており(dial-tone(t1))、かつt1端末からt4端末への着信を禁止する機能が起動されており(m-tcs(t4, t1)、t4端末が空き状態となっている(idle(t4))状態において、t1端末がt4端末に発呼すると(dial(t1, t4))、t1端末において発信ができなかったことを示す音がなる状態(busy-dial(t1))に遷移する。

ステップ：

（1）合成した状態の生成

r_{tcs}とr_{cfv}の現状態記述から図3.5で示す状態s_1を生成する。

$s_1 = cs_{rtcs} \cup cs_{rcfv}$

$= \{dial\text{-}tone(t_1), idle(t_4), m\text{-}tcs(t_4, t_1)\}$

$\cup \{dial\text{-}tone(t_1), path(t_2, t_3), path(t_3, t_2), m\text{-}cfv(t_2, t_4), idle(t_4)\}$

$= \{dial\text{-}tone(t_1), path(t_2, t_3), m\text{-}cfv(t_2, t_4),$

$path(t_3, t_2), idle(t_4), m\text{-}tcs(t_4, t_1)\}$

（2）s_1が実際に出現するかどうかの判定

この判定法（文献22、23）に関しては既に提案されているため、ここではその説明を省略する。

（3）ルールの実行後の合成サービスの状態の導出

状態s_1においてr_{cfv}を実行する。これは、r_{cfv}の現状態に記述されている全ての状態記述要素を次状態に記述されている全ての状態記述要素に置き換えることによって実現することができる。以下に、状態s_1の遷移後の状態s_2を以下に示す。

$s_2 = \{ringback(t_1, t_4), path(t_2, t_3),$

$m\text{-}cfv(t_2, t_4), path(t_3, t_2), ringing(t_4, t_1), m\text{-}tcs(t_4, t_1)\}$

（4）着信拒否サービスの状態の抽出

状態s_2における着信拒否サービスの状態s_{tcs2}を抽出する。以下に抽出式を示す。

$s_{tcs2} = terminal\text{-}state(s_2, terminal(s_{tcs1}))$

$\cup relation\text{-}state(s_2, relation(s_{tcs1}))$

$= terminal\text{-}state(s_2, terminal(\{dial\text{-}tone(t_1), idle(t_4), m\text{-}tcs(t_4, t_1)\}))$
$\cup relation\text{-}state(s_2, relation(\{dial\text{-}tone(t_1), idle(t_4),$
$m\text{-}tcs(t_4, t_1)\}))$

$= terminal\text{-}state(s_2, \{t_1, t_4\})$

$\cup relation\text{-}state(s_2, \{<t_1, t_4>\})$

$= \{\phi\} \cup \{ringback(t_1, t_4), ringing(t_4, t_1), m\text{-}tcs(t_4, t_1)\}$

$= \{ringback(t_1, t_4), ringing(t_4, t_1), m\text{-}tcs(t_4, t_1)\}$

（5）異常な状態遷移の判定

着信拒否サービス仕様を規定したルールにより生成される状態にはステップ（4）において抽出された状態s_{tcs2}（着信拒否サービスでは禁止している状態）は出現しない。従って、状態s_{tcs2}は着信拒否サービス仕様には規定されていない状態であると判断し、図3.5の状態s_1から状態s_2への遷移は異常な状態への遷移である。

出力：

異常な状態への遷移を検出した着信拒否サービスと着信転送サービスを合成した現状態s_1と次状態s_2、及びその遷移に関するルールr_{tcs}とr_{cfv}が出力される。

3.6 評価及び考察

着信転送（CFV）、着信拒否（TCS）、発信拒否（OCS）の3つのサービス仕様をSTR言語を用いて記述し、提案した異常な状態への遷移検出法の有効性を机上実験により評価した。尚、検出アルゴリズムの正当性は、定義1より明らかである。

3.6.1 検出実験

実験で用いたSTR言語で記述したCFVとTCSの基本仕様の例を付録3.1に示す。これらの個々のサービスの記述は端末の状態遷移の様子をアニメーション的に表示できるSTR言語のインタプリタ[21]を用いて確認している。実験では、CFVとTCS及びCFVとOCSの2つの組み合わせを4端末で行った。それぞれのルールの和集合を構成して、インタプリタにより逐次実行させながら提案したアルゴリズムに従って異常な状態への遷移の机上検出実験を行った。検出結果を表3.3に示す。表3.3の第1行の略号の意味は、以下の通りである。

T：ルール形式で記述したサービス仕様として規定されている実行可能な状態遷移の数。

T_{ab}：検出された異常な状態への遷移の数。

T_d：検出されるべきサービス競合の数。

T_{ab}／T：絞り込み率。

表3.3　サービス仕様競合検出の実験結果

サービスの組み合わせ	T	T_{ab}	T_d	T_{ab} / T
CFV+TCS	112	7	1	0.06
CFV+OCS	168	23	1	0.13

3.6.2　検出精度

ルール形式で記述したサービス仕様を合成した場合に提案手法を用いて検出できる異常な状態への遷移の検出精度は、その有効性を評価するうえで重要となる。何故ならば、もし全て検出されなければ、残りの検出されなかった状態遷移を見つけ出すために、合成した仕様に含まれる実行可能な全ての状態遷移を設計者が再度全て確認する必要があり、機械的に検出した効果が無くなるからである。記述実験した3つのサービスのCFVとTCS、CFVとOCSをそれぞれ組み合わせた場合の検出実験を行った。この結果、検出された異常な状態への遷移の中に、検出されるべきサービス競合は全て含まれていたため、異常な状態への遷移の検出により、状態遷移の意味的矛盾に起因するサービス競合の検出支援として有効であることを確認できた。

3.6.3　人手による検出削減効果

ルール形式で記述した2つのサービス仕様を組み合わせて、それぞれのサービス動作を満足する合成サービス仕様の設計において、サービス仕様を組み合わせることによって生成される状態が、合成サービス仕様として含むべき状態であるか否かを設計者が判断しながら、設計を進める必要がある。提案手法により、このような設計者が検討しなければならない状態を機械的に検出できるため、設計者が2つのサービス仕様を組み合わせた場合に全て実行可能な状態遷移を検討する必要が無くなり、検出された状態だけに絞り込めるため、設計の効率化が図れる。

CFVとTCS、CFVとOCSをそれぞれ組み合わせた場合の検出実験の結果（表3.3）から、絞り込み率はおよそ0.1となり、本手法により設計者が確認しなければならない状態が10分の1に削減されたことを示している。

3.6.4　実現上の課題

提案手法を実現する場合の検出精度に関する課題として、3.4.1節で述べた検出アルゴリズムのステップ（2）における合成サービスの状態s_iが実際に存在することを判定する手法の実現が重要となる。本判定法は、ルールの集合として記述したサービス仕様にお

いて、全ての状態への可到達性の判定であり、この可到達性判定法が提案検出アルゴリズムのカバー率に影響する。ルールで記述したサービス仕様では、端末数に比例してサービスの状態が増大するため、端末数に依存して全てのサービス状態が生成されるか、されないかが変わってくる。例えば、端末数ｎの場合に生成される状態には、サービス状態s_iは存在しないが、ｎ＋１の場合にはs_iが存在する場合が考えられる。これは、端末数をｎとして検出アルゴリズムを実行した場合には異常な状態は検出されないが、端末数をｎ＋１として検出アルゴリズムを実行した場合には、異常な状態が検出される可能性があることを示している。検出精度を100%とするためには、端末数を無限とする必要があり、実現が困難となる。しかし、ルールで記述したサービス仕様における可到達状態の数は、端末数の増加に対してある端末数以上では増加しなくなることが証明されている[23]。従って、検出される異常な状態が飽和する端末数が存在し、その端末数を与えることにより全ての異常な状態が検出できる。実際の通信サービスの検証支援システムでは、端末数の任意な設定を可能とし、端末数をパラメータとして振らして繰り返し実行することにより実現できる。但し、文献２３では、可到達状態集合の大きさが飽和する端末数は3^{k-1}と3^kの間（ｋは任意）の３倍の区間に存在することを証明している。しかし、端末数を任意に設定することにより実現する通信サービス検証システムにおいては、パラメータとして振らせる端末数として、どのようなｋの区間の値をとるべきかと、ｋの値が大きくなるとその区間の全ての端末数による繰り返し実行数が多くなり、検証効率の低下が課題となる。端末数をパラメータとして振らせて繰り返し実行により実現するための具体的な端末数を特定できること、又は絞り込めることができることを証明する必要がある。このような端末数の存在に関する定量的な証明は、今後の課題である。

3.6.5　ルール集合の管理検証に関する考察

本稿で提案した手法は既存のルール集合に新しいルール集合を追加した場合のルール集合間の矛盾に関する問題である。このような問題は知識情報処理の分野でも研究されており、非専門家に対する利点と矛盾検出法の観点から比較・考察する。

（１）非専門家に対する利点

既存ルール集合に新しいルール集合を追加した場合のルール集合間の矛盾回避、無矛盾保持等の方式として、知識情報処理ではＴＭＳ、ＡＴＭＳが研究されている[24][25]。これらの方式では、ルール集合間の矛盾の機械的な検出と回避が可能であるため、非専門家に対しては、以下のような利点がある。

ａ）非専門家自身が人手でルール集合間の矛盾を検出しなくてもよい。尚、もし検出しようとした場合には、既存の全てのルールとそれらのルール集合から推論される結論を非専門家が理解する必要がある。

ｂ）また、非専門家が解消方法を考えなくても、システムより提示が得られる。

本稿で提案した手法は、上記ａ）と同じ利点を狙っている。即ち、非専門家が既存のルール集合により定義されたサービス仕様、具体的には単一の状態遷移を定義した個々のルールの意味とそれらのルールを適用することによって生成される全ての状態遷移を知らなくても異常な状態への遷移として矛盾が機械的に検出できる。また、上記ｂ）については、検出した異常な状態への遷移を解消できる仕様がルールとして定義できることは確認している[26]。しかし、非専門家に対しては、解消ルールの機械的な獲得支援が必要と考えており、具体的な手法については、今後の課題である。

（２）矛盾検出法

ルール集合間の制約チェックとしてデータベース等で行われている矛盾検出法[25]がある。この手法は、制約とデータを命題論理や述語論理の論理式の集合として捉え、論理演算を行うことにより論理的な矛盾として検出する方法である。この手法では、状態の論理的矛盾、例えば論理演算の結果、状態Ａとその否定状態￢Ａが導き出されたならば、その論理式の集合の矛盾は検出できるが、状態の意味的な矛盾は検出できない。本稿で提案する手法は、if-thenのルール集合で規定された状態遷移の意味的な矛盾の検出を狙っている。即ち、意味的な矛盾を異常な状態への遷移として形式的に定義し、その定義に基づく矛盾の検出アルゴリズムを提案している。

3.7　結言

通信サービスを迅速に開発するための、重要な課題であるサービス競合検出を支援するための方法について提案した。具体的には、状態遷移モデルに基づいて記述された通信サービス仕様を対象として、２つの単独サービス仕様を合成した時に発生する意味的矛盾の機械的な検出法について示した。意味的矛盾として、「異常な状態への遷移」の定義を

示した。また、異常な状態への遷移を機械的に検出するために必要となる仕様記述言語への要求条件を示した。その記述言語に基づき記述された仕様を合成した場合の異常な状態への遷移を検出するアルゴリズムについて提案した。更に、実際のサービス仕様をＳＴＲ言語により記述して実験を行い、サービス競合として設計者が検証しなければならない状態遷移の数が約１割に絞り込むことができることを示した。

第４章
通信サービスにおけるメッセージシーケンス図記述からルール記述への仕様変換法

4.1　緒言

ソフトウェア開発の上流工程において、要求仕様の誤りを早期に検出・解消することにより、下流工程での試験が効率的に実施でき、手戻りの少ない開発により生産性の向上が期待できる。このような目的から、ソフトウェアの要求仕様を形式的な仕様記述法を用いて定義することにより仕様検証の機械的な支援を目指す技術が期待されている。これに伴い電話サービスを実現する通信サービスソフトウェアにおいてもメッセージシーケンス図（ＭＳＣ）、ＳＤＬで代表されるような手続き型の仕様記述法[(1)~(3)]とif-then 型のルール集合による宣言型の仕様記述法[(4)]が提案されている。しかし、それぞれの仕様記述法は検証能力と機械的支援のし易さの点で差別化されている。

本章では、ＭＳＣの手続き型の仕様記述法の長所を損なうことなく仕様検証能力を向上させることを目的として、ＭＳＣで記述された仕様のif-then 型のルール集合による仕様への変換法を提案する[(5)~(6)]。具体的には、まず、端末と通信網間のメッセージに基づくＭＳＣ記述要素と端末状態とイベントの関係を明確化し、変換アルゴリズムを提案する。尚、本論文で提案する方法の具体的な実現は容易に可能である。そして、提案した方法に基づいて、具体的な適用例として、文献１と２においてＩＮのサービス仕様入力として提案されているＭＳＤＬ（Message Sequence Description Language)による仕様記述から文献４で提案されているＳＴＲ言語への変換を示し、その適用例を用いて考察を行う。

4.2　ＭＳＣによる仕様記述

変換先の記述法であるルール集合による仕様記述法は2.3節に示すとおりであり、本節では、変換元サービス仕様としてＭＳＣによる仕様を定義する。

定義１：ＭＳＣによるサービス仕様定義

ＭＳＣによるサービス仕様ＭＳＰＣを以下のように定義する。

ＭＳＰＣ＝＜Ｎ，P_{msc}，Ｃ，Σ＞

ここで、

（１）ＮはＭＳＣに記述されているノードの集合を示す。Ｎの要素としては、端末の集合と１つの通信網を示すノードより構成される。

（２）P_{msc} はＭＳＣに記述されているＭＳＣ記述要素ｍの集合を示す。ＭＳＣ記述要素ｍとして、分岐のための条件値（ｖ）を含むものと含まないものの以下の２種類の表記法を定める。

m_i ＝記述要素名（始点，終点，ﾊﾟﾗﾒｰﾀ1,・・・・,ﾊﾟﾗﾒｰﾀk ）

m_j ＝記述要素名（始点，終点，ﾊﾟﾗﾒｰﾀ1,・・・・,ﾊﾟﾗﾒｰﾀh ：ｖ）

始点、終点には、Ｎの要素である端末識別子と通信網を示す識別子が記述され、その組み合わせは表4.１の通りである。始点と終点に端末識別子を含むＭＳＣ記述要素を特にメッセージ記述要素と呼ぶ。また、始点に通信網を終点に“NULL”を持つＭＳＣ記述要素を特に網内処理記述要素と呼ぶ。パラメータは、記述要素定義において詳細情報の設定が必要な場合のみ設定される。

（３）Ｃは表4.２に示す制御文の集合である。

（４）ΣはＭＳＣ記述要素と制御文の手続き（時系列の流れ）を規定している。

□

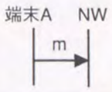
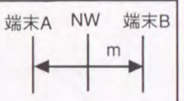
以上で定義したＭＳＣによる仕様記述の例を図4.１に示す。図4.１の例では、端末ＡとＢの２つの端末と通信網間のメッセージシーケンスが記述されている。図4.１（ａ）はテキスト形式での記述であり、図4.１（ｂ）はその図形式による表現を示す。

4.3　ＭＳＣからルールへの変換法

4.3.1　前提条件

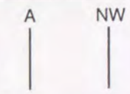
本論文で提案するＭＳＣからルールへの変換では、メッセージと端末状態及びイベントの間には、以下の関係が存在することを前提条件とする。

表4.1　MSC 記述要素における始点と終点の組み合わせ

種別	始点	終点	意味	図形式例
メッセージ記述要素	端末識別子	NW	端末から通信網へ送信される信号を示す	
	NW	端末識別子	通信網から端末へ送信される信号を示す	
	端末識別子	端末識別子	端末間の状態を示し、通話中と保留中の時に設定される	
網内処理記述要素	NW	NULL	網内処理を示す	

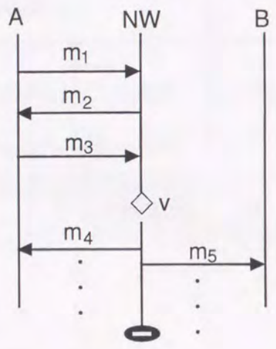
（注）NULL ：特別情報終点を示し、NULL の文字が設定される
NW　：通信網を示す。

表4.2　MSC 記述要素のための制御文

制御文	意味	テキスト例	図形式例
if 文	記述要素の戻り値(v) を評価する	.if (v= VALUE)	
or 文	端末動作による分岐を示す。ある時で、端末の動作で起こりうる記述要素列を示す	記述要素列1 | 記述要素列2	
end 文	制御の終了を示す	.end(NW, process)	
path 文	端末と通信網の接続構成を定義する	.path (A, NW)	

```
process (xxxxx)
{
.path (A, NW);
.path (NW, B);
   m1 (A, NW) ;
   m2 (NW, A) ;
   m3 (A, NW :v) ;
.if (v=v1(B) ) {
   m4 (NW, A);
   m5 (NW, B);
      .
      .
.stop(NW, xxxxx);
```

(a)テキスト形式



(b) 図形式

図4.1　MSC による仕様記述例

（1）ＭＳＣで記述した仕様とルール集合に変換した仕様は意味的に同じとする。従って、ＭＳＣ記述要素とその対応する端末状態を表す状態記述要素及びイベントは同じ意味を持つものとする。

（2）ＭＳＣ記述では、端末の初期状態は明示的に記述されていないため、変換アルゴリズムにおいて状態遷移として矛盾しない状態を設定することとする。

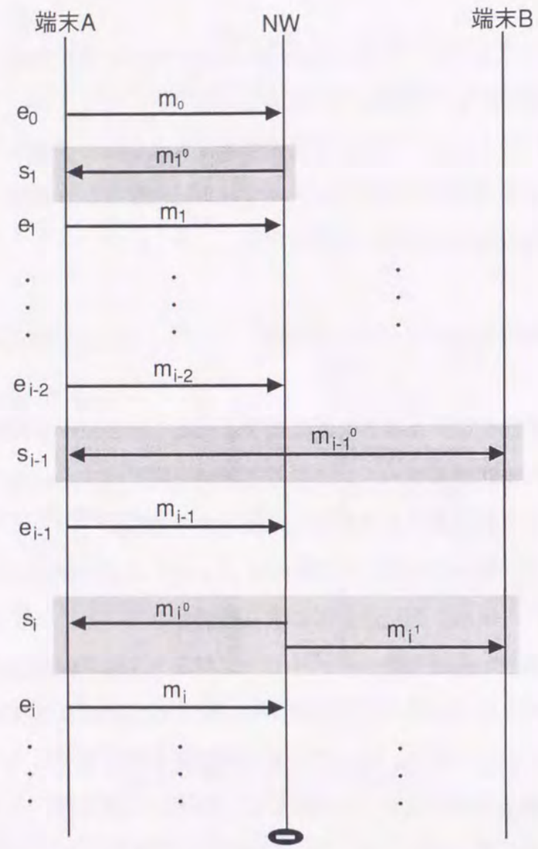
（3）端末から通信網へのメッセージ記述要素を端末に入力されたイベントと対応させる。

（4）端末の最終状態は通信網から端末へのメッセージ記述要素から変換するのではなく、ＭＳＣ記述におけるstop文を解釈して変換する。

4.3.2　ＭＳＣ記述と端末状態とイベントの対応

ＭＳＣで記述された仕様をルールの集合に変換するためには、ＭＳＣ記述要素とルール記述の構成要素である端末状態及びイベントの対応関係を明確化する必要がある。ＭＳＣで記述された仕様は、4.2節の表4.1で説明した端末から通信網へのメッセージ、通信網から端末へのメッセージ、端末間の通信状態を示すメッセージからなるメッセージ記述要素と網内の処理を記述した網内処理記述要素により構成されている。4.3.1節で示した前提条件（3）により、端末から通信網へのメッセージがルール記述におけるイベントに対応する。端末から通信網へのメッセージ記述要素は、ＭＳＣで記述された仕様において繰り返して複数記述されている。また、ルールは1つの状態遷移を規定していることから、ルール記述における状態は、このイベントに対応する端末から通信網へのメッセージ記述要素と次ぎの端末から通信網へのメッセージ記述要素間に含まれるＭＳＣ記述要素が端末の状態と対応する。しかし、ＭＳＣ記述仕様の最後はstop文として記述され、そのstop文以降の端末から通信網へのメッセージは記述されないため、この場合は、端末から通信網へのメッセージからＭＳＣ記述におけるstop文までが状態に対応する。

図4.2に以上で説明したＭＳＣ記述とイベントと端末の状態の関係を示す。端末から通信網へのメッセージ$m_0, m_1, \cdots, m_{i-2}, m_{i-1}, m_i$がイベント$e_0, e_1, \cdots, e_{i-2}, e_{i-1}, e_i$にそれぞれ対応している。また端末の状態$s_1, \cdots, s_{i-1}, s_i$はイベントに対応するメッセージに挟まれるＭＳＣ記述要素と対応する。



（注）e_i：イベントに対応するMSC記述部分
s_i：端末状態に対応するMSC記述部分

図4.2　MSC 記述と端末状態およびイベントの関係

4. 3. 3　状態とイベントへの変換

（1）状態への変換

端末の状態に変換するためのＭＳＣ記述要素（特に状態として変換されるＭＳＣ記述要素を以下では状態変換要素と呼ぶ）の種別は複数あり、その一覧を表4. 3に示す。1つの状態変換要素がルール記述において用いている状態記述要素に変換される。また、イベントに変換されるメッセージm_{i-1}とm_iに挟まれるＭＳＣ記述要素の中に複数の状態変換要素が含まれる場合（これらをＭＳＣ記述要素列Δm_iと呼ぶ。例を付録4. 1に示す）は、端末の状態は変換された状態記述要素の和集合として得られる。状態変換要素には、メッセージ記述要素から構成されるものと網内処理記述要素から構成される2種類がある。表4. 3は、図4. 2においてｉ－1番目の端末から通信網へのメッセージの次の状態s_iと直前の状態s_{i-1}に反映される状態変換要素種別とそれぞれのルールの状態記述要素への変換を示したものである。

表4. 3の項番1から6は、メッセージ記述要素の状態変換要素種別とそれぞれの状態記述要素への変換を示したものである。項番1のメッセージ記述要素列は、端末xから通信網へのメッセージ$m_{i-1}(x, NW)$に対して通信網が応答した状況を示しており、$m_i{}^0(NW, x)$が状態記述要素$p_{mi}{}^0(x)$に変換できる。

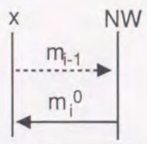
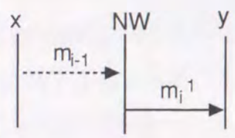
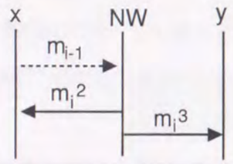
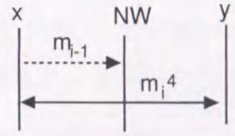
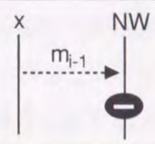
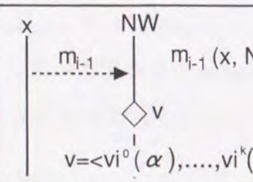
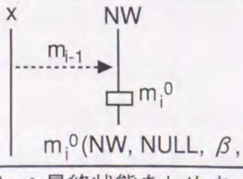
項番2は端末xからのメッセージを通信網が受信して、端末yに通信網が指示している状況を示している。$m_i{}^1(NW, y)$が状態記述要素に変換されるのであるが、そのパラメータである端末識別子には、端末xからの要求により端末yに指示が伝達されたと解釈でき、端末相互間で関連があることを示すためにxとyを設定し、$p_{mi}{}^1(x, y)$とする。しかし、$m_{i-1}(x, NW)$に続く端末xから通信網へのメッセージが以降stop文まで記述されていない場合は、端末の終了要求と解釈し、端末xは最終状態f_0に、端末yは$m_i{}^1(NW, y)$に対応する状態記述要素である$p_{f0}(x)$, $p_{mi}{}^1(y)$に変換する。

項番3は、端末xから通信網へのメッセージに対して、端末xとyに指示が伝達されると解釈し、双方の端末に関連がある状態記述要素$p_{mi}{}^2(x, y)$, $p_{mi}{}^3(y, x)$に変換する。

項番4は端末xから通信網へのメッセージに対して、端末xとyが相互に安定した状態に遷移したことを示し、$m_i{}^4(x, y)$を端末xとyが相互に関連するそれぞれの状態記述要素$p_{mi}{}^4(x, y)$, $p_{mi}{}^4(y, x)$に変換する。

項番5は端末xから通信網へ終了を示すメッセージ$m_{i-1}(x, NW)$が記述された場合であ

表4.3　状態変換要素

項番	変換要素種別	状態記述要素 p_i への変換	状態 s_i , s_{i-1} への反映
1		$P_{m_i^0}(x)$	$s_i \leftarrow \{s_i \cup p_i\}$
2		$P_{m_i^1}(x, y)$ 端末x からのm_{i-1}に続くメッセージがない場合は以下とする $P_{m_i^1}(y)$, $P_{f0}(x)$	
3		$P_{m_i^2}(x, y)$, $P_{m_i^3}$ (y, x)	
4		$P_{m_i^4}(x, y)$, $P_{m_i^4}$ (y, x)	
5		$p_{f0}(x)$	
6		$P_{v_i^0}(\alpha)$,...., $P_{v_i^k}(\alpha)$	$s_{i-1} \leftarrow \{s_{i-1} \cup p_i\}$
7		$P_{m_i^0}(\beta, \gamma\beta)$	$s_i \leftarrow \{s_i \cup p_i\}$ $s_{i-1} \leftarrow \{s_{i-1} \cup p_i\}$

(注) 1) P_{f0} ：最終状態をしめす。
2) α：項番6の 分岐値に関連する端末識別子が入り、MSC において記述される。
3) β：項番7の m_i^0 に関連する端末の端末識別子が入り、MSC において記述される。
4) γ β：βに設定された端末と関係をもつ端末識別子が入る。
5) U：和集合を構成することを示す。

り、ＭＳＣによる仕様の記述の終了を示すstop文があり、端末xの状態は最終状態 f_0 に遷移することを示す状態記述要素 $p_{f0}(x)$ に変換する。

項番６は端末から通信網へのメッセージ $m_{i-1}(x, NW: v)$ の中に分岐条件 $v = < v_i^0(\alpha), \cdots, v_i^k(\alpha) >$ が記述された場合であり、これらの分岐条件値は端末から通信網へのメッセージ m_{i-1} に対するイベントが入力できるための条件を示す状態であると解釈でき、分岐条件値をそれぞれに対応した状態記述要素である $p_{vi}^0(\alpha), \cdots, p_{vi}^k(\alpha)$ （ここで、αはＭＳＣの仕様記述において記述された端末識別子を示す）に変換し、１つ前の状態 s_{i-1} の状態として追加する。

項番７は、網内処理記述要素の状態変換要素種別と状態記述要素への変換を示す。網内処理記述要素 $m_i^0(NW, NULL, \beta, \gamma\beta)$ は、端末から通信網へのメッセージ $m_{i-1}(x, NW)$ に対するイベントが入力できるための条件を示す状態及び遷移後も維持されるサービスの起動条件等を定義するための状態であると解釈できる。従って、変換された状態記述要素 $p_{mi}^0(\beta, \gamma\beta)$ （ここで、βはＭＳＣの仕様記述において記述された端末識別子を示し、γβはβと関連ある端末識別子を示す）は、状態 s_i と１つ前の状態 s_{i-1} の状態として追加される。

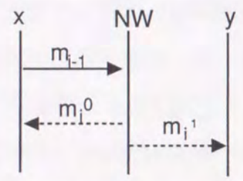
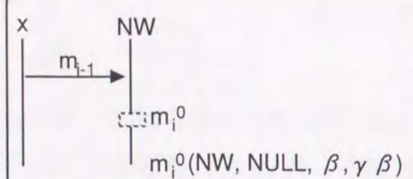
（２）イベントへの変換

次に、イベント変換要素種別について表4.4に示す。表4.4の項番１は、端末xから通信網へのメッセージ $m_{i-1}(x, NW)$ を受けて通信網は、端末xとyに指示するメッセージ $m_i^0(NW, x)$ と $m_i^1(NW, y)$ が記述されている場合を示している。m_{i-1} を変換して得られるイベントは端末xとyを関係付けるためのイベントであると解釈できるため２つの端末識別子をパラメータとして持った $e_{i-1}(x, y)$ に変換する。

項番２は、端末xから通信網へのメッセージ $m_{i-1}(x, NW)$ に対して網内処理記述要素 $m_i^0(NW, NULL, \beta, \gamma\beta)$ が記述されている場合を示している。m_i^0 を変換して得られるイベントの第１パラメータはメッセージの送出元を示す端末xが設定できる。また、関連する端末を示す第２パラメータはβがxの場合はγβとして設定されている端末xと関連があることを記述した端末が入り、γβがxの場合はβとして設定されている端末xと関連付けられている端末が入る。

項番３は、項番１と２以外の状況を示しており、イベントの持つパラメータは、端末xから通信網へのメッセージの送出元（始点に設定される）の識別子だけを持つ $e_{i-1}(x)$ と

表4.4　イベント変換要素

項番	変換要素種別	イベントe_{i-1}
1		e_{i-1} (x, y)：
2		e_{i-1} (x, β又はγ β)：
3	上記以外（表4.3参照）	e_{i-1} (x)：

（注）β，γ βは表4.3の（注）3），4）参照。

して変換される。

4.3.4　変換アルゴリズム

（1）前提条件

ＭＳＣで記述されたサービス仕様は静的検証[7]され、論理的な矛盾は解消されているものとする。

（2）変換アルゴリズム

（入力）ＭＳＣによる記述仕様

（出力）ルールの集合

ステップ１：仕様構成端末の抽出

ＭＳＣで記述された仕様からpath文を解釈して端末数を抽出し、それぞれに端末識別子を付与する。

ステップ２：変換すべきＭＳＣ記述要素列の抽出

端末から通信網へのメッセージm_{i-1}とm_i、又はm_iが存在しない場合は、m_{i-1}とstop文を検出し、それぞれのメッセージ記述要素の間に含まれるＭＳＣ記述要素列Δm_iを抽出する。

ステップ３：状態s_iへの変換

ＭＳＣ記述要素列Δm_iから表３に示す状態変換要素を抽出し、状態記述要素に変換する。Δm_iに含まれる全ての状態変換要素から変換した状態記述要素の和集合をとり状態s_iを生成する。

ステップ４：状態s_{i-1}への追加

ＭＳＣ記述要素列Δm_iに、表4.3の項番６と７に示す分岐を含むメッセージ記述要素と網内処理記述要素に対する状態変換要素が含まれる場合は、表4.3のそれぞれ変換表に従い状態記述要素に変換して、状態s_iの直前の状態s_{i-1}に追加する。また、状態s_iに含まれる端末の状態が状態s_{i-1}の中に既に定義されているかを判定し、定義されていない場合は、その端末の初期状態記述要素p_{s_0}を追加する。

ステップ５：イベントへの変換

端末から通信網へのメッセージm_{i-1}を表4.4のイベント変換要素に対応する変換に従いイベントe_{i-1}に変換する。

ところで、分岐を含んだ端末xから通信網へのメッセージの場合（付録4.2参照）、同じメッセージで分岐毎に複数種別のイベント変換要素が対応する場合がある。この場合は、端末から通信網への同じメッセージでイベントのパラメータとして2つの端末識別子を持つ場合と1つの端末識別子を持つ場合に変換されるが、このような場合は2つの端末識別子を持つ$e_{i-1}(x,y)$だけをイベントとして変換する。

ステップ6：ルールの構成

ステップ3～5で変換した状態s_{i-1}, s_i及びイベントe_{i-1}から以下のルールを構成する。

s_{i-1}　e_{i-1}:　s_i.

但し、状態s_{i-1}を構成している全ての状態記述要素に含まれる全ての端末識別子が、イベントの端末識別子から辿れない端末識別子を含む状態記述要素は（付録4.3参照）、端末識別子の逐次代入に基づくルールマッチングの正当性を保証できないため現状態から削除してルールを構成する。また、既に同じルールが生成されている場合は、ルール集合として蓄積しない。尚、同じルールとは、端末識別子を相対的に入れ替えた場合に、シンタックスが同じになるルールを言う。

ステップ7：ＭＳＣ記述仕様の最後の判定

a）ステップ2でstop文を解釈したかを判定する。もし、解釈していれば、次のｂ）に進む。また、解釈していなければ、ステップ2に戻る。

b）全てのif文とor文を通したかを判定する。もし、全て通っていれば、処理を終了する。通っていない場合は、次のif文とor文に分岐してステップ2に戻る。

以上で示したアルゴリズムの処理フローを図4.3に示す。

4.4　適用例と考察

本章では、前章で述べたＭＳＣ記述仕様をルール集合に変換する方法を用いて具体的な変換の適用例を示した後、本論文で提案した変換法の評価を行い、またその応用として仕様確認、仕様検証の観点から変換法の有用性を示す。

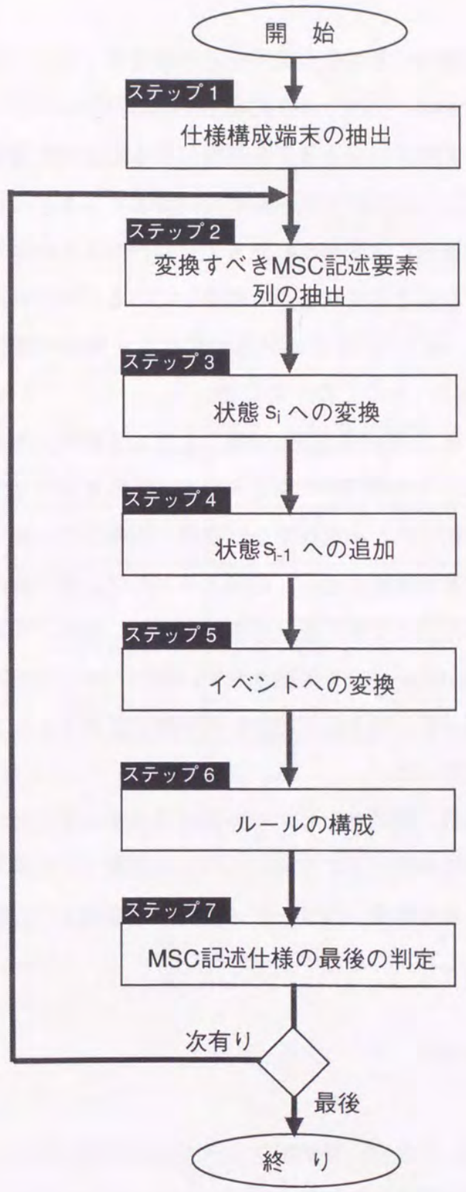


図4.3　変換処理フロー

4.4.1　具体的事例への適用と変換法の正当性

通信サービス仕様として、2者間の通話を実現する基本電話サービス（POTS）仕様への適用例を示す。ＭＳＣの仕様記述は、文献1と2に基づくＭＳＤＬのステップ1記述法とし、ルールによる仕様記述は、文献4によるＳＴＲ言語による記述法を適用する。

ＭＳＤＬのステップ1記述によるPOTSの仕様記述例を付録4.5（1）に示す。また、付録4.5（1）の仕様に変換法を適用して変換されたＳＴＲ言語による仕様記述例を付録4.5（2）に示す。ここで、ＭＳＤＬとＳＴＲ言語で使用しているそれぞれの記述要素は同じ表記名を用いている。また、端末の初期状態記述要素p_{s0}と最終状態記述要素p_{f0}は「idle(x)」（ここで、x=A又はB）として表している。

更に、既存の電話サービス仕様（CW,CFV,TCS：付録2.1と4.4参照）の基本仕様に関してＭＳＤＬのステップ1記述による仕様記述に本論文で提案したアルゴリズムを適用してＳＴＲ言語への変換を机上で実験した。実験結果を記述量の比較として表4.5に示す（尚、記述量の差異については4.4.2節で考察する）。変換アルゴリズムの一般的な正当性の証明は困難であるので、以下の方法によりその正当性を確認した。即ち、それぞれのサービス仕様において、ＭＳＣの情報が過不足なく変換されていることを目視で確認すると共に、ルールのインタプリタ[8]を用いて、ＭＳＣで記述した仕様と変換されたＳＴＲ言語の仕様が同一の動作をすることを確認した。

選択した4つの既存サービスは、電話サービスの特徴を代表するサービスであり、これらの特徴を組み合わせて規定されるサービス仕様については提案した変換アルゴリズムが適用できると考える。しかし、より数多くのサービスに実際に適用して、評価する実験は今後の課題である。

4.4.2　サービス仕様確認への適用

サービス仕様をＭＳＣで記述した後で、当然記述すべき要求仕様が正しく含まれているかの確認が必要となる。ＭＳＣで記述した仕様の目視での確認作業は、ＭＳＣ記述要素の時系列を逐次辿っていき、設計者の頭の中で要求仕様を断片的に組み上げながら進められると考えられる。従って、ＭＳＣで記述された仕様を断片的な意味のある端末の1つの状態遷移に変換し、且つそのような状態遷移を図形シンボルに変換して逐次実行表示できれ

表4.5　記述量の比較

サービス名	MSDL(step数)	STR（ルール数）
POTS	22	6
CW	46	8
CFV	33	6
TCS	14	4

ば、サービス要求仕様の確認支援として有効である。

表4.5には、ＭＳＤＬで記述した仕様のステップ数と変換した得られたＳＴＲ言語のルール数の比較が示されている。変換したＳＴＲ言語のルール数はＭＳＤＬの記述量と比べ少なくなっている。これは、ルール適用の再利用性を示すと共に、ＭＳＤＬの数ステップがＳＴＲ言語の１つのルールに対応していることを示している。そのＭＳＤＬの数ステップは端末の１つの状態遷移を表したものである。例えば、付録4.5の中から１つの例を示すと次ぎのようになる。サービス記述仕様の全体の時系列の流れの中で以下の５ステップのＭＳＤＬ記述のまとまりが、

```
dial-tone(NW,A);
dial(A,NW:v);
.if(v=idle(B))  {
  ringback(NW,A);
  ringing(NW,B);
```

以下の１つのルールに変換される。

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

この記述の意味は、端末Ａがダイヤルトーン受信状態で、端末Ｂがアイドル（空き状態）の時、端末Ａが端末Ｂにダイヤルすると端末Ｂが呼び出され(ringing)、端末Ａが呼び返し音受信状態(ringback)になることを示している。従って、ＭＳＤＬで記述された手続きを辿らなくても宣言的なルールに変換することにより、断片的な１つの遷移単位として、ＭＳＤＬでの記述の中に要求通りの仕様が定義できているかの確認が容易となる。また、変換したルールをインタプリタを用いて実行させることにより、実行状況のアニメーション表示[8]によるＭＳＤＬ記述に含まれる要求仕様の充足性の確認が更に容易となる。

以上の説明から、提案したＭＳＣ記述からルール記述への変換法はサービス要求仕様の確認支援の観点からの有用性を示していることがわかる。

4.4.3　サービス仕様競合検証への適用

ＭＳＤＬによる記述仕様から変換したＳＴＲ言語による仕様は、宣言型の仕様記述であり、２つの仕様を単純に混ぜ合わせるだけで２つの仕様の合成が可能となる。ルール適用の非決定性を検出することにより、サービス仕様競合の機械的な検出法が文献９により提

案されている。また、サービス仕様を合成した場合の異常な状態への遷移（合成前の個々のサービス仕様では定義されていない状態への遷移）を検出する方法を３章で示した。記述実験した４つのサービス仕様をＳＴＲ言語に変換し、文献９～１１による手法で競合が検出できることを確認した。例えば、CWとCFV サービスのＭＳＤＬ記述の以下に示す部分から提案した変換法を用いると、それぞれのサービスのＳＴＲ言語による記述が以下のように変換して得られる。

１）CFV サービス仕様例の一部

1-a)ＭＳＤＬによる記述例

```
dial-tone(NW,A);
dial(A,NW:v);
.if(v=not[idle(B)]) {
      m-cfv(NW,NULL,B,C);
      ringback(NW,A);
      ringing(NW,C);
```

1-b)ＳＴＲ言語への変換例

```
dial-tone(A),not[idle(B)],m-cfv(B,C),idle(C) dial(A,B):
                                    m-cfv(B,C),ringback(A,B),ringing(C,A).
```

２）CWサービス仕様例の一部

2-a)ＭＳＤＬによる記述例

```
dial-tone(NW,A);
dial(A,NW:v);
.if(v=not[idle(B)]) {
      m-cw(NW,NULL,B);
      ringback(NW,A);
      cw-ringing(NW,B);
```

2-b)ＳＴＲ言語への変換例

```
dial-tone(A),not[idle(B)],m-cw(B) dial(A,B):
                                    m-cw(B),ringback(A,B),cw-ringing(B,A).
```

上記の変換して得られた2-a)と2-b)のルールは以下に示す端末の状態で、イベントdial(A,B):が選択されると非決定となることが文献９により示されている。

端末A：dial-tone(A)

端末B：not[idle(B)],m-cw(B),m-cfv(B,C)

端末C：idle(C)

このようにして、提案した変換法を用いることによりＭＳＤＬ記述においてもサービス仕様競合検証を行うことができるようになる。

以上は、ＭＳＣによる手続き的な仕様記述をルール記述による宣言的な仕様記述に変換して得られる利点を実際に示したものであり、提案手法の有用性を示している。更に、サービス仕様競合検証において検出された矛盾を解消した結果をＭＳＣで記述された仕様に反映できれは、仕様記述支援として更に高い効果が得られると予想される。このような支援を実現するためのルール集合からＭＳＣ記述への変換法については今後の課題である。

4.5　結言

本章では、ＭＳＣで記述された通信サービス仕様をルール集合による仕様に変換する手法を提案した。特に、メッセージと端末の状態及びイベントをいかに対応させるかを示した。具体的には、ＭＳＣ記述要素を分析し、状態変換要素とイベント変換要素の抽出の考え方を述べ、それぞれの変換要素に対応する状態記述要素への対応関係を示した。

本変換法の目的に対しては、以下の成果を得ることができた。即ち、これまで、ＭＳＣで記述された仕様に対して報告のなかったシミュレーションによる仕様確認及びサービス仕様競合検証への適用可能性を示すことができた。

第５章
信号規則による通信サービス仕様設計支援法

5.1　緒言

通信サービスは一般に広く分散した通信プロセスからなる協調ネットワークとサービスユーザによって実現されている。例えば、各サービスは図5.1に示すようにサービス制御プロセスと端末制御プロセス及びユーザの利用する端末によって実現されている[1]。このようなネットワークモデルに基づくサービス仕様は協調するプロセスとユーザ間の相互動作の集合としてモデル化できる。分散協調システムでは様々な相互動作が存在し、サービス設計者が全ての可能なサービス動作を正確に仕様化することは大変困難となる。

本章では、様々なサービス設計に対応するために、再利用可能なメッセージシーケンス部品である信号規則（ＭＳＲ：Message Sequence Rules）を設計知識として持つ設計支援システムを提案する[2]~[9]。ＭＳＲを獲得するためのメッセージの抽象化と入出力メッセージ間の関係のモデル化手法とＭＳＲの適用法を示す。さらに、ＭＳＲを使ったメッセージシーケンス図（ＭＳＣ）による代表的サービス動作の仕様設計とそれに付随するサービス動作の推論により仕様の完全化を図る、２ステップからなる設計支援法を示す。提案手法を試作し、具体的なサービス仕様に適用した結果を考察する。また、ルールの再利用性についての記述実験結果を示す。

5.2　通信サービス設計

通信サービス仕様設計において、ＭＳＣはプロセス間の関係が理解し易いため、サービス仕様表現として広く使われている。ＭＳＣを使って、設計者は目的とするサービス機能を満たした代表的なサービス仕様を容易に記述することができる。次に、その代表的なサービス仕様に付随する可能なサービス動作を仕様化するために、サービス動作の全ての状態においてユーザと関連プロセス間の全てのメッセージを検討する必要がある。例えば、図5.2は基本電話サービスにおける代表的なサービス動作を表現したＭＳＣを示している。

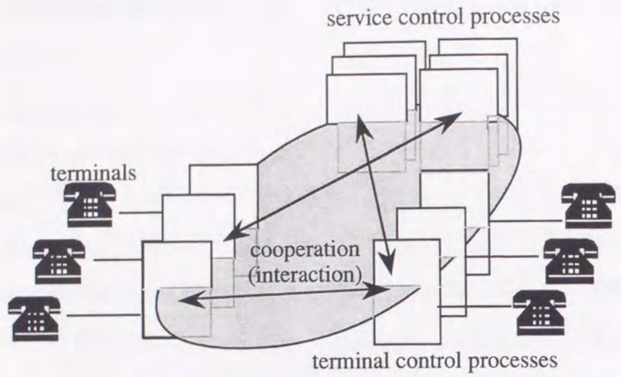


図5.1　通信サービス

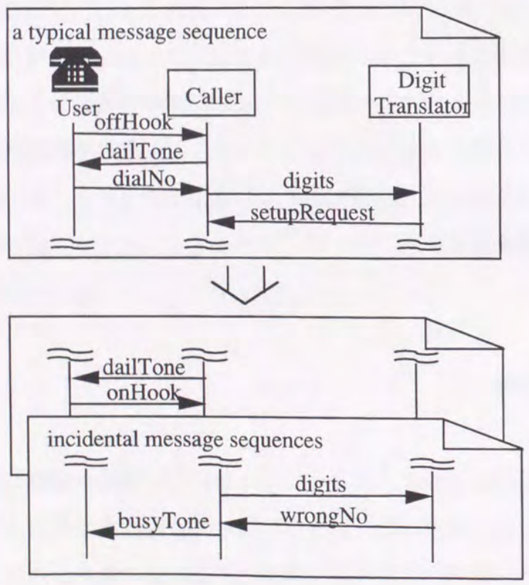


図5.2　通信サービス設計

ユーザがoffHook するとdailToneが聞こえ、ユーザはダイヤル(dialNo)し、digitsがネットワークで番号翻訳され、ルーチングされる。しかし、ユーザは早まってoffHook か又はdialNoの後に受話器を下ろすかもしれない。もし、dialNoが間違っていたら、digitsに対する応答として、DigitTranslater はwrongNo を返すかもしれない。この時、設計者はこのような代表的なＭＳＣに付随するメッセージに気付き、代表的なサービス動作に加えて、このようなサービスの振る舞いを考えてメッセージシーケンスを完全化しなければならない。関連するプロセスの数は高度なサービスでは多くなるため、このような付随するメッセージシーケンスはもっと複雑で数も増えてくる。このような状況は、設計者が通信サービスを設計することをもっと困難にする。

本章では、このような代表的なサービス動作の設計と付随的なサービス動作のＭＳＣによる仕様化をＭＳＲを使って支援する手法を述べる。

5.3　ＭＳＲ記述法

サービス動作を記述したＭＳＣに含まれるプロセスの入出力メッセージの時系列を「メッセージ系列」と呼ぶ。例えば、図5.2において、callerプロセスに着目すると以下のメッセージ1～4が系列として抽出される。

メッセージ1：ユーザからoffHook を受信

メッセージ2：ユーザへdialToneを送出

メッセージ3：ユーザからdialNoを受信

メッセージ4：DigitTranslater にdigitsを送信

5.2節に示した支援技術は再利用可能なメッセージの系列部品であるＭＳＲに基づいている。ＭＳＲはルール適用によって様々なメッセージの系列を生成する能力を備える必要があり、以下の2つのステップにより獲得する。

（1）既存サービス仕様定義で使われている具体的メッセージの抽象化

（2）抽象メッセージ間の時系列的な関係を3種類に分解

最初のステップはメッセージの再利用性を保証するためのものであり、後者のステップはメッセージ系列としての規則性を与えるものである。以下では、ルールの獲得法、表記法及び適用法を示す。

5.3.1　メッセージの抽象化

各種のサービスに使われている全ての具体的なメッセージはその送信と受信に共通する意味を持っている。例えば、図5.2に示したoffHook はユーザがcallerプロセスに対して基本電話サービスの開始を要求していることを意味する。dialToneはcallerプロセスがユーザにサービスの開始要求の受理を通知することを意味している。このように、それらの意味に応じてメッセージを抽象化することができる。例えば、offHook とdialToneはそれぞれrequest とnotifyを意味する別名として置き換えることができる。従って、様々なサービス仕様を検討し、具体的メッセージをそれらの意味に従って分類することにより抽象メッセージと呼ぶ8種類の別名を獲得した。これらのメッセージを表5.1に示す。図5.3は抽象メッセージによって再定義された図5.2の代表的なＭＳＣを示している。

抽象メッセージは以下の2つの形式で記述する。

<sender-variable>？<abstract-message>

<receiver-variable>！<abstract-message>

最初の形式はあるプロセスへの入力メッセージの記述のためのものであり、後者は出力メッセージの記述のためのものである。senderとreceiver-variable はメッセージの送出プロセスと受信プロセスを示す識別子としてそれぞれ使用する。senderとreceiver-variable は英文字で記述する。？と！はメッセージの受信と送信をそれぞれ示す。メッセージ記述例を以下に示す。

ex1）あるプロセスはユーザｘからメッセージrequest を受信する。

x?request

ex2）あるプロセスはユーザｘにメッセージnotifyを送信する。

x!notify

ex3）あるプロセスが他のプロセスｘからメッセージinvokeを受信する。

x?invoke

ex4）あるプロセスが他のプロセスｘにメッセージack を送信する。

x!ack

表5.1　抽象メッセージ

abstract message	meaning
request	・A user requests a process to execute a function.
complete	・A user cancels/completes an executing function of a process. ・A process cancels/completes an executing function of another process.
accept	・A user accepts a request from another user via processes. ・A process accepts a request from another user via processes.
notify	・A process notifies a user of the current state.
query	・A process queries something to another process with a database.
invoke	・A process invokes a function of another process.
ack	・A process acknowledges something queried by another process. ・A process acknowledges invocation from another process.
reject	・A process rejects something queried by another process. ・A process rejects invocation from another process.

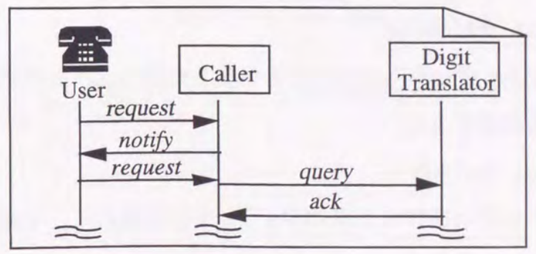


図5.3　抽象メッセージシーケンス図

5.3.2　メッセージ間の関係モデルとその形式化

図5.3に示す各メッセージは相互に関連付けられており、それらの関係はメッセージシーケンスの意味に基づいている。例えば、ユーザへのnotifyはユーザからのrequest がきっかけとなっている。DigitTranslator からのack はquery に対する応答である。さらに、ユーザからのcompleteは、もしユーザからのrequest が発生したならば、発生するチャンスを持っている。即ち、メッセージシーケンスの内容をメッセージ間の関係として分解することができる。様々なサービス仕様を検討し、図5.4に示す3種類の入出力メッセージ間の関係モデルを抽出した。ＭＳＲはこれらの関係を形式化したものであり、if-then タイプのプロダクションルール表現を用い、LHS(left-hand side) 部とRHS(right-hand side)部から構成される。LHS とRHS 部はそれぞれ条件部とアクション部として記述する。

（1）入力と出力メッセージ間の関係

抽象メッセージ記述法を使って、この関係を記述すると以下のようになる。

<sender-variable> ? <abstract-message> →
<receiver-variable> ! <abstract-message>

この表現は、もしLHS 部の入力メッセージが発生したならば、RHS 部の出力メッセージが可能であることを意味している。この記述例を以下に示す。

ex1)あるプロセスがユーザｘからrequest を受信したならば、そのプロセスはユーザｘにnotifyを送信する。

x?request →x!notify

ex2)あるプロセスがユーザｘからrequest を受信したならば、そのプロセスは他のプロセスｙにquery を送信する。

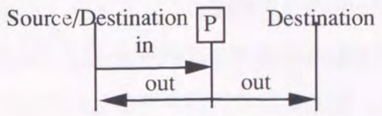
x?request →y!query

2つの変数はそれぞれ異なった英文字ｘとｙで記述される。なぜならその値がお互いに異なっていなければならないためである。

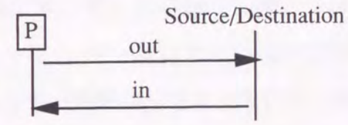
ex3)あるプロセスがユーザｘからrequest を受信したならば、そのプロセスはユーザｘにnotifyを送信するか、又は他のプロセスｙにquery ををを送信する。

x?request →x!notify | y!query

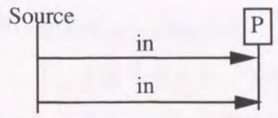
オルタナティブメッセージを記述するために、ORオペレータ（|）を定義した。



(a) Input-Output Type



(b) Output-Input Type



(c) Input-Input Type

図5.4　メッセージ間の関係モデル

ex4)あるプロセスが他のプロセスｘからinvokeを受信したならば、そのプロセスはプロセスｘにack を送信し、且つユーザｙにnotifyを送信する。

x?invoke→x!ack ∧y!notify

複数の出力メッセージを記述するために、AND オペレータ（∧）を定義した。

ex5)あるプロセスがユーザｘからcompleteを受信したならば、そのプロセスは他のプロセスｙにcompleteを送信するか、又は出力を何も送信しない。

x?complete→y!complete | NONE

あるプロセスがその実行している機能を終了することを要求した時、同じrequestを他の関連するプロセスに出力として送出するかもしれない。しかし、もし関連するプロセスがなかった場合には、そのプロセスはそれ自身の機能を終了させるだけかもしれない。このような状況を記述するために、特別なシンボルとしてNONEを定義した。このルールの適用においては、NONEはメッセージのないことを意味する特別な抽象メッセージとしてみなされる。

（2）出力メッセージに続く入力メッセージの関係

抽象メッセージ記述を使って、この関係を以下のように記述する。

<receiver-variable>！<abstract-message> →
<sender-variable> ? <abstract-message> U
(<sender-variable> ? <abstract-message>)

この表現は、もしLHS 部の出力メッセージが発生し、（ ）内の入力メッセージが無いならば、RHS 部の入力メッセージが可能であることを意味している。この記述例を以下に示す。

ex1)あるプロセスが他のプロセスｘにquery を送信したならば、そのプロセスはプロセスｘからack を応答として受信する。

x!query →x?ack U(x?ack)

もし、ack が発生したならば、ack の可能性は取り消される。

ex2)あるプロセスがinvokeを他のプロセスｘに送信し、そのプロセスがack か又はrejectをプロセスｘから応答として受信する。

x!invoke→x?ack U (x?ack | x?reject)
| x?reject U (x?ack | x?reject)

もし、ack か又はrejectが発生したならば、ack の可能性は取り消される。なぜならば、ack とrejectはそれぞれオルタナティブであるからである。

（3）入力メッセージに続く入力メッセージの関係

抽象メッセージ記述を使って、この関係を以下のように記述する。

<sender-variable> ? <abstract-message> →
<sender-variable> ? <abstract-message> U
(<sender-variable> ? <abstract-message>)

この表現は、もしLHS 部の入力メッセージが発生し、（ ）内の入力メッセージが無いならば、RHS 部の入力メッセージが可能であることを意味している。この記述例を以下に示す。

ex1)あるプロセスがユーザｘにrejectを受信したならば、そのプロセスはユーザｘからcompleteを受信する。

x?request →x?complete U (x?complete)

もし、completeが発生したならば、completeの可能性は取り消される。

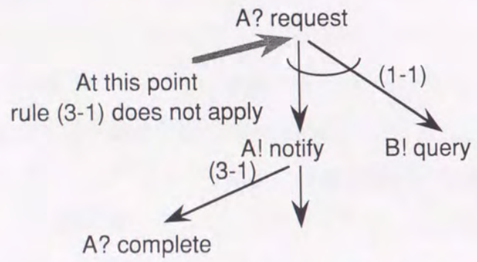
以上のようなＭＳＲをまとめてた例を付録5.1に示す。

5.3.3 ルール適用法

ＭＳＣによる代表的なサービス動作の記述を支援し、付随するＭＳＣを生成するためのルール適用法について示す。

（1）ルールのLHS に記述されたメッセージが代表的なサービス動作として記述したＭＳＣの中の着目するプロセスのメッセージに合致した時、そのルールのRHS が例外メッセージ系列生成のために適用される。しかし、これらのRHS が入力と出力メッセージ間の順序性（出力メッセージは入力メッセージの後にのみ可能である）の観点から適当でない場合は、そのRHS はそのメッセージの基では適用されない。例えば、図5.5に示すように付録5.1のルール(1-1) と(3-1) は共に入力メッセージA?request を満足している。RHS が出力メッセージを持っているルール (1-1)がA?request において適用され、RHS が入力メッセージを持つルール(3-1) はA?request に続く次の出力メッセージの時に適用される。

（2）あるプロセスにおいて、適用されたルールのRHS の中に送信又は受信変数を持ち、それぞれの変数に対応するプロセスが既に特定されていなければ、RHS のそれらの



note: (i-j) indicates a fired message sequence rule identifer in Appendix 5.1.

図5.5　メッセージ系列の入出力制約に基づくMSR適用法

メッセージは生成されない。図5.6(a)において、プロセスＡはルール(1-1) のnotify とquery のsender-variable をプロセスＴとＢとして特定できる。しかし、図5.6(b)では、プロセスＡはnotifyのsender-variable はプロセスＴとして特定できるが、query のsender-variable は特定することができない。従って、メッセージquery は生成されない。

5.4　ＭＳＲによる仕様設計支援法

提案する設計支援法は以下の２ステップからなるＭＳＣによる仕様設計工程を支援する。

（１）代表的なサービス動作の仕様化

（２）正常及び準正常も含む付随的なサービス動作の仕様化

上記の２ステップからなる設計支援の概要を図5.7に示す。各ステップの詳細な内容を以下に示す。

5.4.1　代表的なサービス動作の支援

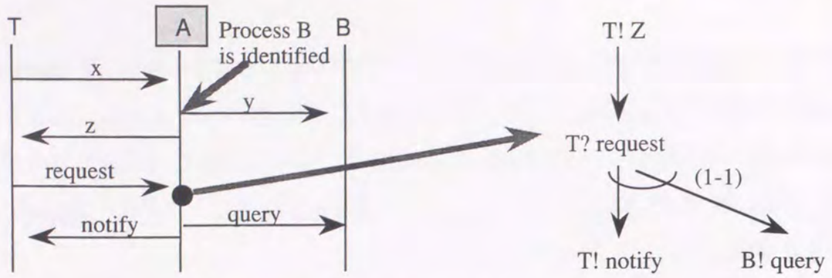
通信サービスを設計するために、サービスの特徴を最も示す代表的なサービス動作を記述したＭＳＣが最初に仕様化されるべきである。ＭＳＲを適用した代表的なメッセージシーケンスの設計手順は次のようになる。

（１）まず、設計者がメッセージを記述するためにシステムに支援を要求した時、システムは抽象メッセージの候補を含んだメニューを表示する。例えば、抽象メッセージのメニューとして、request, invoke等が図5.7に示すように表示される。

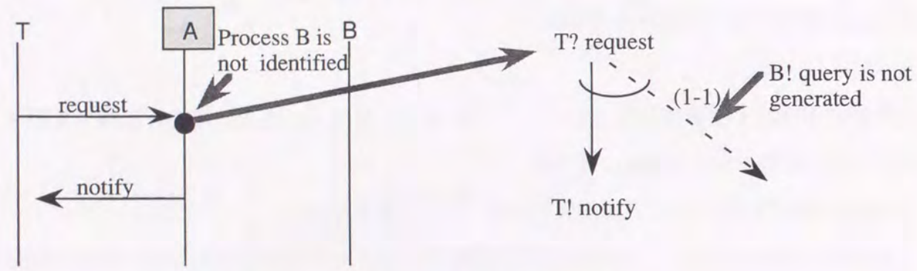
（２）設計者がメニューから適切なメッセージを選択し、その抽象メッセージに具体的なメッセージ名を与えた時、システムはＭＳＣの中にそのメッセージを書き入れる。

（３）システムはそのメッセージにＭＳＲを適用して、続いて記述されるべきメッセージの候補を獲得する。これらのメッセージもまた、図5.7に示すようにメニュー中に表示される。

（４）システムによって提示された各メッセージは抽象メッセージ名と方向性を持っているので、設計者はメッセージが実際に発生し、サービス動作として適切であるか、を判断できる。その後で、設計者はメニューの中の適切なメッセージに具体的なメッセージ名を



(a) When Process B has already been identified.



(b) When Process B has not been identified.

note: (i-j) indicates a fired message sequence rule identifer in Appendix 5.1.

図5.6　対向プロセス識別を考慮したMSR適用法

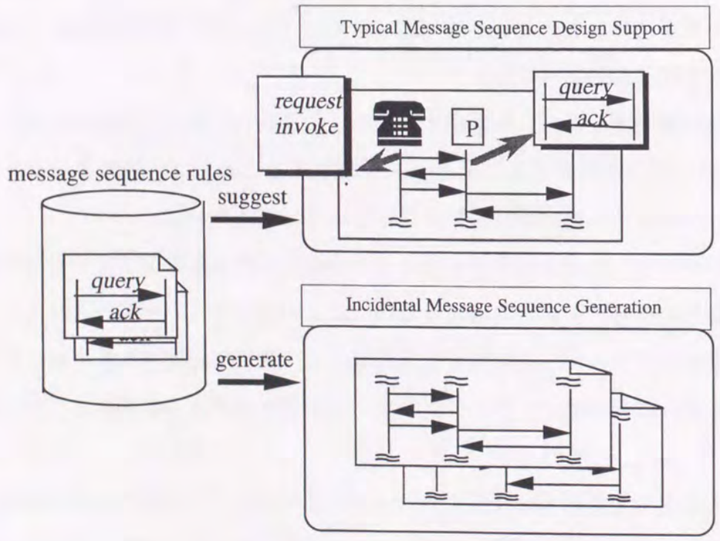


図5.7　MSRによる設計支援概要

付与する。

このようにして、システムは上記の手順を繰り返し、記述すべきメッセージの候補を提示しながら、ＭＳＣによる代表的なサービス動作の設計を促進する。

5.4.2　付随するサービス動作の支援

通信サービスをＭＳＣで完全に設計するためには、代表的なサービス動作以外の様々な付随するサービス動作を記述する必要がある。ＭＳＲを使って、システムは図5.7に示すような準正常手順も含んだＭＳＣを生成する。付随的なサービス動作を含んだＭＳＣ生成は以下の2つのステップより構成されている。

（1）各プロセスにおける可能なメッセージ系列の生成

まず、システムは代表的なサービス動作として定義したＭＳＣの中に記述されている各プロセスに着目し、それらのプロセスが送受するメッセージ系列を抽出する。次に、システムは抽出したメッセージ系列にＭＳＲを適用して、全ての状態において可能なメッセージを推論する。さらに、システムは続いて得られたそれらのメッセージにルールを繰り返し適用して、プロセス毎の可能なメッセージ系列を生成する。

（2）プロセス間のメッセージ系列の合成

システムは各プロセスのメッセージ系列において送受のメッセージの関係が一致するプロセス間のメッセージ系列を生成する。システムは生成された準正常手順を含む付随的に可能な全てのＭＳＣを提示する。これらのＭＳＣは抽象メッセージで構成されているので、設計者は各ＭＳＣがどのようなサービス動作に対応しいるかを判断し、具体的なメッセージ名を付加する。

5.5　具体的な設計支援例

本節では、着信転送(CFV)サービスの例を使って、提案した支援手法を説明する。着信転送サービスは着信呼を指定された転送先に転送するサービスである。ユーザは転送先の番号に続けてactivation codeをダイヤルすることによってこのサービスを起動できる[10]。サービスを提供するために、図5.8に示すように以下のようなプロセスが必要である。

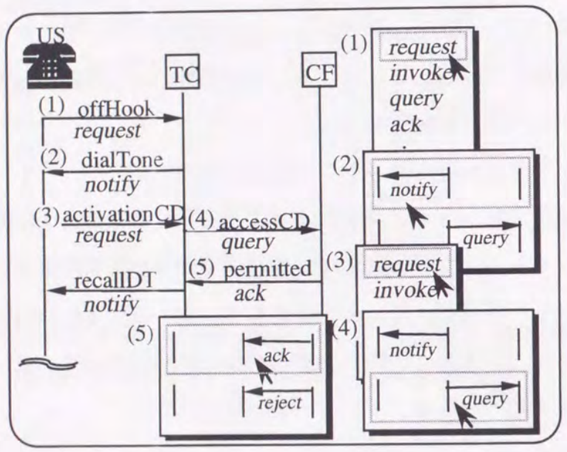


図5.8　代表的なサービス動作の設計

（1）サービス制御プロセスＣＦは転送先番号の保持と呼びの転送機能を提供する。

（2）端末制御プロセスＴＣはサービスを起動するためにユーザＵＳとプロセスＣＦ間の相互通信機能を提供する。

以下の例では、サービス設計者が着信転送サービスの起動に関する仕様を設計している場合を示す。付録5.1のＭＳＲを用いて具体例を示す。

5.5.1　着信転送サービスの代表的動作の設計

設計者が着信転送サービスの起動に関する仕様の代表的な手順に相当するＭＳＣを記述しはじめた時、システムはメニューに8種類の全ての抽象メッセージを表示し、設計者が最初に記述すべきメッセージを選択するのを要求する。ユーザＵＳがまず受話器を外して、プロセスＴＣにサービスの開始を要求していることを想定しながら、設計者は抽象メッセージであるrequest を選択し、具体的なメッセージ名としてoffHook を、図5.8(1)に示すように割り当てる。システムはモニタ上にoffHook を描き、ＭＳＲをoffHook に適用するＴＣからの出力メッセージを推論する。適用されるルールは以下のルールである。

x?request →x!notify | y!query

offHook はrequest として意味付けられているので、発側プロセスＵＳには、送信変数であるｘがセットされる。ＵＳはまたnotifyの受信変数としてもセットされる。query の受信変数はｘと異なるべきであるので、ｙはプロセスＵＳとＴＣ以外のプロセスＣＦに付与される。この結果、ＵＳにnotifyを送出するか、又はＣＦにquery を送出するか、となる。offHook に続くこのようなオルタナティブメッセージは図5.8(2)のようにメニューとして表示される。システムの提示に基づき、設計者はnotifyを選択し、メッセージdialToneをnotifyに付与する。システムはＵＳからの後続の出力メッセージを獲得するために、dialToneにルールを適用しようとする。しかし、ルールはヒットしない。従って、システムは全ての種類の抽象メッセージを表示する。設計者は、図5.8(3)に示すようにＣＦＶサービスユーザを識別するメッセージであるactivationCDのためにメニューからrequest を選択する。システムはactivationCDを書き込み、ルールをそのメッセージに適用する。その結果はoffHook にルールを適用した場合と同じになるが、設計者は図5.8(4)に示すようにaccessCDメッセージに対応するＣＦプロセスに対するquery を選択する。accessCDにルールを適用した結果は以下のようになる。

x?query →x!ack ｜x!reject；ＴＣプロセスへのack か又はrejectの送出

このack はプロセスＣＦがプロセスＴＣを経由してユーザＵＳからサービス起動要求を確認したことを示す。このrejectはプロセスＣＦがその要求を拒否したことを示す。ackは着信転送サービスの起動に関する正常手順として適当であるので、設計者はack を選び図5.8(5)に示すようにメッセージ名permitted を割り当てる。

システムは記述されたメッセージにルールを繰り返し適用して、後続のメッセージの候補を提示する。設計者は提示されたメッセージの候補の判断と具体的なメッセージ名の付与だけの作業により、代表的なサービス動作を記述したＭＳＣを完成させることができる。

5.5.2　例外メッセージ系列の生成

本節では、図5.8で示した代表的な着信転送サービスを記述したＭＳＣを使って、例外メッセージ系列を生成する例を示す。

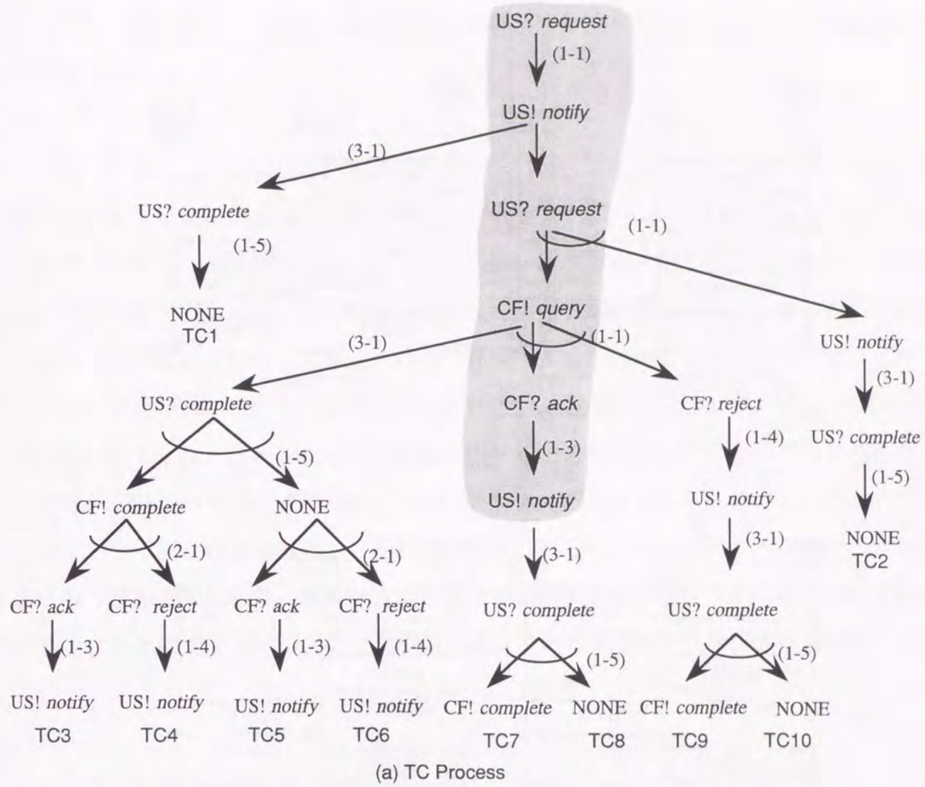
（1）ＴＣとＣＦプロセス間の可能なメッセージ系列の生成

各プロセス毎の代表的なメッセージ系列にルールを前向きに適用して、可能なメッセージ系列がツリー形式で生成される。ツリーの各ノードは入力又は出力メッセージに対応している。5.3.3節で示したルール適用法と付録5.1のルールを使って、プロセスＴＣとＣＦの可能なメッセージ系列が図5.9に示すように生成される。

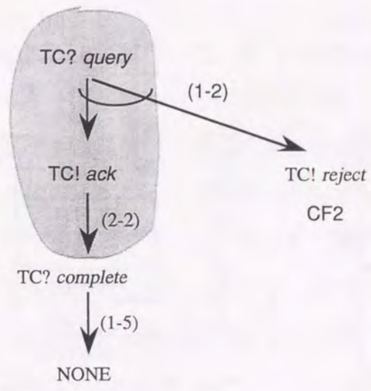
（2）ＴＣとＣＦプロセスのメッセージ系列の統合

次に、システムは入力メッセージと出力メッセージ間の対応関係に従ってＴＣとＣＦプロセス間のメッセージを統合する。プロセスＴＣにおけるメッセージ系列ＴＣ６、ＴＣ３、ＴＣ７とＴＣ１０がプロセスＣＦのＣＦ２、ＣＦ１、ＣＦ１とＣＦ２のそれぞれと統合できる。メッセージ系列ＴＣ１とＴＣ２はプロセスＣＦとインタラクションを持たないので、システムはこれらの正当性を判断することができない。その他のメッセージ系列であるＴＣ５、ＴＣ４、ＴＣ８とＴＣ９はどのメッセージ系列とも統合できない。

最後に、システムはＣＦＶサービス起動に関する正常と準正常を含む図5.１０に示すような可能なＭＳＣを提案する。設計者は各ＭＳＣがどのようなケースに対応するかを決定し、それらのＭＳＣに具体的なメッセージ名を付与することができる。例えば、図5.１０(a)はユーザがactivationCDをダイヤルした後に早まって受話器を置くと、ＣＦプロセスはaccessCDに対する応答としてrejectを返送する時に発生する。この時、設計者は具体的



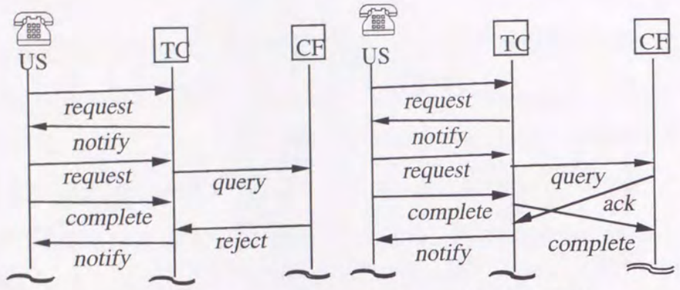
(a) TC Process



(b) CF Process

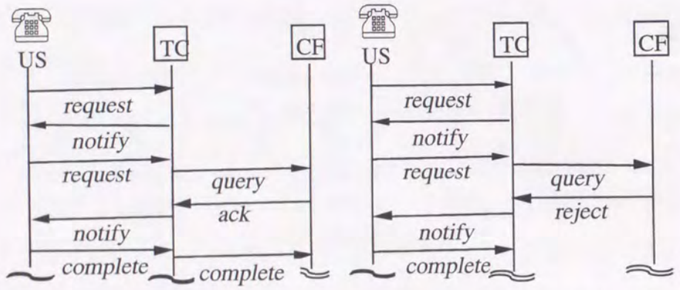
note (1) (i-j) indicates a fired message sequence rule identifer in Appendix 5.1.
(2) The hatched part of the message sequence is the typical message sequence.
(3)TCi and CFi indicate some derivation from the typical message sequnce.

図5.9　TCとCFプロセスにおいて生成されたメッセージ系列



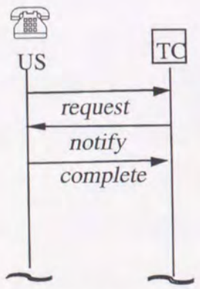
(a) TC6-CF2

(b) TC3-CF1

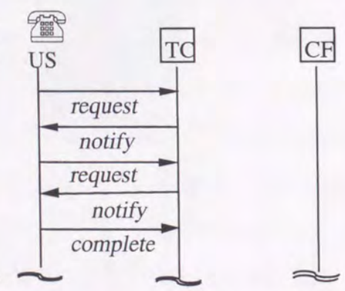


(c) TC7-CF1

(d) TC10-CF2



(e) TC1



(f) TC2

図5.10　TCとCFプロセス間で一致した付随的なサービス動作

メッセージ名として、onHookとwrongCD を抽象メッセージであるcompleteとrejectにそれぞれ付与する。

上記の手順により生成された例外ＭＳＣの中には、非決定的な振る舞いを引き起こす記述が含まれている。例えば、図5.１０(a)はプロセスＴＣがcompleteを受信した後に、ＣＦプロセスに対して何も送出しないことを示している。一方、図5.１０(b)はＴＣプロセスが同じ状態において、ＣＦプロセスにcompleteを送信していることを示しているが、動作の決定は具体的なメッセージに設定されるパラメータとその時のプロセスの状態のどちらかににによって行われ、それらは異なるかもしれない。その他の例として、図5.１０(c)と(d)はプロセスＣＦがack とrejectをそれぞれ送出することを要求しているが、ＣＦプロセスの動作は、query パラメータの分岐によって決定されるべきである。このようにシステムはただ全ての可能な状況を提案し、設計者が見逃したこのような注目すべき状況に対して、設計者の注意を換気する。本設計支援法により、設計者はシステムから提示された候補の判断及び具体的メッセージ名の付与を繰り返しながら通信サービス仕様を設計できる。

5.6　設計支援環境の試作

提案手法をLISP環境においてWS上に試作した。また、設計した仕様を実行し、その状況を視覚的に表示するシステム[11][12]において、仕様を確認した。図5.１１に設計環境の構成を示す。各WSの実行システムは通信プロセスの役割を担い、ＦＳＭ(Finite State Machine)ベースの仕様に従って入力イベントから出力アクションを獲得する[11]。表示システムはサービスの実行イメージを表示することによって、実行システムに対する視覚的なマンーマシンインタフェースを提供する[12]。そのインタフェースはユーザがサービスを理解することを支援し、また設計者が仕様を確認することができる。

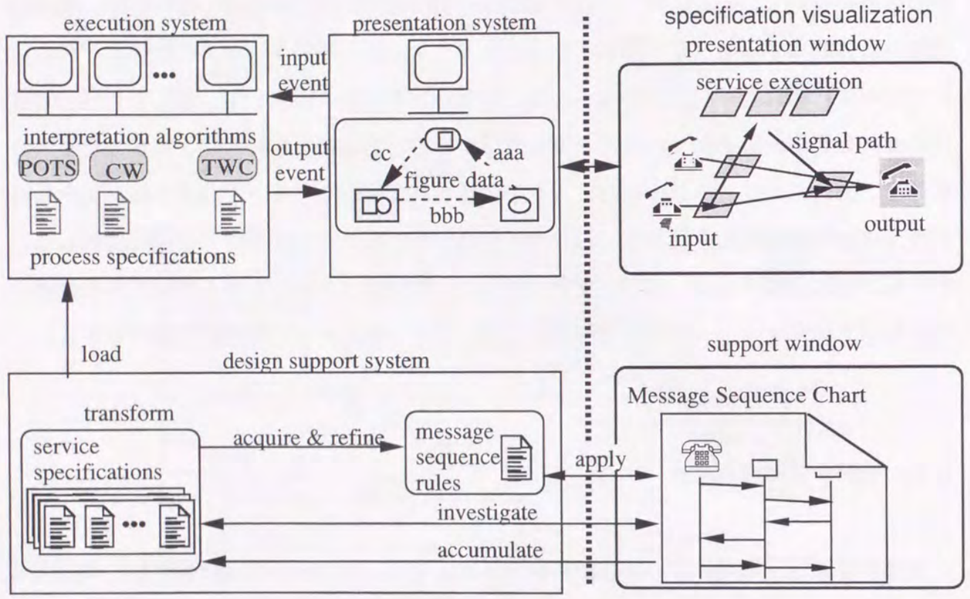


図5.11　統合設計環境

5.7　評価及び考察

5.7.1　ルールの再利用性

ＭＳＲの再利用性を検証するために、基本電話サービス(POTS)、話中着信(CW)及び三者通話(TWC) サービス[10]の各端末制御プロセスの設計を対象とした適用実験を行った。

5.3.1節で示した8つの抽象メッセージに従ってPOTS仕様からルールを獲得し、そのルールをCWサービスの仕様設計のために適用した。さらに、POTSとCWサービス仕様から獲得したルールをTWC 仕様に適用した。実験の結果を表5.2に示す。CWにおいて付加されたメッセージの98%はPOTSから得られたルールによって生成することができた。TWC において付加されたメッセージの97%はPOTSとCWから獲得されたルールによって生成することができた。

サービスユーザからのサービスに依存する少数のメッセージ系列は生成することができなかった。また、CWとTWC の各プロセスによる同時的な制御に関する少数の出力メッセージも生成することができなかった。しかし、これらの実験結果はルールの高い再利用性を示し、ルール獲得法が再利用可能なメッセージ系列部品を開発するために効果的であることを示している。

5.7.2　設計支援法の有効性

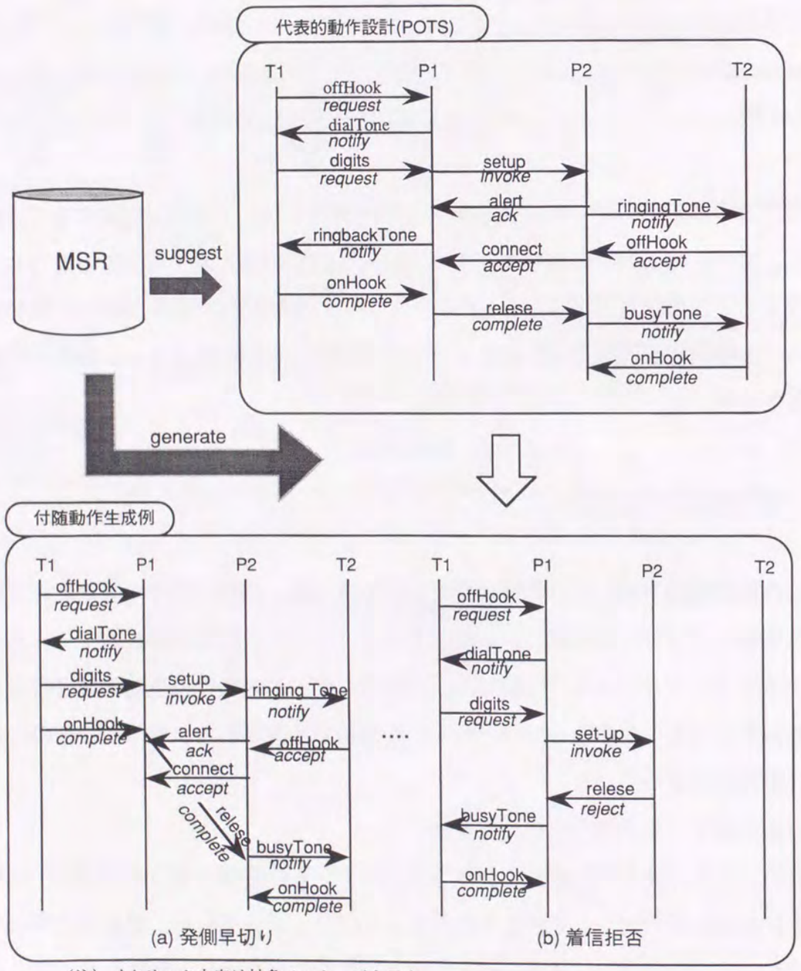
試作した設計環境を用いて、POTSの設計実験を行った。POTSは発着の端末とその端末を制御する発着のプロセス(P1,P2) から構成することとした。POTSの代表的なサービス動作を図5.12に示すようにＭＳＣで記述し、付随的なサービス動作の生成実験を行った。生成されたＭＳＣの主なものとしては、図5.12に示す発側早切り、着信拒否等の付随するサービス動作が含まれていた。

（1）付随的なサービス動作生成の充足度

生成されたＭＳＣがどの程度まで例外を包含しているかを調べるため、POTSの状態遷移表をもとに生成されたＭＳＣを表5.3に示すようにマッピングした。表5.3の網掛け部分は代表的なサービス動作を記述する時に規定されたところを示し、それ以外のところは付随的動作として生成されるはずのところであり、この例外部分の遷移を示すシーケンスは

表5.2　MSR適用実験

service	added messages	generated messages	generation rate
POTS	53	—	—
POTS + CW	53	52	52 / 53 98%
POTS + TW	90	87	87 / 90 97%



（注）イタリック文字は抽象メッセージを示す。

図5.12　POTSの設計手順と付随動作生成例

表5.3　POTSの状態遷移表と例外生成の充足度

event ＼ state		S0 空	S1 dial Tone 送出中	S2 alert受信待	S3 通話中	S4 ringing 送出中	S5 ringback 送出中	S6 busyTone 送出中
端末 | P 間	off Hook *(request/accept)*	S1 dial Tone *(notify)*	☒	☒	☒	S3 connect *(ack)*	☒	☒
	on Hook *(complete)*	☒	S0 —	S0 relese *(complete)*	S0 relese *(complete)*	☒	S0 relese *(complete)*	S0 —
	digits *(request)*	☒	S2 set-up *(invoke)*	☒	☒	☒	☒	☒
P1 | P2 間	set-up *(invoke)*	alrt *(ack)* S4 / ringing *(notify)*	S0 relese *(reject)*	S0 relese *(reject)*	S0 relese *(reject)*	S0 relese *(reject)*	S0 relese *(reject)*	S0 relese *(reject)*
	alert *(ack)*	☒	☒	S5 ringback *(notify)*	☒	☒	☒	☒
	connect *(accept)*	☒	☒	☒	☒	☒	S3 —	☒
	relese *(complete/reject)*	☒	☒	S0 busy Tone *(notify)*	S0 busy Tone *(notify)*	S0 —	☒	☒

（注）(1) ▭ ：代表的なサービス動作設計で規定したところ。 その他は例外として生成された。

(2) ☒ ：ありえない組み合わせ又はイリーガルイベントとして廃棄され例外生成には無関係

全て生成されていた。一方、生成されたＭＳＣからＳＤＬ記述への変換を考えた場合、着側プロセスの状態（通話中着信、発信中着信等）、又はサービス契約情報・メッセージパラメータ等のデータを分析して分岐する記述を必要とする場合がある（例えば、表5.3のset-upイベントの各カラム）。ＳＤＬ変換の支援法としてシステムが設計者に設定箇所を提示し、このような情報を入力するインタフェースを設ける必要がある。

（2）意味のあるＭＳＣの絞り込み

サービスの特徴を記述した代表的なＭＳＣからプロセス毎のメッセージ系列を抽出し、この系列に対してＭＳＲを前向きに適用して、プロセス毎に可能性のあるメッセージ系列をツリー形式で生成している。このツリー展開において、5.3.2節の（1）のタイプのルールは入力メッセージに対してメッセージを他のプロセスに送出するルールであるが、送出する先のプロセス名はＭＳＲだけでは決定できないため、代表的なサービス動作を記述したＭＳＣで定義された可能な全てのプロセスに対してメッセージを送出することを想定したツリーが生成される。生成されたプロセス毎のツリーを各々のプロセス間で照合することにより付随するサービス動作の候補としてＭＳＣが生成されるが、ＭＳＲからは可能性のある全てのケースが網羅されるため、設計対象のサービスに適合するＭＳＣの他に、その断片的な一部分を示すシーケンス図も生成される。設計者は生成されたＭＳＣから意味のあるシーケンス図を選択する必要がある。

このような断片的なＭＳＣは、ユーザプロセスでのユーザ動作の不確定性に起因している。設計対象のサービスにおいてユーザ動作を規定した第2章で示したユーザの視点からの仕様を記述した状態遷移知識を適用することによって絞り込みの支援が可能となる。

5.8　結言

本章では、通信サービスを分散プロセスモデルとして仕様化するための支援システムを提案した。システムは設計知識としてＭＳＲを持ち、それにより支援することを特徴とする。ＭＳＲの獲得法として、メッセージの抽象化とメッセージの入出力間の関係のモデル化による手法を示し、ＭＳＣでサービス動作を記述するためのルール適用法を示した。ＭＳＲを用いて、設計者がユーザと関連プロセス間の可能なインタラクションを完全化する支援法として、2ステップからなる方法を提案した。ステップ1では、ＭＳＣによる代表的なサービス動作の記述支援のために候補メッセージを提案し、ステップ2では、付随的なサービス動作として仕様化するために可能なメッセージ系列を推論し、ＭＳＣとして生成する。提案手法を試作し、ＭＳＲの再利用性と支援法の評価を行った。3つの具体的なサービスに適用した記述実験では、再利用性の高さを示し、ＭＳＲの獲得法の有効性を示した。また、基本電話サービスの仕様化実験では、設計者が全ての状態での可能なメッセージを検討することから開放されることを示し、設計者は支援システムからのメッセージ候補の判断と具体的メッセージ名の付与だけで、通信サービスを仕様化できることを確認した。

第６章
高度ＩＮによるパーソナル通信サービスのための網機能配備法

6.1　緒言

通信サービスの多様化に伴い、通信のパーソナル化、マルチメディア化、インテリジェント化が望まれている。特に通信のパーソナル化のための網機能は自動車電話、携帯電話、PHS(Personal Handy Phone System)等の移動通信システムの発展に伴い、ネットワークの基本技術として確立されつつある[1]。パーソナル通信サービスでは、ユーザに付与されたパーソナル番号のモビリティ、サービスのパーソナル化とサービスのポータビリティ（可搬性）をネットワークが提供する必要がある[1][2]。これまでのシステムでは端末に付与された番号のモビリティ（移動性）が実現されているだけであり、パーソナル通信サービスとしての高度化が要求されている。一方、サービス開発の即応化とサービスのカスタマイズ化等を狙ったネットワークのインテリジェント化の研究も着実に進められ、高度インテリジェントネットワーク（ＩＮ：Intelligent Network ）システム[3] として構築されつつある。

本章では、パーソナル通信サービスを実現するための高度ＩＮに基づく新しいネットワークアーキテクチャを提案する[4]~[12]。具体的にはパーソナル通信サービスのための所要機能を示し、高度ＩＮでの実現法を明確化する。特にサービスのパーソナル化とサービスのポータビリティ（可搬性）の高度ＩＮでの実現法について示す。また、サービス展開のためにグローバルなポータビリティを提供するための方式について考察する。

6.2　パーソナル通信サービスと要求条件

6.2.1　コンセプトと要求条件

本節では、パーソナル通信サービスを提供するための通信網への要求条件について、ユーザの視点から考察する。

パーソナル通信のコンセプトは[1]、図6.1に示すように「いつでも」「どこでも」「だれとでも」「どんなメディアでも」「個人対応のサービスが受けられる」ことである。例えば、会社でも、家庭でも、出張先でも、移動中でも、歩きながらでも通信することができ、且つ１ケ所で契約したサービス及び自分でスケジューリングしたサービスが変わらずに受けられる。また、自分の現在の状態、例えば会議中、お客様と応接中、運動中、休息中、や発信者、例えば家族から、上司から、友人から、恋人から等、に応じて、また時間帯を識別して着信方法を自在に設定できる。どの端末でも、例えは電子手帳、自宅と会社及び出張先のパソコン、携帯電話、ページャ等が自分の端末として利用でき、自分に必要な情報、例えば私宛の電子メール、ナビゲーション情報、料理店の情報等が得ることができる。さらに、通信品質（聞きづらくてもよい、急がない、高臨場感）、課金方法（品質に応じた料金、割り勘課金、第３者課金等）についても様々に組み合わせて利用することができる。

このようなパーソナル通信では、ユーザがどこへ行っても通信できることが重要であり、ユーザ対応に付与されたパーソナル番号又は端末のポータビリティが確保されている必要がある。さらに、ユーザ対応のサービスがユーザにより任意に定義できるサービスのパーソナル化とユーザがどこに移動しても契約したサービスを受けることができるサービスのポータビリティが確保されていることである。課金についても契約ユーザに課金される必要がある。音声、データ、画像等の各種メディアに対応できる必要がある[13]。通信網は様々なネットワーク、例えば固定網(PSTN 、ISDN、データ網、LAN 、WAN 等）、移動網により構成されており、ユーザがネットワークのこのような構造を意識して通信することを避けるためにシームレス化が必要となる[14]。以上をまとめると通信網に対しては以下の条件が要求される。

- ●番号又は端末のポータビリティ
- ●サービスのパーソナル化
- ●サービスのポータビリティ
- ●サービスのマルチメディア化
- ●通信網のシームレス化

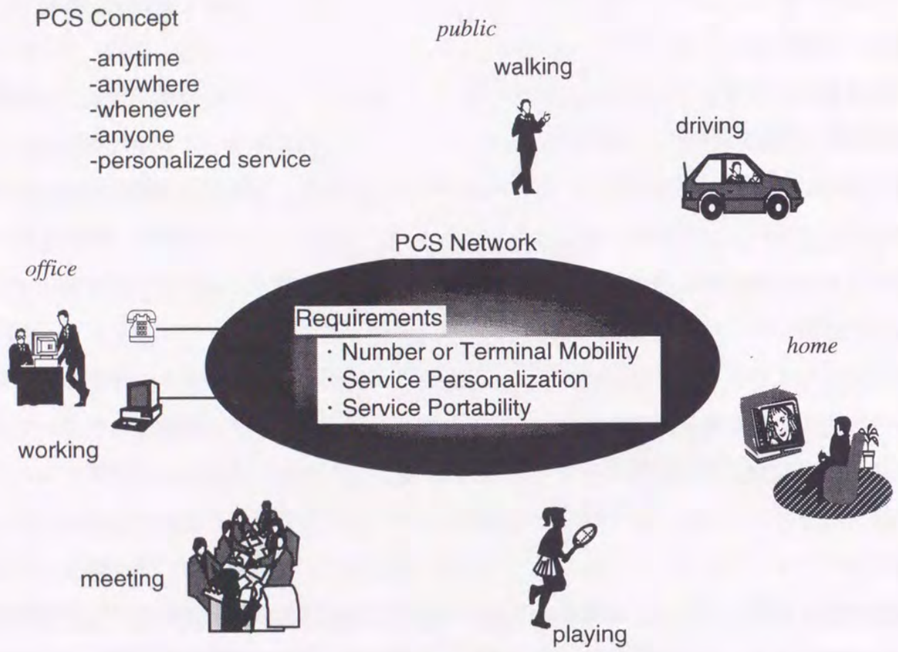


図6.1 パーソナル通信サービスと要求条件

6.2.2 通信網の所要機能

前提とするパーソナル通信のネットワークモデルを示し、必要となるネットワーク機能を明確化する。

パーソナル通信ネットワーク(PSCN)と要求条件を図6.2に示す。提案するPSCNはPSTN、ISDNのような有線系アクセスと移動体通信網のような無線系アクセスを提供する必要がある。PSTNとISDNによるＩＮでは伝達系(Network Transport)に配備されたサービス交換ポンイト(SSPs)と高機能系(Network Intelligence)に配備されたサービス制御ポイント(SCPs)より構成され、SSPsとSCPsは互いにNo.7網(Network Signal)により接続されている。従来の電話手順とISDN端末のアウトチャネルアクセスを可能とするため、UNI において、呼に関連するユーザインタラクション(call related)と呼に関連しないユーザインタラクション(call un-related) の2つのユーザインタラクションを提供しなければならない。このモデルにおいては、DSS1プロトコル、即ちcall relatedとcall un-related のアウトチャネルインタラクションが無線系アクセスとして利用する必要がある。ユーザローミングを提供するために、他のPCSNとのGWを介してた相互接続を行う必要がある。

PCS のために要求される所要機能を以下に示す。

（1）認証

パーソナル通信サービスでは、移動するユーザに対してサービスを提供する必要があるため、ユーザがサービス登録している本人であるかをネットワークが検証する必要がある。このような機能は認証制御と呼ばれ、ユーザ又は端末とネットワークがインタラクションしながらパーソナル番号とパスワードにより行う方法と認証キーと乱数による演算により行う方法がある。後者の方が「強い認証」機能を提供でき、アウトチャネル手順により行われる。前者は一般的にインチャネル手順により行われる。

（2）基本接続

パーソナル通信においては、端末及びパーソナルモビリティを保証する必要がある。モビリティを保証し、且つ通信を保証するためには、位置登録、追跡接続、着信接続の基本機能を提供する必要がある。位置登録は、移動するユーザの現在の位置をネットワークが認識するために必要な機能である[1]。位置登録はユーザが移動した場所の任意の端末からユーザがネットワークと対話しながら行う方法とICカードを挿入することにより端末が自動的行う方法がある。また、無線系アクセスの場合には、位置登録エリアを跨がって移

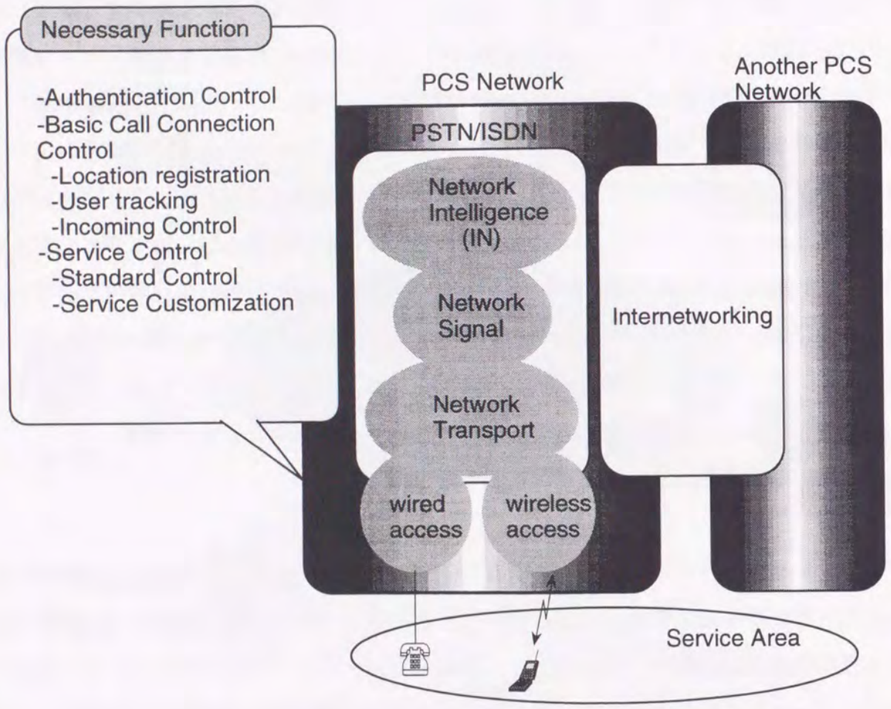


図6.2 PCS網の所要機能

動した場合には、端末が自動的に検出して行われる。追跡接続は、ユーザに対して発呼があった場合に、位置登録情報からユーザが現在在圏しているエリアまでのルーチングと回線を設定する機能である。着信接続は、在圏するエリア内のユーザを呼び出し、認証を行った後、回線を接続する機能である。無線系アクセスの場合には、「一斉呼び出し」により端末との無線チャネルを確保して行われる。また、無線エリアが変わっても通信の継続性を保証するために「ハンドオーバ」機能が必要となる。

（3）サービス制御

サービスのポータビリティを確保するために有線系アクセスと無線系アクセスに依存しないサービス制御機能が必要となる。また、パーソナルサービスをユーザが任意に定義し、条件を動的に変更できるためのには、サービスカスタマイズ機能が必要となる。さらに、既に提供しているサービスと新たに追加したサービスが矛盾なく動作することを保証するためにサービス競合制御機能が必要となる。競合制御には静的な制御法と動的な制御法がある。静的な制御法としては、ユーザが移動した場所の端末を使って位置登録した場合に、ユーザが契約しているサービスがその端末で提供できる能力を持っているかを検証し、制御する機能である。また、動的な制御法とは、サービス実行時に提供する機能である。

パーソナル通信サービスのコンセプト及び網構成上の条件から導き出される網が具備すべき機能をまとめると以下のようになる。

- 認証制御
- 基本接続制御
  - 位置登録
  - 追跡接続
  - 着信接続
- サービス制御
  - 標準サービス制御
  - サービスカスタマイズ
  - サービス競合制御

6.3　パーソナル通信網アーキテクチャ

本節では、高度ＩＮを核とした通信網においてパーソナル通信サービスを提供するためのネットワークアーキテクチャを明らかにする。

6.3.1　高度ＩＮによるアーキテクチャ

（1）高度ＩＮ[3]

ITU-T においてＩＮの標準化が進められており、固定電話網を対象とした能力セット1(CS-1:Capability Set-1)が既に勧告化されており、移動通信も含めた能力セット2(CS-2)（高度ＩＮ）の検討が現在進められている。高度ＩＮは伝達系と高機能系から構成され、高機能系はさらにサービス管理系とサービス制御系から構成されている。図6.3に高度ＩＮの分散機能モデルを示す。SSP では基本呼処理機能が実現され、交換機の呼処理プロセスを発側／着側に論理的に分割した各部分(Call Segmentと呼ぶ）を1つの制御単位としてモデル化されている。各Call Segmentは基本呼状態遷移モデル（BCSM:Basic Call State Model)に沿って動作する。サービス制御のためのインタフェースは規格化され、INAP(Intelligent Network Application Protocol)として標準化されている。SCP(Service Control Point)ではサービス制御機能(SCF）とサービスデータ機能(SDF）が定義され、SMP(Service Management Point）からダウンロードされたサービス論理プログラム（ＳＬＰ）が実行できる環境が提供されている。ＳＬＰはサービス生成環境(SCE）において作成され[15]、従来のシステム開発者でなくてもサービスを実行するプログラムが作成できる。サービスは、BCSMのDP(Detection Point）からのイベント通知とDPへの制御指示を書き下したＳＬＰにより制御される（付録6.1）。

（2）ネットワークアーキテクチャ

6.2節で抽出した機能を提供するための網機能配備を図6.3に示す。SCP のSDF には、端末及びパーソナルモビリティのための位置情報、認証制御のための認証キー及びサービスのカスタマイズ情報等を格納するためるLR(Location Register）を配備する。LRは端末とパーソナルモビリティを独立に保証する必要があるため、T-LRとP-LRをそれぞれ分けなければならない。SCP のSCF 内に配備されるＳＬＰはユーザによるサービスのカスタマイズを許容するSLP(C)(Service Logic Program for Customization）とサービス対応のネッ

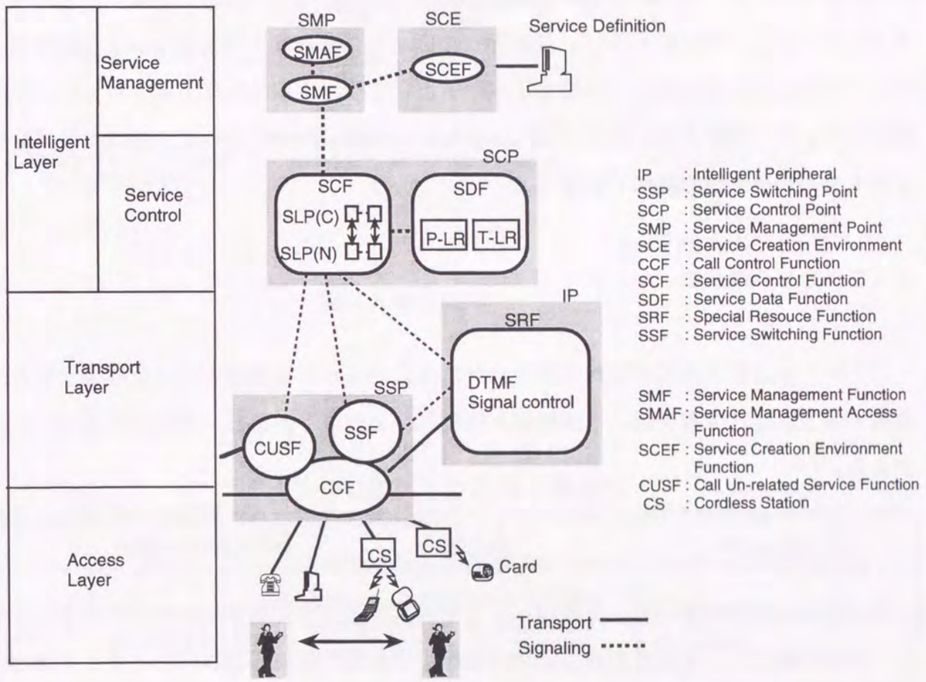


図6.3 高度INによるPCSN

トワーク制御を実現するSLP(N)(Service Logic Program for Network Control) に分割して定義し、配備する。SSP のSSF にはインチャネルとアウトチャネルによる位置登録と認証制御を実現するために、呼に関連するユーザインタラクションをサポートするためのCCF/SSF におけるBCSMと呼に関連の無いユーザインタラクションをサポートするためのCUSF(Call Un-related Service Function)におけるBCUSM(Basic Call Un-related State Model)を具備する。BCUSM を図6.4に示す。PIA(Point In Activation)はユーザ網間でのアソシェーションの状態を示し、各DP(Detection Point) はアソシェーションの要求／解放およびユーザからのデータ受信をイベントとして検出するために定義されている。また、各種メディアを制御するために、SRF(Service Resouce Function) にテキスト音声変換、プロトコル変換、等の機能を配備する。

6.3.2 ＳＬＰ構成法

パーソナル通信を実現するためのＳＬＰには、パーソナル通信のために必要となる基本機能である認証、位置登録、追跡接続を実現し、また以下のＳＬＰ構成条件を満たす必要がある。

（1）SCF からはhalf call 単位でサービスが制御されるため、発BCSMと着BCSMのそれぞれの制御ために独立したＳＬＰを配備する必要がある。

（2）１つの呼びにおいて、多数のＳＬＰを実行するためにはサービスプラットフォームの処理能力上、不利であり、同時に生起する頻度が高い機能は同一のＳＬＰ内で定義する必要がある。従って、位置登録機能と位置登録時に行う認証処理は同一のＳＬＰで実現する。

（3）同じ機能を複数のＳＬＰに定義すると、仕様変更時に影響箇所が多くなり、保守運用上好ましくない。従って、実現する機能は極力１つのＳＬＰに記述すべきである。

上記の観点から、表6.1に示すＳＬＰが定義できる。発信認証ＳＬＰはパーソナル通信ユーザからの発信要求時に起動し、認証に必要な情報（認証鍵等）を伝達ノードに通知する。着信認証ＳＬＰはパーソナル通信ユーザへの着信要求時に起動し、認証に必要な情報（認証鍵等）を伝達ノードに通知する。位置登録ＳＬＰはパーソナル通信ユーザからの位置登録要求時に起動し、認証正常の場合に位置登録を行う。起動ポイントはBCUSM のAc-

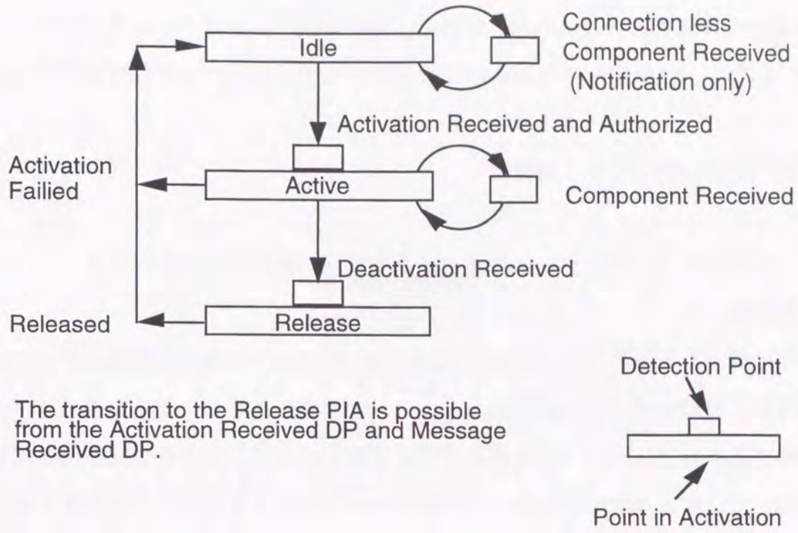


図6.4 BCUSM

表6.1 PCSのためのSLP構成法

	name of SLP	features	triggered DP
SLP-1	Origination Authentication	Send authentication data to transport layer at PCS user origination	Origination Attempt in BCSM
SLP-2	Termination Authentication	Send authentication data to transport layer at PCS user termination	Termination Attempt Authorized in BCSM
SLP-3	Location Registration	Register location data if authenticated	Activation Received and Authorized in BCUSM
SLP-4	Incoming Connection	Get location data in SDF and route calls	Analyzed Information in BCSM

tivation Receieved and Authorized である。着信接続ＳＬＰはパーソナル番号による接続要求時に起動し、位置情報の取得及びその情報に基づくルーチングの指示を行う。

6.3.3　パーソナルモビリティ制御法

提案ネットワークアーキテクチャに基づく基本接続の概要を図6.5に示す。

（1）位置登録

ユーザの位置をネットワークに登録する方法は、有線系アクセスと無線系アクセスでは手順が異なる。有線系アクセスの場合は、ユーザが移動した場所の端末を使って、ユーザ自身が登録する場合とICカードを端末に挿入して自動的に設定できる方法がある。本手法においては、インチャネルでアクセスする場合とアウトチャネルでアクセスする場合がある。無線アクセス系については、端末又はICカードを持ったユーザが位置登録エリアを跨がって移動した場合に、自律的にアウトチャネルから位置登録される方法が実用化されている。インチャネルの場合は、情報分析完了DPから起動されたSCF 内のＳＬＰがSRF を制御し、端末から入力されるコードをSRF で受信・解釈してSDF のLRに登録される。アウトチャネルの場合には、ベアラと関係しないBCUSM からのActivation Received DPからのトリガにより起動されるＳＬＰでSRF 制御を行うことなくSDF のLRに位置が登録される。

（2）追跡接続

追跡接続は、着番号として入力されたパーソナル番号をキーとして現在の位置情報を検索し、在圏エリアまでのルーチングすることにより行われる。発側BCSMの情報分析完DPにTDP(Trigger Detection Point)を設定し、トリガを検出して着信接続ＳＬＰを起動する。このＳＬＰにおいて着パーソナル番号からLRを参照して位置情報を取り出し、SSF/CCF に転送し、伝達系での回線設定を行うことにより在圏エリアまでの接続を行う。

（3）着信接続

在圏エリアのSCP 内に配備された着側BCSMの着呼分析完DPにTDP を設定し、トリガを検出することにより着信接続ＳＬＰを起動し、着端末を呼び出し着信ユーザの認証を行った後で接続する。無線系アクセスの場合には、在圏エリアのどこにいるか分からないので、一斉呼び出しにより応答を得て、認証を行う。有線系と無線系での着信接続手順が違ってくるが、手順に応じたＳＬＰを配備することにより対応することができる。

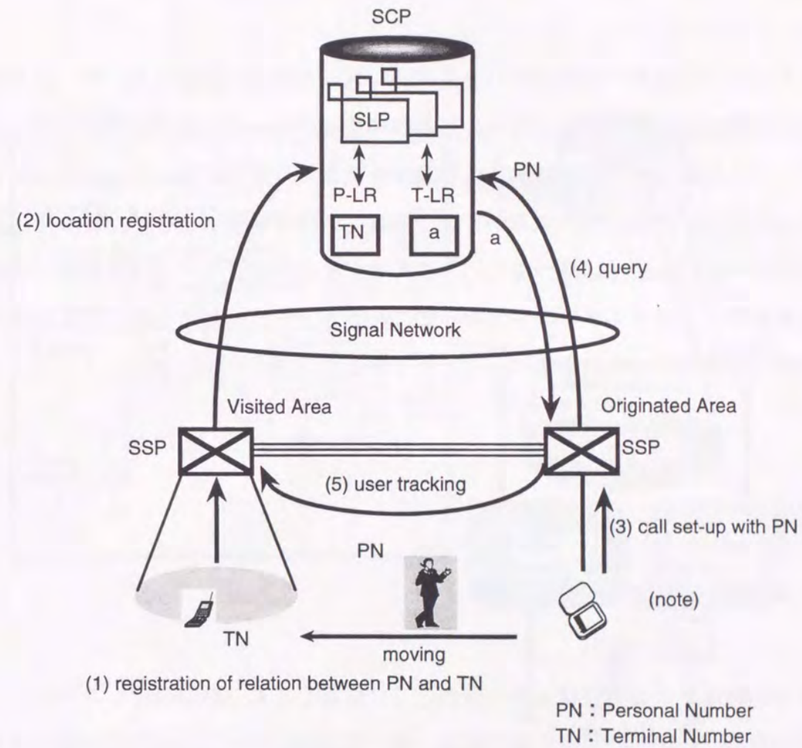


図6.5 パーソナルモビリテイ制御法

6.3.4　サービスのパーソナル化とポータビリティ

サービスのカスタマイズ化は図6.6に示すようなSLP(C)とSLP(N)が連携することにより実現される[16]。SLP(C)にはスクリーニング条件、接続条件等が定義され、SLP(N)がその条件を参照することによりサービスが提供される。SLP(C)はユーザが端末から変更することを許容している[15]。

高度ＩＮでは、交換機の呼処理プロセスを統一された形態に仮想化し、サービス制御を高機能系に配備し、多様なサービスへの対応が可能なアーキテクチャを採用している。交換機のサービス制御ポイントは基本呼状態遷移モデルより動作するcall segmentとして規格化され、このモデルはアクセス系が有線、無線に依存せず適用できる。従って、図6.6に示すように、このモデルに基づき記述されたＳＬＰよって、ユーザが有線系アクセスエリア又は無線系アクセスエリアのいづれの網に移動してもサービスを実行することができ、ユーザは個人対応のサービスを受けることができる。

6.4　考察

6.4.1　サービス展開法

パーソナル通信サービスでは、ユーザがどこに移動しても一様な通信システムとして見せる必要がある。しかし、現実の通信網は、様々な機能を持った個別の伝達系により構成されている。高度ＩＮを核としてパーソナル通信サービスを提供するために、個々の伝達系の配下にユーザが移動しても連続してサービスが受けられる機能（サービスのポータビリティ）を提供する必要がある。

ネットワークの機能拡充は段階的に行われており、機能の異なった伝達系が混在している。パーソナル通信のためのネットワーク機能の観点から分類すると、ネットワークのインテリジェント化が進めされていない伝達系（非ＩＮ伝達系）とITU-T で標準化が進められている最新のＩＮ構造を提供できる網（高度ＩＮ系）がある。

このような機能の異なった複数も伝達系が混在する過渡期においては、また将来はパーソナル通信網を提供する事業者も複数存在することを考えると、パーソナル通信網はサー

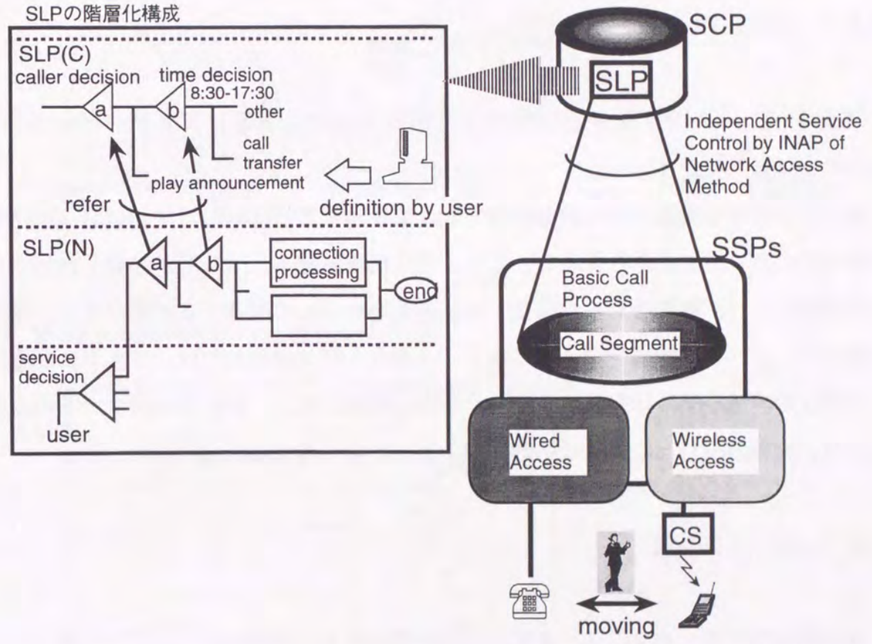


図6.6 サービスのパーソナル化とポータビリティ

ビスの連続性を保証するために、図6.7に示すような非ＩＮ伝達系制御とローミング接続機能を具備する必要がある。

非ＩＮ伝達系制御は、高度ＩＮのためのサービス制御機能を持っていない伝達系の配下にユーザが移動した場合でも、サービスを提供するための機能である。ローミング接続は、ユーザがサービス契約している通信網とは事業者の異なる通信網に移動した場合でも、継続的にサービスを提供するための機能である。

6.4.2　ＮＡＰアーキテクチャ

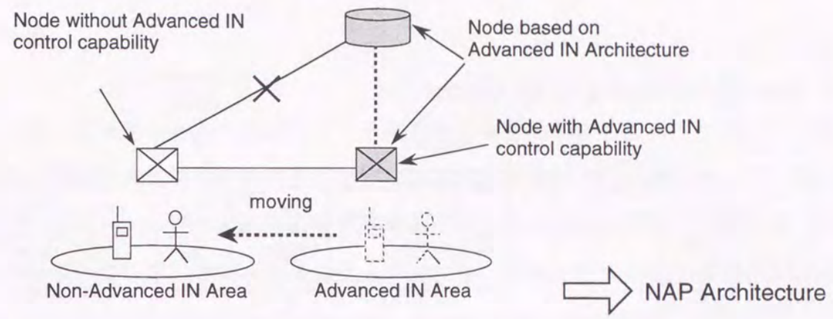
高度ＩＮアーキテクチャとは関係なく構築された伝達系はＮＡＰ(Network Access Point)と呼ばれている。

非ＩＮ伝達系を高度ＩＮの機能要素で敢えて定義するとCCF(Call Control Function)のみを持ったノードであると言える。このような伝達系配下にユーザが移動した場合のサービス制御は、SSF 機能をもった高度ＩＮ伝達系を経由して制御する必要がある。非ＩＮ伝達系のサービス制御形態を図6.8に示す。このサービス制御法では、ＮＡＰに収容されている端末の回線をSSP に引き込みＳＬＰからの制御を行う。SSP は中継ノードの位置付けとなり、発着のcall segmentは中継BCSMに沿って呼処理を進める必要がある。

6.4.3　ローミング法

複数網間でのサービスのポータビリティを実現するためのローミング方式について説明する。ローミング方式は図6.9に示すように、複数網間に渡ってサービスを継続的に提供するための手法である。ホーム網のLRをHLR(Home Location Resister) と呼び、在圏網のLRをVLR(Visitor Location Resister)と呼ぶ。端末とパーソナルモビリティを保証するためにP-/T-HLR、P-/T-VLRを定義する。VLR 方式ではユーザが移動した在圏網において、位置登録時に、ホーム網のHLR に対してパーソナルデータの転送要求を行い、コピー情報としてVLR を生成する。在圏網内では、このVLR を参照して在圏網のSCP によるサービス制御が可能となる。本方式では、着ユーザがスクリーニング等の発信系のサービスを契約している場合には発信網のSCP で制御することができる[17]。また、SLP(C)も転送することにより在圏網でのパーソナル化サービスが提供できるが、実現するためにはSLP(N)と

● Move to Non-Advanced IN Node Area



● Move to other PCS network carrier

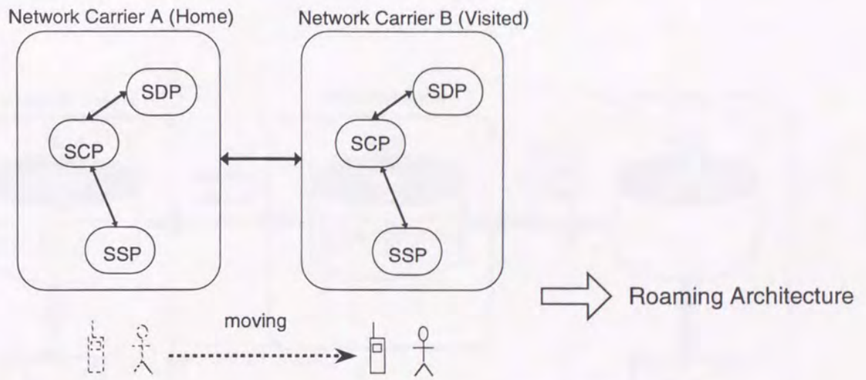


図6.7 サービス展開法

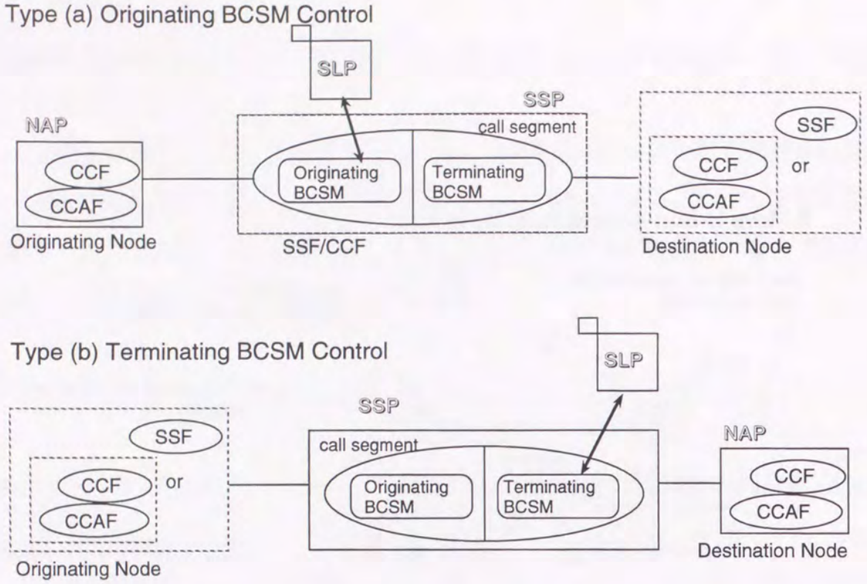


図6.8 NAPアーキテクチャによるサービス制御法

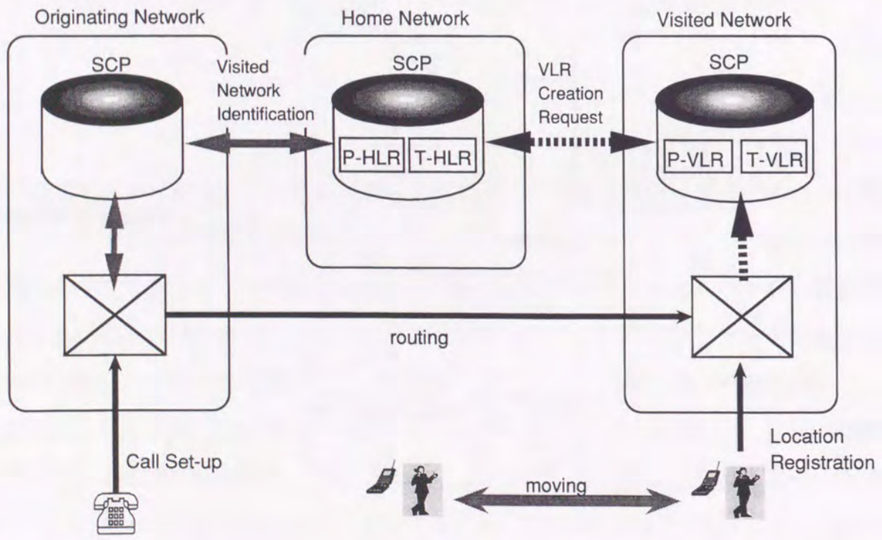


図6.9 ローミング法

SLP(C)間インタフェースの標準化やセキュリティを確保するための技術が課題となる。さらに、ローミング先の在圏網が提供できる能力とユーザが契約しているサービスを提供する上で必要となる能力間の差異を検出・制御する機能を検討する必要がある。

6.5 結言

パーソナル通信サービスを高度ＩＮアーキテクチャで提供するための手法について提案し、高度ＩＮの技術を利用することにより要求される機能がほぼ実現できることを示した。本章では、パーソナル通信のための要求条件とPCSNへの要求機能を示し、PCS 網アーキテクチャを明らかにした。本アーキテクチャでは、呼に関係するユーザインタラクションと関係しないユーザインタラクションを提供するためのSSF の機能配備法を示した。また、サービスポータビリティを提供するための有線系と無線系のアクセス法に依存しないサービス制御法及びPCS を提供するためのＳＬＰ構成法とそのカスタマイズ法を示した。さらに、既存網へのサービス展開法として、ＮＡＰアーキテクチャとローミング方法を明確化した。

第７章
知識を用いた通信サービス仕様の段階的な詳細化支援法

7.1　緒言

通信サービスの要求仕様から実際のネットワークで動作可能なソフトウェア仕様への変換が機械的手法により支援できれば、誤りの無いソフトウェアの生成が可能である。しかし、要求仕様からソフトウェア仕様を生成するためには、ネットワーク制御やプロトコル等に関する複雑で膨大な知識を必要とし、このような知識は一般にターゲットネットワークのアーキテクチャに依存する。ネットワークに依存した情報を知識として定義し、ソフトウェア仕様を生成するために利用する手法は、非エキスパートによるソフトウェア開発を支援する上で重要な課題である。

本稿では、まず通信サービスの抽象化モデルと仕様化工程の関係を示し、提案する仕様詳細化支援法の概要について説明する[1]。次に、高度ＩＮ[2]において実行されるサービスのＳＬＰソフトウェア仕様への詳細化支援技術について提案する[3]~[11]。具体的には、端末とネットワーク間のメッセージシーケンスとして定義されたサービス機能詳細仕様から実際に動作可能なＳＬＰのソフトウェア仕様への仕様詳細化知識を用いた変換法について提案する。また、仕様記述要素置き換えによる詳細化を実現するための知識として、仕様記述要素間の関係と置き換え制約を状態遷移形式により規定する方法を提案する。さらに、提案変換法をＳＬＰ生成ツールとして試作し、具体的事例へ適用した結果と評価について考察する。

7.2　仕様詳細化支援法

7.2.1　概要

通信サービスを表現するための形式的なモデルは複数の抽象化レベルより構成される。

例えば、図7.1に示すようなユーザの視点レベルとネットワークを分散プロセスによりモデル化したレベルがある。これらの２つのレベルに対応した仕様化フェーズを以下に示す。

（１）サービス要求仕様記述と検証フェーズ

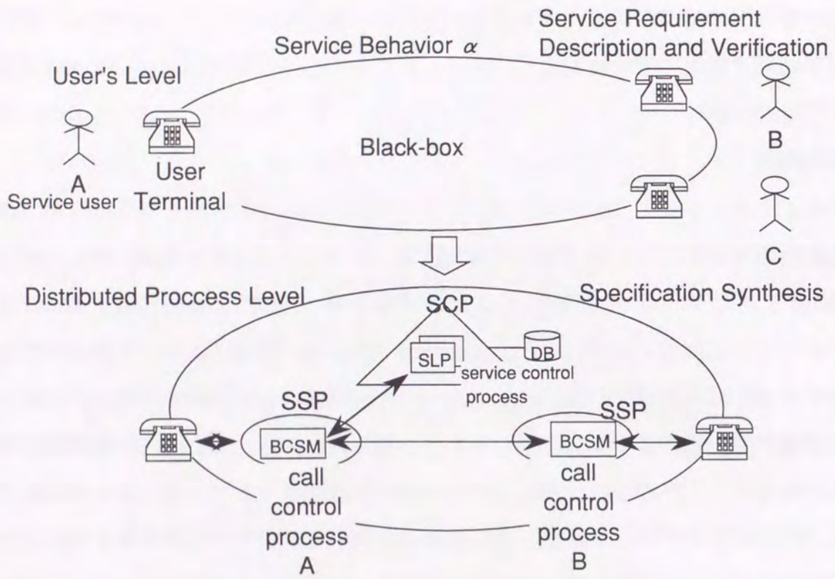
通信システムはブラックボックスとしてモデル化され、サービスは分散した端末を通してシステムにアクセスするユーザ動作として仕様化できる。サービス設計者はユーザがどのようなサービスを要求しているかを分析し、サービス要求記述に必要な情報を明確化する。このような分析の結果を形式的に記述するために、設計者はサービス要求をユーザの視点から形式的記述言語を使って記述し、新サービスと既存サービス間のサービス競合を検証する。

（２）仕様詳細化フェーズ

この抽象化レベルにおいて、通信サービスはユーザ端末からは観測できないメッセージを交換する複数のネットワークプロセスの協調によってユーザに提供される。フェーズ１において記述されたサービスを実現するためには、呼制御プロセス(Call Control Process)やサービス制御プロセス(Service Control Process) のようなネットワークプロセスが必要となり、SSP(Service Switching Point)とSCP(Service Control Point)にそれぞれ配備される[2]。このような分散プロセス間の振る舞いを規定したサービス機能仕様はサービス要求仕様から合成され、端末とネットワーク間とネットワークとノード間のプロトコルとして詳細化される。さらに、各プロセスにおいてサービスを実現するためのソフトウェア仕様はサービス機能仕様から導きだされる。

7.2.2　詳細化支援工程

提案する詳細化支援は、図7.2に示す３ステップにより構成される。ステップ１では、第２章で示した手法によりＳＴＲ言語で定義されたサービス要求仕様からネットワークアーキテクチャに依存しないサービス機能概要仕様（以下、概要仕様と呼ぶ）を生成する。概要仕様は端末とネットワーク間のメッセージシーケンスとして規定される。ステップ２では、高度ＩＮのネットワークアーキテクチャに依存したサービス機能詳細仕様（詳細仕様と呼ぶ）を定義するため、ステップ１で生成されたメッセージシーケンス図（ＭＳＣ）に網内処理記述を追加する。ステップ３では、ステップ２の詳細仕様から高度ＩＮのSCP上で動作するＳＬＰのソフトウェア仕様を仕様変換知識を利用して生成する。



(note)
SSP : Service Switching Point
SCP : Service Control Point
SLP : Service Logic processing Program

図7.1 通信サービス抽象化モデルと仕様化工程

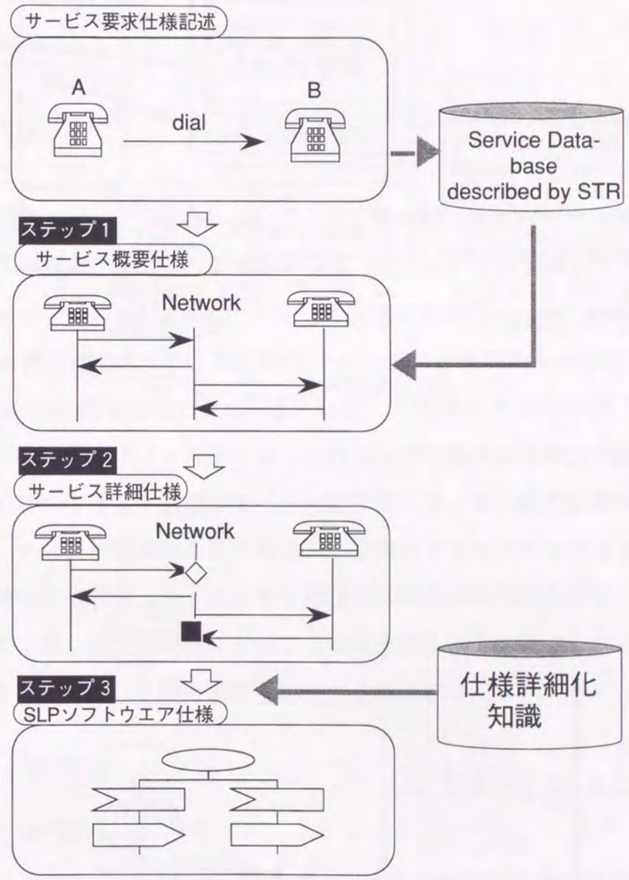


図7.2 仕様詳細化支援概要

本章では、特にステップ２の詳細仕様からステップ３のＳＬＰを生成するためのソフトウェア仕様への変換法について述べる。尚、ステップ１の変換アルゴリズムの詳細については付録7.１に示す。

7.3　仕様定義法

7.3.1　詳細仕様定義

高度ＩＮにおけるサービスは、第６章で示したようにSCPのサービスプラットフォーム上で動作するＳＬＰと呼ばれるソフトウェアを作成することにより実現される。ＳＬＰは高度ＩＮの分散機能プレーンにおける機能エンティティSCF(Service Control Function)の動作を規定するロジックであるため、ＳＬＰの記述にはSCFと他の機能エンティティ（SSF,SRF,SDF）間のプロトコル規定（INAP：IN Application Part）や、各機能エンティティの状態遷移に関する知識が要求される。従って、ＳＬＰを効率的に作成するためには、これらの複雑な知識をサービス開発者からいかに隠蔽できるかが重要である。また、詳細仕様定義はＳＬＰソフトウェア仕様生成に必要となる情報以外に、サービス開発者が仕様入力時にサービス全体の流れを容易に把握できるように、発呼から切断までの接続手順である１callスルーを定義する方式を採用している。詳細仕様は、図7.３に示すように図的に記述され、テキストは第３章で示した仕様記述言語ＭＳＤＬで記述される[12]。

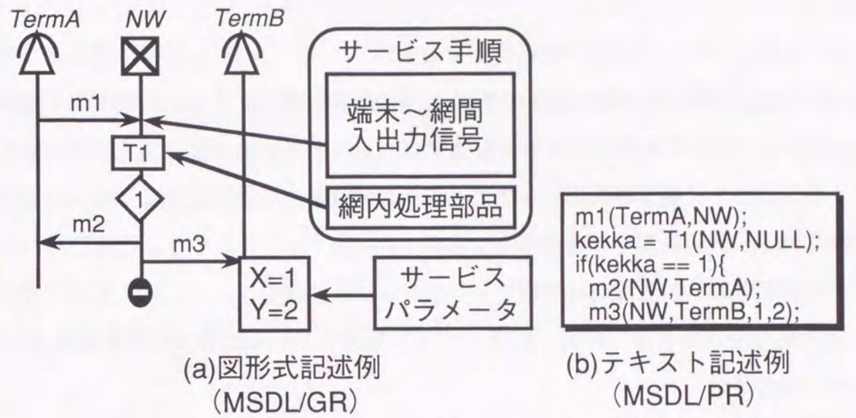
7.3.2　ＳＬＰソフトウェア仕様

サービス開発者が定義した詳細仕様から自動生成されるＳＬＰソフトウェア仕様は、ITU-Tで勧告化されている仕様記述言語ＳＤＬにより表現する（図7.４）。ＳＬＰのソフトウェア仕様SLP-SPECはSCF上で実行させるための状態遷移規定であり、図7.５に示す要素から構成され、以下のように定義する。

SLP-SPEC ＝＜Ｃ，Ｅ，Ｓ，Φ＞

ここで、

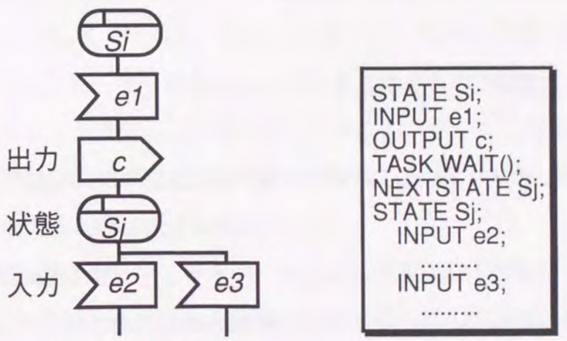
（１）Ｃ：制御命令の集合（要素をｃで表す）



(a)図形式記述例
（MSDL/GR）

(b)テキスト記述例
（MSDL/PR）

図7.3　サービス詳細仕様



(a)図形式表現
SDL/GR

(b)テキスト表現
SDL/PR

図7.4　SLPソフトウェア仕様

制御命令はＳＬＰからの出力として規定され、INAPオペレーションに対応する。INAPオペレーションは、サービスプラットフォームを経由して、SCFからSSF,SRF,SDFに対して送信され、呼接続指示、ガイダンス送出等がある。代表的なオペレーションを付録7.2に示す。

（2）Ｅ：応答イベントの集合（要素をｅで表す）

SSF,SRF,SDFがSCFからの制御命令を受信し、状態遷移規定に基づいて動作した結果、SCFへ返送れるレスポンスであり、サービスプラットフォームを介してＳＬＰに到達し、ＳＬＰへの入力として規定される。応答イベント種別として、成功応答とエラー応答の2つに分類される。代表的なものを付録7.2に示す。

（3）Ｓ：状態の集合（要素をｓで表す）

ＳＬＰが制御命令を送出した後、応答イベントを待っている状態で初期状態s_0も含む。

（4）Φ：遷移関数

ある状態s_iにおいて、制御命令c_iを送出し、応答イベント列e_iを受信すると次の状態s_jに遷移する関数である。

$$s_j = \Phi(s_i, c_i, e_i)$$

7.4　詳細仕様からソフトウェア仕様への変換

7.4.1　コンセプトと課題

詳細仕様から、ＳＬＰソフトウェア仕様への変換法は以下のコンセプトに基づいて設計する。

（1）1 callスルーの接続手順記述からＳＬＰソフトウェア仕様を生成する。

（2）ユーザの途中放棄などの準正常手順は、変換の過程で自動補完する。

（3）サービスプラットフォームの能力の追加・変更に対しても柔軟に対応する。

（4）ターゲットサービスプラットホームのプログラミング手法に柔軟に対応できる。

上記に示したコンセプトを実現するためには、以下の課題を解決する必要がある。

（課題1）コンセプト（1）を達成するために、サービス開発者が定義した詳細仕様の中から、ＳＬＰソフトウェア仕様として必要な高度ＩＮのSCP 上での実現機能を意味するＭＳＤＬ記述要素を判定し、抽出する手法を検討する必要がある。

（課題2）コンセプト（3）に対応して、図7.5に示すＳＬＰソフトウェア仕様の適切な構成要素を選択し、ＭＳＤＬの各記述要素をＳＬＰソフトウェアの部品（以下、ＳＬＰ部品）にマッピングするルールを、仕様変換アルゴリズムとは独立に仕様詳細化知識として規定する必要がある。

（課題3）コンセプト（2）を実現するため、詳細仕様において記述が不足している準正常手順を補完して、ＳＬＰソフトウェア仕様を完全化する必要がある。

（課題4）コンセプト（4）を達成するため、まずINAPに基づくサービスプラットフォームに非依存なＳＬＰソフトウェア仕様をＳＤＬで表現したミドルウェアとして生成し、その後各プラットフォームに依存したプログラミング言語に変換するアルゴリズムとする必要がある。

7.4.2　仕様詳細化知識

本節では、課題（1）～（3）を達成するための知識の構成について述べる。本章で提案する仕様詳細化は、ＳＬＰが実行すべきアーキテクチャが高度ＩＮとして既に明確化されているため、変換アルゴリズムの効率を重視して、サービス詳細仕様を定義したＭＳＤＬ記述要素で構成された変換ユニット（u）からＳＬＰソフトウェア仕様を構成する部品（制御命令（c）、応答イベント（e）、状態（s））への「置き換え（u→c，e，s）」を基本とする。従って、変換ユニットと置き換えるべき部品との関係及び置き換えるための制約（ＳＬＰが高度ＩＮのサービスプラットフォーム上で正常動作できるための制約）を状態遷移の形式で規定し、データベース化したものを「知識」と呼ぶ。これは探索による問題解決のめたの知識とは異なる。

（1）ＳＬＰ部品知識

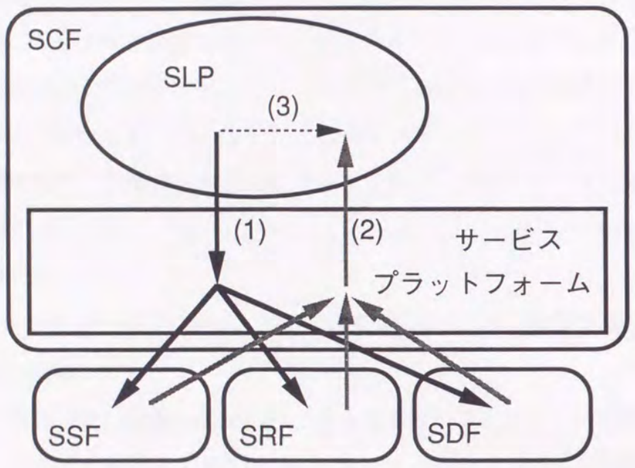
詳細仕様を記述しているＭＳＤＬ記述要素をSLP-SPECを構成する制御命令列を抽出するための知識SCK であり、以下のように定義する。

$$SCK = <U, Q, \Gamma>$$

ここで、

a）Ｕ：変換ユニット（要素をｕで表す）

詳細仕様に含まれる変換対象の意味のある単位であり、単一又は複数のＭＳＤＬ記



(1) 制御命令(output)
(2) 応答イベント(input)
(3) 状態

図7.5　SLPソフトウェア仕様の構成要素

述要素により構成される。例えば、図7.6に示すGuidanceとPB-Inputの一組のＭＳＤＬ述要素が制御命令であるPrompt And Collect User Information に変換されるべき意味のある単位として規定される。

b）Q：制御命令列の集合（要素はqで表す）

ＳＬＰは複数のINAPオペレーションを時系列的にまとめて送出できる仕様（マルチコンポーネント）が可能であるため、制御命令を連続的に記述した制御命令列を定義する必要がある。制御命令列qは送出順序を規定した制御命令の系列$c_i+\cdots+c_m$として規定する。また、制御命令列の部分集合をQ_s（$Q_s \subset Q$）と表す。

c）Γ：マッピング関数

同一の変換ユニットであってもSSF,SRF,SDF の状態に依存して、複数の制御命令列への変換が可能であるため、制御命令列の部分集合Q_sと対応させる関数が必要となる。Γは、変換ユニットu_iと制御命令列の部分集合Q_{si}との対応関係を規定した関数である。

$$Q_{si}=\Gamma(u_i)$$

ＳＬＰ部品知識SCKの例を図7.7に示す。１つの変換ユニットに対して複数の制御命令列の候補を定義した構成である。

（２）グローバル状態遷移知識

ＭＳＤＬ記述要素をＳＬＰソフトウェア仕様へと詳細化するためには、ＭＳＤＬ記述要素に対応している網機能エンティティの実行条件に応じて、ＳＬＰソフトウェア仕様を構成する制御命令および応答イベントとして適切なものを選択するための知識が必要となる。ＳＬＰが送出できる制御命令や受信できる応答イベントは、以下の３つの条件によって変化するため、これらの条件を考慮した知識を構成する必要がある。

１）制御命令発行時の機能エンティティ（SSF,SRF,SDF）の状態

例えば、ＭＳＤＬ記述要素のGuidanceは、SSFとSRFの状態に応じて異なった制御命令に変換される。もし、SSFとSRF間のコネクションが設定されていないならば、Connect To Resource & Play Announcement が適切な制御命令として選択されねばならない。一方、SSFとSRF間のコネクションが設定されていたならば、Play Announcement のほうが選択されねばならない。

２）サービス開発者が詳細仕様として定義したサービスパラメータの値

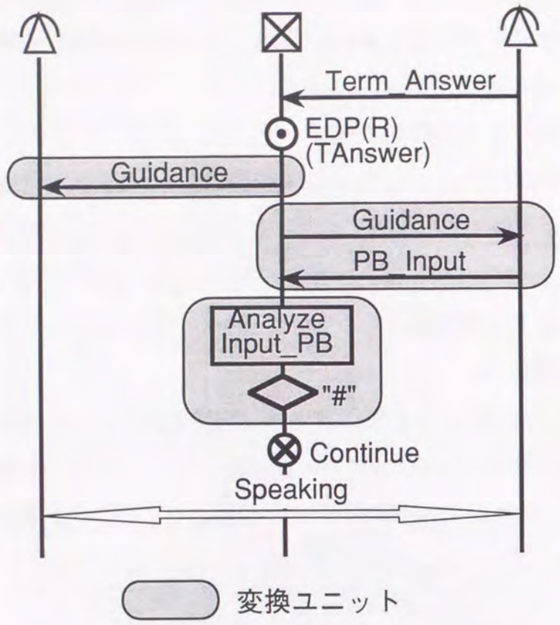


図7.6　詳細仕様における変換ユニット例

```
Unit#5 /*BusyTone*/
{
  : /*empty*/ ReleaseCall
  | DisconnectForwardConnection+ReleaseCall;
}
Unit#12 /*Guidance*/
{
  : /*empty*/ PlayAnnouncement
  | ConnectToResource+PlayAnnouncement
  | RequestReportBCSMEvent+ConnectToResource+PlayAnnouncement
  | EstablishTemporaryConnection+PlayAnnouncement
  | RequestReportBCSMEvent+EstablishTemporaryConnection+PlayAnnouncement;
}
```

図7.7　SLP部品知識構成例

例えば、ＭＳＤＬ記述要素「ガイダンス送出」に対応したPlay Announcement に対する応答として、応答イベントであるアナウンス完了通知（INAPオペレーションのSpecialized Resource Report)を受信するか否かは、サービス開発者が詳細仕様記述時に指定したパラメータ（request Annoucement Completeの通知要求有無）により変化する。「通知要求有り」にパラメータが指定された場合はアナウンス完了通知が必要となる。

3）送出した制御命令の履歴

SCFから連続的に送出される制御命令（マルチコンポーネント）に対する応答イベントは制御命令のタイプとそれらの順番により決定される。例えば、図7.8において、制御命令ＡとＢはエラー応答のみを持つものであり、Ｃはエラーと成功応答の両方を持つのであるとすると、期待されるべき応答イベントはＡ、Ｂ、Ｃのエラー応答か又はＣの成功応答である。Ｂのエラー応答はＡが正常に終了し、規定されている入力を受信した後の機能エンティティの状態が制御命令の順番に依存していることを示している。

上記の１）～３）の条件を形式化できる知識の構造としては、各機能エンティティが状態遷移モデルとして規定されているため、状態遷移型の構成をとることが適当である。この知識をグローバル状態遷移知識GSK と呼び、以下のように定義する。

$GSK = < G, Q, E, R_C, R_E, \Pi >$

ここで、

ａ）Ｇ：グローバル状態の集合（要素をｇで表し、初期状態をg_0とする）

個々の機能エンティティ（SSF,SRF,SDF）の状態を考慮して定義される。

ｂ）Ｑ：制御命令列の集合（要素をｑで表す）

ｃ）Ｅ：応答イベント集合（要素ｅで表す）

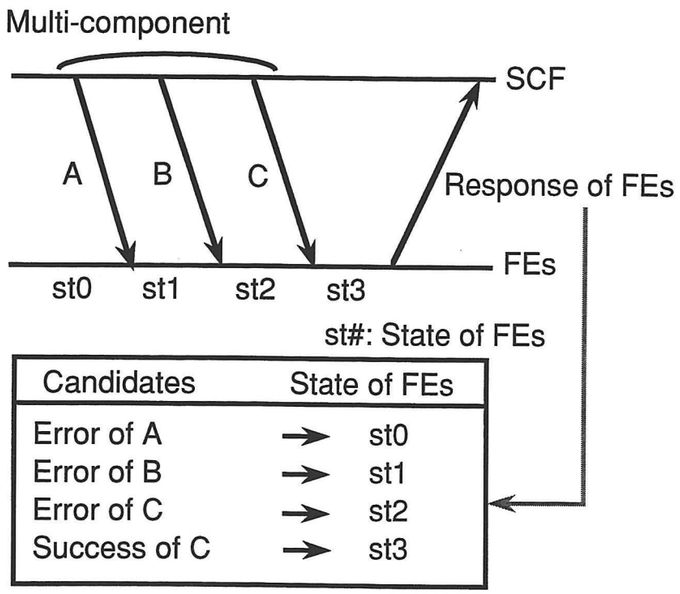
ｄ）R_C：制御命令送出制約の集合（要素をr_cで表す）

制御命令を送出するための条件を規定したものである。詳細仕様記述においてサービス開発者が定義したサービスパラメータ値による。

ｅ）R_E：応答イベント受信制約の集合（要素をr_Eで表す）

応答イベントの受信の可否に関する条件を規定したものである。詳細仕様記述においてサービス開発者が定義したサービスパラメータ値による。

ｆ）Π：状態遷移関数



（注）FE：機能エンティティ

図7.8　マルチコンポーネントの例

あるグローバル状態g_iにおいて、条件r_{ci}が一致したならば、制御命令列q_iを送出し、また、条件r_{Ei}が一致したならば、応答イベントe_iを受信し、次のグローバル状態g_jに遷移することを示す関数である。

$$g_j = \Pi\ (g_i, q_i, r_{ci}, e_i, r_{Ei})$$

グローバル状態遷移知識GSK は、SSF,SRF,SDFの状態およびサービス開発者が定義したサービスパラメータの条件下で送出可能な制御命令と、その制御命令を送出した場合の応答イベントを一意に決定できる構造を持っている。知識の一般的な図的表現を図7.9(a)に、具体例を図7.9(b)に示す。グローバル状態g_iはSCFから見たSSF,SRF,SDFの状態に対応し、状態遷移枝は以下の情報が記述されている。

a）ＳＬＰが送出する制御命令列（q）

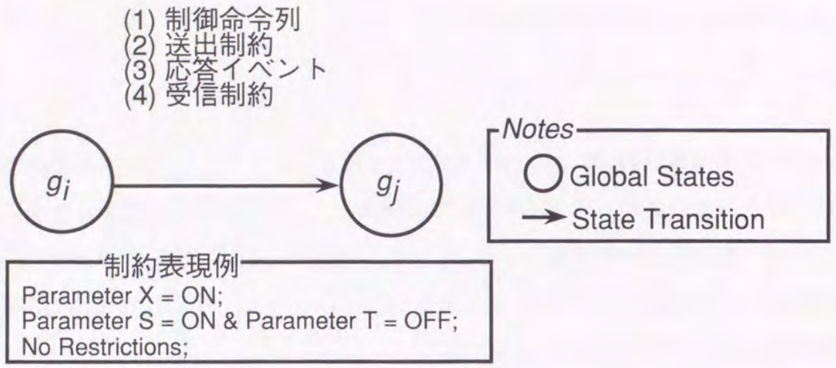
b）制御命令を限定するサービスパラメータ値（r_c）

c）ＳＬＰが受信する応答イベント（e）

d）応答イベントに基づき状態遷移を限定するサービスパラメータ値（r_E）

一方、SCFとSSF,SRF,SDFとの相互動作は処理効率の向上のため一般に非同期に行なわれため、SCFから送出した制御命令に対する応答イベントを一意に決定するためには、グローバル状態は過去に送出した制御命令の履歴（送出制御命令種別と送出順序）をSSF,SRF,SDFの状態として全て保持する必要がある。しかし、このような状態の組み合わせは膨大となり、システム構成上好ましくない。

ところが、SDFアクセスは、分岐の判定値や、SSF/SRFへの制御命令のサービスパラメータ値の取得を目的として行なわれており、次の制御命令の送出はSDFからの応答を受信した後に同期的に行なわれる。さらにSDFはSCF以外の機能エンティティとインタフェースを持たないため、SDFに送出した制御命令によりSSF/SRFの状態が変化することはなく独立した状態として規定できる。また、INAPのクラス１オペレーション（成功応答とエラー応答の両方が定義されているオペレーション）相当の制御命令や、送出後BCSMイベントをモニタするPIC系オペレーション(Analyze Information, Select Facility, Continue 等）の制御命令を送出する場合、その応答イベントを受信した後に次の制御命令が送出される（同期的に処理される）。従って、同期的に処理される制御命令と非同期的に処理される制御命令を識別し、且つ非同期に連続送出する制御命令の組み合わせ、およびその送出順序を固定することより、現実的なグローバル状態を構成することができる。



(a)一般的な図的表現法

現状態	制御命令列	応答ｲﾍﾞﾝﾄ	次状態	送信制約	受信制約
******** State 001 *********					
001	RRBE+FCI+AI	RE_RRBE	001	SEND_RRBE=ON&SEND_FCI=ON	SEND_RRBE=ON
001	RRBE+FCI+AI	RE_FCI	001	SEND_RRBE=ON&SEND_FCI=ON	SEND_FCI=ON
001	RRBE+FCI+AI	RE_AI	001	SEND_RRBE=ON&SEND_FCI=ON	NO_RESTRICTIONS
001	RRBE+FCI+AI	OAbandon	000	SEND_RRBE=ON&SEND_FCI=ON	OABANDON_REQUEST=ON
001	RRBE+FCI+AI	ODisc	000	SEND_RRBE=ON&SEND_FCI=ON	ODISC_REQUEST=ON
001	RRBE+CTR+PA	RE_RRBE	001	SEND_RRBE=ON	SEND_RRBE=ON
001	RRBE+CTR+PA	RE_CTR	001	SEND_RRBE=ON	NO_RESTRICTIONS
001	RRBE+CTR+PA	RE_PA	002	SEND_RRBE=ON	NO_RESTRICTIONS
001	RRBE+CTR+PA	SRR	002	SEND_RRBE=ON	SRR_REQUEST=ON
001	RRBE+CTR+PA	OAbandon	000	SEND_RRBE=ON	OABANDON_REQUEST=ON
001	RLC	NULL	000	NO_RESTRICTIONS	NO_RESTRICTIONS
******** State 002 *********					
002	PA	RE_PA	002	NO_RESTRICTIONS	NO_RESTRICTIONS
002	PA	SRR	002	NO_RESTRICTIONS	SRR_REQUEST
002	PA	OAbandon	002	NO_RESTRICTIONS	OABANDON_REQUEST=ON
002	DFC	NULL	001	NO_RESTRICTIONS	NO_RESTRICTIONS

(b)具体例

図7.9　グローバル状態遷移知識

7. 4. 3　準正常手順補完

詳細仕様において記述が困難な準正常手順をグローバル状態遷移知識から獲得し、補完して、ＳＬＰソフトウェア仕様を完全化する。ＳＬＰのソフトウェア仕様において発生する準正常を以下のように定める。図7.１０にそれらの例を示す。

（１）SCFからSSF, SRF, SDFへの制御命令送出後の応答イベント受信時に発生

(1-1)ユーザからの放棄、切断

(1-1-1)ユーザ入力待ち状態における放棄、切断

(1-1-2)上記以外の状態における放棄、切断

(1-2)送出した制御命令の失敗

（２）SCFにおける判定処理直後に発生

・サービスが継続できない判定結果

上記の(1-2)と(1-1-2)の準正常手順は、大半が同一手順であり記述するとシーケンスが煩雑になり、サービス開発者に意識させないことが望ましい。従って、詳細仕様からソフトウェア仕様へ仕様変換する際に、これらの準正常に対する手順を補完し仕様を完全化する。

7. 4. 4　変換アルゴリズム

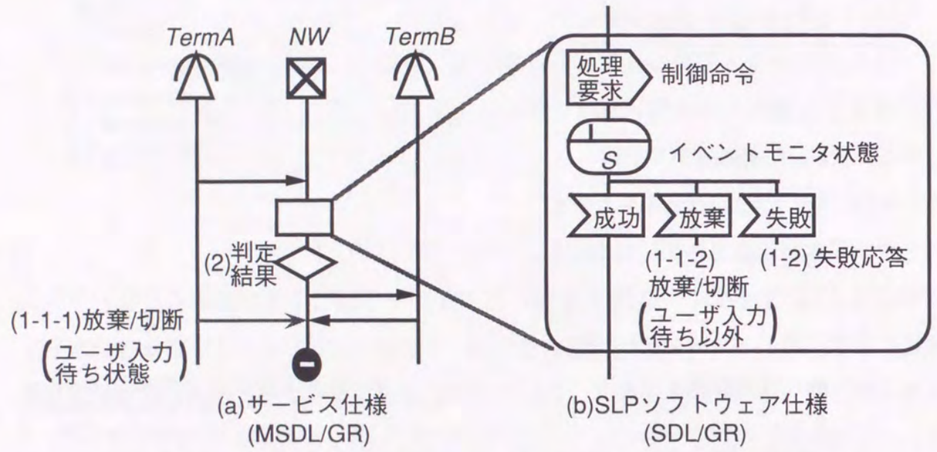
ＭＳＤＬで記述された詳細仕様は、以下に示す手順によりＳＤＬで記述されたＳＬＰソフトウェア仕様に変換される。図7.１１に処理フローを示す。

ステップ１：高機能系制御のためのＭＳＤＬ記述要素集合の抽出

高度ＩＮでは伝達系と高機能系間での制御の移行点が定義されており、TDP(Trigger Detection Point), EDP(Event Detection Point), PIC(Point In Call)と呼ばれている。これらの移行点は詳細仕様にも記述されるため、その記述要素を検出し、その記述要素間の記述要素を高機能系制御のためのＭＳＤＬ記述要素集合として抽出する。また、TDP タイプとそのモニタモードからグローバル状態遷移知識の変換開始点である初期状態 g_i を決定する。

ステップ２：制御命令列候補の収集

ＭＳＤＬ記述要素集合からＳＬＰ部品知識SCK に規定されている変換ユニットに対



(1-1-1)～(1-2)：SLPからの制御命令に対する応答イベント受信時に発生
(2)：SLPにおける判定処理の直後に発生

図7.10　SLPソフトウェア仕様における準正常と発生ポイント

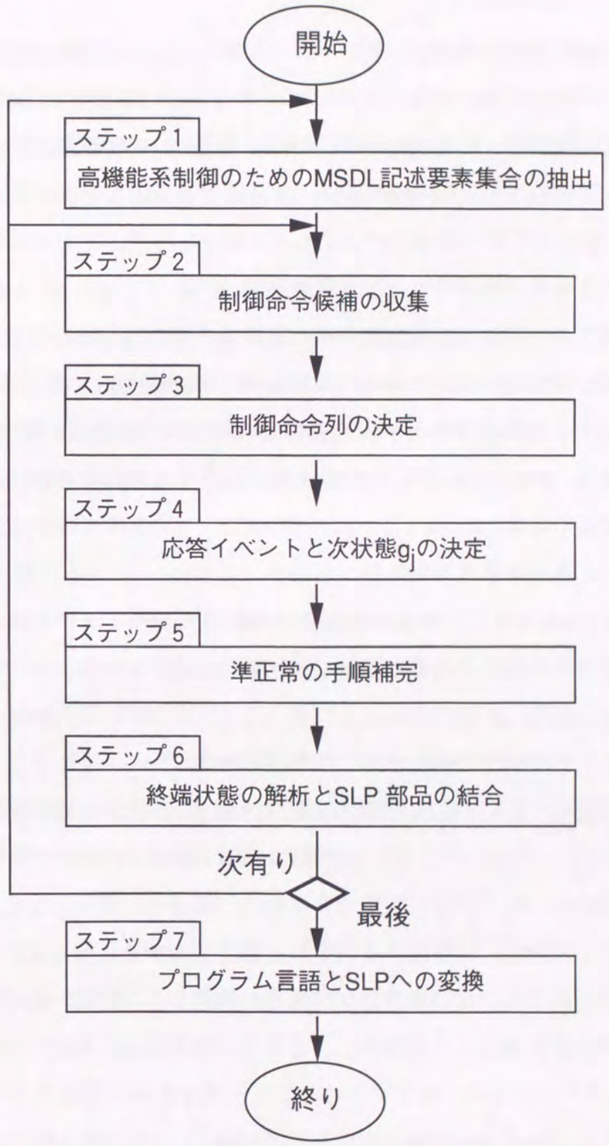


図7.11　変換処理フロー

応する記述要素u_iを特定し、ＳＬＰ部品知識SCK から制御命令列の部分集合（$Q_{si}=\Gamma(u_i)$）を獲得する。

ステップ３：制御命令列q_iの決定

グローバル状態遷移知識のステップ１で特定した状態g_iを参照し、ステップ２で抽出した制御命令列の候補Q_{si}の中から、状態g_iに状態遷移枝として結合されている枝に記述されている制御命令列q_iと合致するものを選択することにより制御命令列を決定する。

ステップ４：応答イベントe_iと次状態g_jの決定

ステップ３で決定した制御命令列に関連する状態遷移枝を検索し、その中で成功応答イベントが記述されているものについて、状態遷移枝の情報から応答イベントe_iを獲得する。遷移関数Πより状態遷移枝の終点の状態を次状態g_jとして獲得する。この状態g_jは次のＭＳＤＬ記述要素を詳細化する際にＭＳＤＬ記述要素実行時の状態として利用する。

ステップ５：準正常手順補完

状態g_iに結合され、制御命令列と合致する遷移枝でエラー応答イベントが記述されている枝を探索し、初期状態g_0への遷移に必要な制御命令列をグローバル状態遷移知識から獲得し、補完する。

ステップ６：終端状態の解析とＳＬＰ部品の結合

次状態g_jがＳＬＰを終了可能な初期状態g_0か否かを判定する。ＳＬＰが終了可能な条件は、「SCF～SSF,SRF,SDF間のrelationshipがすべて消滅している」ことである。ＳＬＰが終了可能と判定できなかった場合は、獲得したＳＬＰ部品を結合し、ステップ１で抽出した後続するＭＳＤＬ記述要素がまだ残っている場合は、ステップ２に戻り、残っていない場合は次の集合を抽出するために、ステップ１に戻る。また、詳細仕様の全てのルートを探索し、ＳＬＰを終了可能と判定した場合に、ステップ７に進む。

ステップ７：プログラム言語とＳＬＰへの変換

変換されたＳＬＰソフトウェア仕様をプログラミング言語、例えばＣ言語に変換し、ターゲットサービスプラットフォームのコンパイラを通して、データ構造の定義と変数の整合がとれたオブジェクトを生成する。

7.4.5　具体的事例への適用

仕様詳細化法を具体的事例により説明する。図7.12にＭＳＣで記述しれた詳細仕様の例を示す。まず、仕様を分析し、図7.12(a)のハッチングで示す高度ＩＮのSCP上で実行機能に関する記述を抽出する。グローバル状態遷移知識GSK の変換開始点の状態g_1はTDP情報を参照することによって決定する。次に、Guidance記述要素に対応する制御命令列の候補を図7.7の例ではＳＬＰ部品知識SCK から５つの候補が獲得される。この中で、適切な制御命令列はg_1から出ている枝の情報を探し、合致する制御命令列があれば決定される。この場合、マルチコンポーネント出力であるRRBE+CTR+PAが選択される。RRBE+CTR+PAに対する適切な成功及びエラー応答イベントがグローバル状態遷移知識から続いて獲得される。SRR,RE(RRBE),RE(CTR),RE(PA),OAbandanが選ばれ、同時に各応答イベントに対する次の状態g_1又はg_2、g_0が決定される。この場合、SRR が成功応答であり、これ以外がエラー応答である。

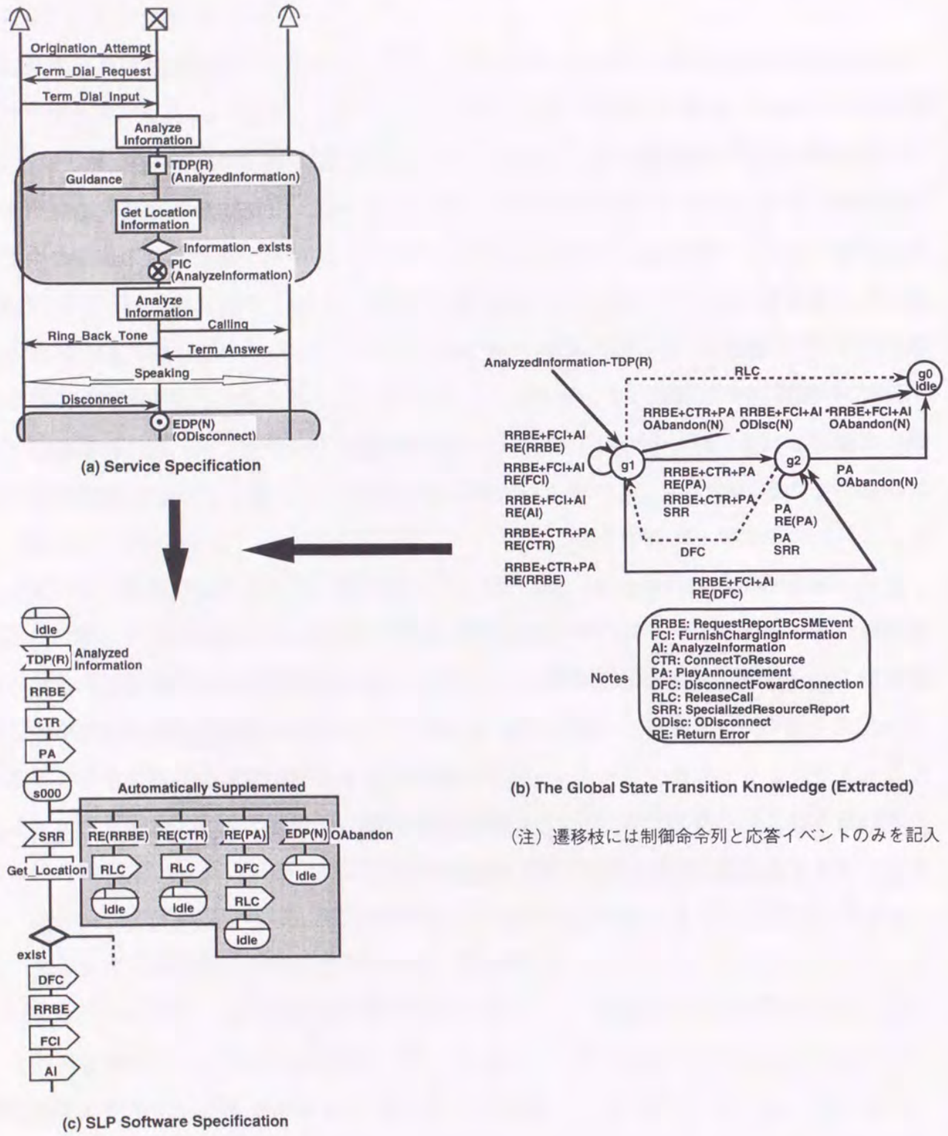
次に，補完処理が実行される。SRRを除くその他の応答イベントが挿入ポイントとして特定される。補完手順がグローバル状態遷移知識をトレースすることにより決定される。RE(PA)の場合、挿入ポイントの状態はg_2であり、g_0に遷移されるために付加される出力は図7.12(b)の破線で示すDFCとRLCであることが分かる。図7.12(c)の生成されたＳＬＰソフトウェア仕様のハッチング部分は補完された手順であることを示している。

次のＭＳＤＬ記述要素であるGet Location Informationは同じように変換され、ＳＬＰソフトウェア仕様の中のSRR入力の後に追加される。

7.5　ＳＬＰ生成ツールの試作

高度ＩＮでは、ＳＬＰをはじめとするサービスのアプリケーションソフトウェアを効率的に生成するための、総合的なサービス生成環境ＳＣＥ(Service Creation Environment)が定義されている[2]。

ＳＬＰ生成ツールは、通信サービスの接続手順を図形式で定義するための仕様入力部[12]、入力された接続手順の論理的な検証を行なう仕様検証部[13]、本稿で提案した接続手順をＳＬＰへと変換する仕様変換・ソフトウェア生成部から構成されている。このような



（注）遷移枝には制御命令列と応答イベントのみを記入

図7.12　仕様詳細化例

構成に基づくツールをワークステーション上に試作した。仕様入力部では、図7.3に示す端末－網間のメッセージシーケンスにより接続手順を定義する。シーケンスを記述するための部品は、アイコンとしてあらかじめ用意されており、サービス開発者はこれらの部品をシーケンス上に張り付けることにより容易に接続手順を定義することができる。シーケンス上で網はひとつのノードとして定義され、網内に分散して配置される機能エンティティは表現されない。本試作では、ITU-TのCapability Set 1+非呼処理関連の能力範囲までの機能を盛り込んだ。グローバル状態遷移知識の状態は約７０となった。変換アルゴリズムはＣ言語で作成した。

7.6　評価及び考察

7.6.1　記述実験と変換アルゴリズムの正当性

WS上で実現されたＳＬＰ実行環境を利用して、パーソナル通信サービスで必要となる第６章で示した４つのＳＬＰの記述実験を行い、変換したＳＬＰの動作の正常性を確認した。また、機械的変換による記述量の削減効果は表7.1に示ように、１０倍以上となった。この効果は、図7.13に示すようにＭＳＣで入力したパラメータが詳細化されたのと準正常手順が補完されたためである。

7.6.2　ＳＬＰ開発工程消化時間の削減度

ＳＬＰ生成ツール（以下、ツール）を利用することにより、ＳＬＰ開発サイクルにおける作業内容が従来の手書きによるＳＬＰ開発時と異なってくる。図7.14に示すような手書きとツール利用時のＳＬＰ開発サイクルに基づき、ＳＬＰ生成ツールのＳＬＰ開発工程消化時間削減に対する有効性について考察する。

各フェーズにおける具体的な作業内容は以下のようになる。

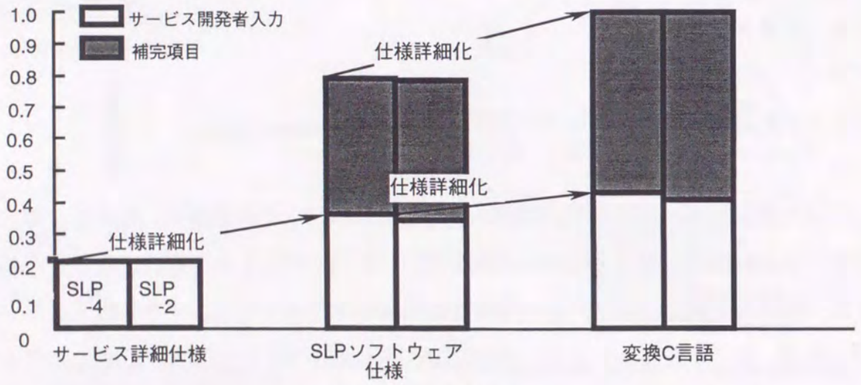
フェーズ１：サービス仕様検討

サービス仕様検討における作業は、サービスの基本的なスペック（利用する番号、課金方式等）の決定であり、手書き、ツール利用とも同様である。従って、サービス

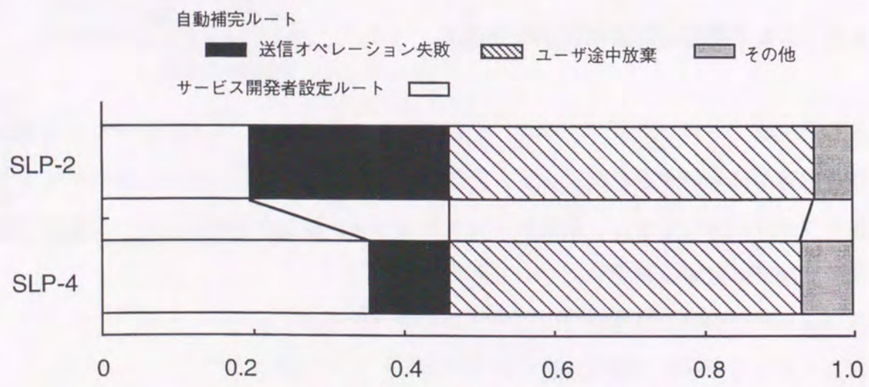
表7.1　記述実験結果

	SLP名	MSDL入力量(d1) (注1)	変換C言語(d2) (注1)	d2/d1 (注2)
SLP-1	発信認証	4	2.0	22
SLP-2	着信接続	10	3.6	18
SLP-3	着信認証	10	3.3	15
SLP-4	位置登録	1	1.0	45

（注1）SLP-4の規模で正規化
（注2）絶対値で比較



(a) 項目設定率



(b)変換C言語における準正常手順内訳

図7.13　仕様補完の内訳

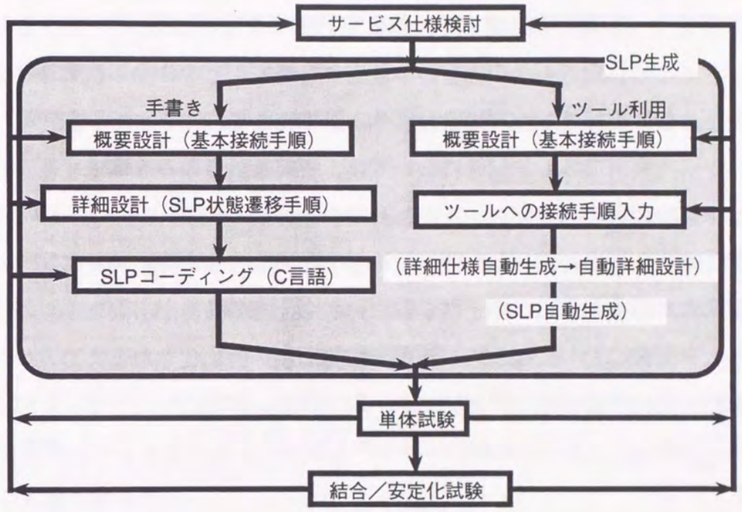


図7.14　SLP開発工程

仕様検討フェーズの工程消化時間に対して、以下の式が成り立つ。

Svt=Svm

Svt(ツール利用による本工程消化時間)

Svm(手書きによる本工程消化時間)

フェーズ2：ソフトウェア仕様設計

手書きによるＳＬＰ開発においては、ＳＬＰ製造に向けてサービス仕様を段階的に詳細化する作業が必要となる。例えば、接続手順の概要（正常ルート）およびパラメータに設定すべき値などを記述した概要設計書をまず作成し、次にSCFと他の機能エンティティとのINAPベースの信号のやりとりを考慮したＳＬＰの状態遷移図（準正常ルートを含む）を記述した詳細設計書へと詳細化する。

一方、ツール利用のＳＬＰ開発においては、概要設計書のみを作成する。詳細設計書は、仕様変換過程においてツールが自動的に出力する。

概要設計書の仕様を詳細化し、詳細設計書を作成する際には、INAPに関する高度な知識が要求されるとともに、準正常手順に対する仕様の検討が必要とされる。ツール利用のＳＬＰ開発においては、この仕様詳細化過程が自動化されるため、ソフトウェア仕様設計時間の大幅な削減が期待でき、以下のように定式化できる。

Aft=Afm

Dft=0

Dfm= {1/(1-Qn)} ×(1/Ip)×Afm

Sft=Aft＋Dft=Afm

Sfm=Afm+Dfm= {1+1/(1-Qn)×(1/Ip)} ×Afm

Afm（手書きによる概要設計書作成時間）

Aft（ツール利用による概要設計書作成時間）

Dfm（手書きによる詳細設計書作成時間）

Dft（ツール利用による詳細設計書作成時間）

Sfm（手書きによる本工程消化時間）

Sft（ツール利用による本工程消化時間）

Ip(サービス開発者のINAPの習熟度、0(低) ＜Ip≦1(高))

Qn(ソフトウェア仕様内の準正常手順の割合、0＜Qn＜1)

フェーズ3：製造

手書きによるＳＬＰ開発においては、詳細設計書をもとにＣ言語により直接コーディングを行なう。一方、ツール利用のＳＬＰ開発においては、概要設計書をもとにツール上で接続手順の入力を行ない、ツールの仕様変換・ソフトウェア生成機能によりＣ言語ソースを生成する。

ツール利用によるＳＬＰ開発においては、以下の項目に関して製造工程消化時間の削減が期待できる。

・ツールが提供するGUIを利用することによる記述効率の向上（プログラミング言語に関する詳細な知識が不要）

・記述量自体の削減（準正常手順に対する記述が不要）

フェーズ4：試験

試験工程における作業は、製造したＳＬＰに対する試験項目を策定し、ノード（モジュール）内の単体試験、および複数ノード（モジュール）間の結合、安定化試験を行なうものであり、作業内容自体は手書き、ツール利用とも同様である。

しかし、ツールが提供する知識に基づいて自動生成されるソースコードは、仕様詳細化およびソフトウェア生成過程におけるバグの混入率が理論上ゼロであるため、試験工程において検出されるバグは、サービス仕様誤りおよび概要設計ミスに起因するもののみである。

製造と試験工程における消化時間は、以下のように式として表現できる。

Mt=In×Mm

Tt=Se×Tm

Mm（手書きによる製造工程消化時間）

Mt（ツール利用による製造工程消化時間）

Tm（手書きによる試験工程消化時間）

Tt（ツール利用による試験工程消化時間）

In（ツール入力時間のＣソースコーディング時間に対する比率）

Se（サービス仕様に依存するバグの割合、 0≦Se≦1)

以上の各工程を総合したＳＬＰ生成ツール利用による開発工程消化時間の削減度ρは、以下の式で表現される。但し、サービス仕様設計、ソフトウェア仕様設計、製造、試験の

各開発フェーズの工程消化時間の全開発サイクルの工程消化時間に占める比率をそれぞれａ、ｂ、ｃ、ｄとする。

$$\rho = 1-(Svt+Sft+Mt+Tt)/(Svm+Sfm+Mm+Tm)$$
$$= 1-[a+b/\{1+1/(1-Qn)\times(1/Ip)\}+c\times In+d\times Se]$$

一般的に、各開発フェーズの工程消化時間の全開発サイクルの工程消化時間に占める比率は、システム特性、規模、開発条件、開発手法等により異なるが、現状の技術レベルでは、a+b=0.35, c=0.2, d=0.45 [11] 程度と言われている。また、過去の通信サービスソフトウェア開発から得られる経験値によれば、Se=0.1, In=0.2 ～0.5 [12] とでき、準正常手順の全サービス仕様に占める割合Qnは、既存ISDNの接続手順（中断、再開手順を除く）[13] における値0.49と7.6.1節に示した記述実験から0.4 ～0.6 程度と想定する。さらに、本ツールが対象とするサービス開発者は、INAPに関する詳細な知識を持たないサービス開発者であることから、Ip=0.1～0.5 とし、a/b=0.5 ～1.0 程度で振らせるとρは0.62～0.79となり、サービス開発サイクル全体で60%～80%の工程消化時間を削減可能であると見積もることができる。但し、本効果は手書きによる開発工程消化時間との比較を実施し、確認する必要がある。

7.6.3　生成されるＳＬＰの品質

ＳＬＰ生成ツールの仕様変換部では、INAPオペレーション送信に関するマルチコンポーネント化等の最適化を行なっている。またソフトウェア生成部においても、サブルーチン関数のインライン展開や、インライン展開時のサブルーチン関数内の不要な分岐の削除処理等により、ダイナミックステップの削減を図っており、手書きで作成されたＳＬＰと同等の処理効率が得られる。

7.6.4　高度ＩＮの能力追加・変更に対する柔軟性

ＳＬＰ生成ツールの仕様変換とソフトウェア生成部では、INAPオペレーションの追加・変更に柔軟に対応するため、INAPに基づくSCFの状態遷移を仕様詳細化知識として変換・生成論理から切り出している。従って、INAPオペレーションの追加・変更は、仕様詳細化知識の追加・変更を行なうだけで対処可能である。仕様詳細化知識は状態と状態遷移のトリガとなるイベントからなるテーブル構造をとっており、内容の修正および修正後のデッドロック等の検証が比較的容易である。SCFの状態遷移に関する知識を有したサービス開発者が追加・変更を実施すると仮定すれば、ＳＬＰの開発時間に比べてごく短い期間での対処が可能である。

7.7　結言

本章では、端末とネットワーク間のＭＳＣによる詳細仕様から高度ＩＮのＳＬＰソフトウェア仕様に詳細化するための手法について提案し、仕様詳細化知識の種別と構成、仕様補完法及び変換アルゴリズムを明らかにした。また、提案手法を試作し、具体的な事例による評価により、ソフトウェアの仕様記述量を１０分の１以下に削減できたことを示した。また従来と本ツールを用いた開発工程のモデルを定式化して比較し、60%～80%の工程消化時間削減の可能性を示した。

本詳細化法の目的に対して、以下の成果を得ることができた。すなわち、ITU-Tの高度ＩＮのCS１＋非呼処理関連の能力範囲内のモデルにおいて、これまで報告の無かったＭＳＣで記述した詳細仕様から知識を用いてＳＬＰソフトウェア仕様を生成する技術を確立することができた。

第８章　結　論

本論文では、これまでの高い技術力を持った専門家に依存した通信サービスソフトウェア開発からの脱却を図るため、非専門家の参画を狙った通信サービスの要求仕様化を支援する技術について研究した。具体的には、(a)仕様入力、(b)仕様統合化、(c)仕様完全化、(d)仕様詳細化の各課題について研究し、(1)視覚的な仕様入力と確認技術、(2)サービス競合検証技術、(3)手続き型から宣言型への仕様変換技術、(4)サービス付随動作の補完技術、(5)サービス仕様を詳細化するための高度ＩＮに基づく網機能配備法、(6)高度ＩＮで実行可能なサービスソフトウェア仕様への詳細化技術、を明らかにした。また、提案手法を試作し、有効性を評価・考察した。本研究の主な成果は以下の通りである。

（１）通信システムの内部動作を考慮しなくても、端末の状態とイベントだけにより、サービス要求を仕様化できる視覚的な支援手法を提案した。プロトタイプを試作し、28種類の付加サービスの記述実験により、図形による仕様記述とアニメーションによる仕様確認の有効性を評価した。図形による仕様記述では、端末とその関係に基づく直感的に仕様が記述できることを確認した。さらに、28種類の付加サービスをサービス動作の観点から７つの代表的な種別に分類し、その網羅性と図形シンボルの汎用性及び仕様記述と端末数との関係を考察した。アニメーションによる仕様確認では、絵シンボルの汎用性を実験により確認すると共にサービス仕様の逐次実行による仕様の理解性向上と意味的な仕様矛盾に対する検証支援としても有効であることを示した。また、状態遷移の非決定性に起因するサービス競合の解消アルゴリズムの正当性を記述実験により示し、その解消結果の図形表示による有効性を示した。

（２）通信サービス開発するための、重要な課題であるサービス競合検出を支援するための方法について提案した。具体的には、状態遷移モデルに基づいて記述された通信サービス仕様を対象として、２つの単独サービス仕様を合成した時に発生する意味的矛盾の機械的な検出法について示した。意味的矛盾として、「異常な状態への遷移」の定義を示した。また、異常な状態への遷移を機械的に検出するために必要となる仕様記述言語への要求条件を示した。その記述言語に基づき記述された仕様を合成した場合の異常な状態への遷移

を検出するアルゴリズムについて提案した。ＳＴＲ言語により記述した３つの代表的なサービスを組み合わせた場合の机上での検証実験を行い、検出されるべきサービス競合は全て検出できたことを示した。また、その被検証状態の数も絞り込まれており、人手による検出と比べ１０分の１に削減されたことを確認した。さらに、知識情報処理の分野で研究されているルール集合の管理検証に関して、非専門家に対する利点と矛盾検出法の観点からの考察を示した。

（３）ＭＳＣで記述された手続き型の通信サービス仕様をルール集合による宣言型の記述仕様に変換する手法を提案した。メッセージと端末の状態及びイベントの対応関係を分析し、ＭＳＣ記述要素と状態変換要素及びイベント変換要素との関係を明確化した。さらに、ＭＳＣによる記述仕様からルールの集合に基づく記述仕様への変換アルゴリズムを示した。具体的な適用事例として、ＭＳＤＬで記述した４つのサービス仕様をＳＴＲ言語で記述した仕様への変換実験を机上で行い、これまで、ＭＳＣで記述された仕様に対して報告のなかったシミュレーションによる仕様確認及びサービス仕様競合検証への適用可能性を示した。宣言型に変換された仕様では、一行が１つの状態遷移に関する断片的な要求として記述され、且つ記述を再利用できるため、手続き型の記述仕様に比べ記述量も少なくなり、仕様確認が容易であることを示した。また、手続き型の記述仕様では不得意とする状態遷移の非決定性の検出に関する机上実験を行い、有効性を示した。

（４）通信サービスを分散プロセスモデルとしてＭＳＣで仕様化するための支援システムを提案した。システムは設計知識としてＭＳＲを持ち、それにより支援することを特徴とする。ＭＳＲの獲得法として、メッセージの抽象化とメッセージの入出力間の関係のモデル化による手法を示し、ＭＳＣでサービス動作を記述するためのルール適用法を示した。ＭＳＲを用いて、設計者がユーザと関連プロセス間の可能なインタラクションを完全化する支援法として、２ステップからなる方法を提案した。ステップ１では、ＭＳＣによる代表的なサービス動作の記述支援のために候補メッセージを提案し、ステップ２では、付随的なサービス動作として仕様化するために可能なメッセージ系列を推論し、ＭＳＣとして生成する。提案手法を試作し、ＭＳＲの再利用性と支援法の評価を行った。３つの具体的なサービスに適用した記述実験では、再利用性の高さを示し、ＭＳＲの獲得法の有効性を示した。また、基本電話サービスの仕様化実験では、設計者が全ての状態での可能なメッ

セージを検討することから開放されることを示し、設計者は支援システムからのメッセージ候補の判断と具体的メッセージ名の付与だけで、通信サービスを仕様化できることを確認した。

（5）パーソナル通信サービスを高度ＩＮで提供するための機能配備法について提案し、高度ＩＮの技術を利用することにより要求される機能がほぼ実現できることを示した。具体的には、パーソナル通信のための要求条件とネットワークへの要求機能を示し、呼に関係するユーザインタラクションと関係しないユーザインタラクションを提供するためのSSF の機能配備法、サービスポータビリティを提供するための有線系と無線系のアクセス法に依存しないサービス制御法、パーソナル通信サービスを提供するためのＳＬＰ構成法とそのカスタマイズ法を示した。さらに、サービスの展開法について考察し、高度ＩＮ構造を持たない既存網への展開法としてＮＡＰアーキテクチャを示し、また他のパーソナル通信網との相互乗り入れを可能とし、網間に跨がるポータビリティを提供するためのローミング方法を示した。

（6）端末とネットワーク間のＭＳＣにより記述された詳細仕様から高度ＩＮのＳＬＰソフトウェア仕様に詳細化するための手法について提案し、仕様詳細化知識の種別と構成法、準正常手順の仕様補完法及び変換アルゴリズムを明らかにした。さらに、提案手法を試作し、ITU-Tの高度ＩＮのCS１＋非呼処理関連の能力範囲内のモデルにおいて、これまで報告の無かったＭＳＣで記述した詳細仕様から知識を用いてＳＬＰソフトウェア仕様を生成する技術を確立することができた。また、具体的な事例としてパーソナル通信サービスへの適用実験により、ソフトウェアの仕様記述量を１０分の１以下に削減できたことを示した。また、従来手法による開発工程と提案手法を組み込んだＳＬＰ生成ツールを用いた開発工程をそれぞれモデル化して比較し、60%～80%の工程消化時間削減の可能性を示した。

本論文では、ユーザニーズの多様化する通信サービスの迅速な開発を目指すために、非専門家による要求仕様の記述を可能とするための支援技術に関する研究成果を示した。通信サービスソフトウェアの生産性を向上させるためには、要求を正確に定義するための上流工程が重要となる。通信サービスがソフトウェアを実装したノードからなるネットワークにより提供される限り、要求記述の曖昧さを補い、検証を正確に行い、ソフトウェア生

成を自動化するための一連の技術が望まれる。将来は、ユーザ自身が独自の通信サービスを自然言語と視聴覚的支援を利用した高度なＨＭＩにより定義し、従来のような複雑な開発工程に気付かないでソフトウェアを生成し、利用するようなネットワーク環境が構築されると思われる。本研究が通信サービスソフトウェアの開発方式の進展の一助となれば幸いである。

参考文献

第１章

（１）Wakahara,Y. and Ito,A. :“A Prototyping System for Telecommunications Software Based on Abstract Execution of Requirements Specifications,” Computer and Networks and ISDN Systems Vol.13, No.2, pp119-128 (1987).

（２）New,D. and Amer,P. :“Adding Graphics and Animation to Estelle,” Information and Software Technology, Vol.32, No.2, pp149-161,(1990).

（３）李、森、白鳥、野口：“G-LOTOS の仕様化環境SEGLの構成と試作”、情報処理学会論文誌、Vol.32,No.3,(1991).

（４）高見、太田：“通信サービスの視覚的な要求仕様化支援法”，情報処理学会論文誌 Vol.36,No.3 pp.728-741,Mar.1995.

（５）高見：“非専門家のための通信サービス仕様記述法”、第１回通信ソフトウェア工学研究会（信学会第２種研究会）、(1993-07)

（６）市川、高見：“通信ソフトウェアの自動生成技術”，電子情報通信学会誌 Vol.77, No.5 ,pp.522-530,May 1994.

（７）高見：“プログラム自動合成に関する研究－通信サービスソフトウェアの自動生成－”、情報処理学会関西支部ソフトウェア研究会・講演会、(1995-06)

（８）Rubin,R.V., Walker II,J.and Golin,E.J. :“Early Experience with the Visual Programmer's WorkBench,” IEEE Trans. on Software Engineering, Vol.16, No.10,(1990).

（９）Wakahra, Y., Fujioka, M., Kikuta, H., Yagi, H., and Sakai, S.:“A Method for Detecting Service Interactions,” IEEE Communication Magagine, Vol.31 No.8, 1993.

(10) R. Boumezbeur, L. Logrippo :“Specifying Telephone Systems in LOTOS,” IEEE Communication Magagine, Vol.31 No.8, 1993.

(11) Harada, Y., Hirakawa, Y., Takenaka, T. :“A Design Support Method for Telecommunication Service Interactions,” GLOBECOM'91, Phoenix, Arizona, Dec. 1991.

(12) 松本、佐藤：“非単調論理と常識推論，”情報処理学会誌，Vol.30 No.6, June 1989.

(13) 人工知能学会：人工知能ハンドブック，1990.

(14) 原田、高見、太田、寺島：“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，”情報処理学会論文誌 Vol.35 No.8, Aug. 1994.

(15) 水野、新津：“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”、電子情報通信学会誌論文誌、Vol.J74-B-I,No.12,pp1042-1055,(1991).

(16) Niitsu,Y. and Mizuno,O. :“Computer-Aided Stepwise Service Creation for The Intelligent Network,” IEICE Trans. Commun. Vol.E75-B,No.10 Oct. 1992.

(17) 特集：通信システムの形式記述技法の標準化、情報処理学会誌、Vol.31 No.1 pp.2-81、(1990-01)

(18) Hirakawa,Y and Takenaka,T:“Telecommunication Service Description Using Sate Transition Rules,” Sixth Int. Workshop on Software Specification and Design, Como, Italy,(1991).

(19) Okamoto,M. and Niitsu,Y. :“A Verification Scheme for Service Specifications Described by Information Sequence Charts,” IEICE Trans. Commun. Vol.E75-B,No.10 Oct. 1992.

(20) Harada,Y., Hirakawa,Y., Takenaka,T. and Terashima,N.:“A Conflict Detection Support Method for Telecommunication Service Descriptions,” IEICE Trans. Commun., Vol.E75-B,No.10,(1992).

(21) Inoue,Y., Takami,K. and Ohta, T. :“Automatic Detection of Service Interactions in Telecommunications Service Specifications,” IEEE International Conference on Communications (ICC'94),May 1994.

(22) Ohta,T., Takami,K. and Takura,A. :“Acquisition of Service Specification in two-stage Detection / Resolution of Feature Interaction,” The 4th Telecommunications Information Networking Architecture Workshop (TINA'93), (1993).

(23) Takami,K., Harada,Y., Ohta,T., and Terashima,N. :“A Visual Design Support System for Telecommunication Services,” IEEE Phoenix Conference on Computers and Communications (IPCCC'93), (1993) .

(24) Zave, P. : "A Distributed Alternative to Finite-State-Machine Specifications," ACM Trans. on Programming Language and Systems, Vol.7, No.1, January, (1985).

(25) Wakahara, Y., Kakuda, Y., Ito, A. and Utsunomiya, E. : "Escort: An Environment for Specifying Communication Requirements," IEEE Software, 3, pp38-43 (1989).

(26) Shiratori, N., Takahashi, K., Sugawara, K. and Kinoshita, T. : "Using Artificial Intelligence in Communication System Design," IEEE Software, 1, pp38-46 (1992).

(27) Nakahara, A., Aida, H. and Saito, T. : "A Method for Constructing FSM model from Event-Sequence Based FDT," IEICE Trans., Vol.J72-B-I, No.12, December, (1989).

(28) R. A. Orr, M. T. Norris, R. Tinker and C. D. V. Rouch:"Tools for Real Time System Design," Proc. 10th ICSE, April, (1988).

(29) 服部： "パーソナル通信の動向と展開",電子情報通信学会誌 Vol.78 No.5 pp.506-509 (May 1995).

(30) 中島： "パーソナル移動通信のためのインテリジェントネットワーク"、電子情報通信学会誌 Vol.78 No.2 pp.143-149 (Feb. 1995).

(31) 鈴木： "高度インテリジェントネットワーク"、電子情報通信学会誌 Vol.77 No.4 pp.410-415 (Apr. 1994).

(32) 江崎、大宮、重松： "インテリジェントネットワークにおける伝達網の仮想化とその制御法"、電子情報通信学会論文誌 Vol.J74-B-I No.11 pp.949-958 (Nov. 1991).

第2章

(1) 高見、太田： "通信サービスの視覚的な要求仕様化支援法"， 情報処理学会論文誌 Vol.36,No.3 pp.728-741,Mar.1995.

(2) K. Takami, Y. Harada, T. Ohta, N. Terashima : "A Visual Design Support System for Telecommunicaton Services", IEEE Phoenix Conference Computers and Communications(IPCCC'93) pp.593-599, March 1993.

(3) K. Takami, T. Ohta, N. Terashima : "A Visual Programming for Telecommunication Services", IEEE/CS Symposium on Visaul Languages(VL'93) pp.360-361, Aug. 1993.

(4) H. Enoki, K. Takami, Y. Kobayashi, T. Ohta:"Swiching Function Conceptual Model for Telecommunication Service Specification Design", IEEE Global Telecommunication Conference(Globecom'94) pp.818-822, Nov. 1994.

(5) M. Nakanura, Y. Inoue, K. Takami, T. Takenaka, T. Ohta : "Telecommunication Service Design Support Method Using Visual Service Description Language STR-GR",IEEE Global Telecommunication Conference(Globecom'92) pp.554-559, Dec. 1992.

(6) 高見、河原崎： "通信サービス仕様記述と視覚的プログラミング"， 電子情報通信学会誌 Vol.78, No.12,pp.1260-1267,Dec. 1995.

(7) 高見、太田： "状態遷移規則による通信サービス記述性評価"、信学春季全大 B-573、 (1993)

(8) 中村、高見、竹中： "通信サービス検索手法について"、信学春季全大 SB-5-4、(1992)

(9) 榎木、高見、太田： "仕様記述要素定義のための通信サービス概念モデルについて"、信学秋季全大 B-464 、 (1993)

(10) 和田、高見、太田： "通信サービスの日本語による仕様記述支援法の一考察"、信学秋季全大 B-463 、 (1993)

(11) 高見、平川、河田、竹中： "ビジュアルプロトタイピングによる通信サービス仕様生成法"、信学技報 SSE91-103、(1991-11)

(12) 高見、原田、太田： "通信サービス記述仕様の確認支援法"、信学技報 SSE92-86、(1992-11)

(13) 井上、高見、竹中： "仕様部品と規則による通信サービスのビジュアルプロトタイピングシステム"、信学技報 IN91-87 、(1991-09)

(14) 中村、高見、竹中： "ビジュアルなサービス記述STR-GRによる通信サービス設計支援法"、信学技報 SSE91-179、(1992-03)

(15) 榎木、高見、太田： "通信サービスの概念モデルの一考察"、信学技報 SSE93-77、

(1993-10)

(16) 和田、高見、太田：“通信サービス仕様記述支援におけるガイダンス手法の一考察”、信学技報 SSE93-60 、(1993-09)

(17) 榎木、高見、小林、太田：“通信サービス概念モデルの応用に関する一考察”、信学技報 SSE93-126、(1993-12)

(18) 和田、高見、太田：“非専門家を対象としたサービス仕様記述支援法”、信学技報 SSE93-138、(1994-01)

(19) Harada,Y, Hirakawa,Y, Takenaka,T and Terashima,N.:“A Conflict Detection Support Method for Telecommunication Service Descriptions,” IEICE Trans. Commun., Vol.E75-B,No.10,(1992).

(20) Hirakawa,Y and Takenaka,T:“Telecommunication Service Description Using Sate Transition Rules,” Sixth Int. Workshop on Software Specification and Design, Como, Italy,(1991).

(21) Bellcore:Feature Common to Residence and Business Customers II ～ III,LSSGR, Technical Reference TR-TSY-000521~000522,Issue 2,(1987).

(22) CCITT:Integrated Services Digital Network (ISDN) General Structure and Service Capabilities, Recommendation I.110~ I.257,(1988).

第3章

(1) E. J. Cameron, N. Griffeth, Y. Lin, M. E. Nilson, W. K. Schnure and H. Velthuijsen : “A Feature-Interaction Benchmark for IN and Beyond,” IEEE Communication Magazine, Vol.31 No.3, pp. 64-69, 1993.

(2) E. J. Cameron, H. Velthuijsen:“Feature Interactions in Telecommunications Systems,” IEEE Communication Magagine, Vol.31 No.8, 1993.

(3) 高見、井上、太田：“サービス競合における異常な状態への遷移の検出法”，情報処理学会論文誌 Vol.36,No.1,pp.93-104,Jan.1995.

(4) 原田、高見、太田、寺島：“通信サービス課金仕様競合検出方式”，情報処理学会論文誌 Vol.34 No.5 pp.1064-1073, May 1993.

(5) 原田、高見、太田、寺島：“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”、情報処理学会論文誌 Vol.35 No.8 pp.1602-1613, Aug.1994.

(6) Y. Inoue, K. Takami, T. Ohta : “Method for Supporting Detection and Elimination of Feature Interaction in a Telecommunication System”, IEEE International Workshop on Feature Interactions In Telecommunications Systems pp.61-81, Dec. 1992.

(7) T. Ohta, K. Takami, A. Takura:“Acquisition of Service Specification in Two-stage Detection / Resolusion of Feature Interaction”, Telecommunications Information Networking Architecture Workshop(TINA'93) pp.Ⅱ-173-187, Sept. 1993.

(8) Y. Inoue, K. Takami, T. Ohta : “Automatic Detection of Service Interactions in Telecommunications Service Specifications”, IEEE International Conference on Communications (ICC'94) pp.1835-1841, May 1994.

(9) 井上、高見、竹中：“複数端末にわたる通信サービスの競合検出法”、信学春季全大 B-525、(1992)

(10) 原田、高見、太田：“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法”、信学秋季全大 B-385、(1992)

(11) 井上、高見、太田：“処理の実行順序に着目したサービス競合解消法、信学秋季全大 B-380、(1992)

(12) 井上、高見、太田：“サービス競合における異常な状態遷移の検出法”、信学春季全大 B-578、(1993)

(13) 原田、高見、太田：“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”、信学春季全大 B-577、(1993)

(13) 井上、高見、太田：“通信サービス仕様間の矛盾の自動検出法”、情報処理学第47回全国大会、(1993)

(14) 原田、高見、太田：“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”、信学春季全大 B-619、(1994)

(15) 井上、高見、太田：“通信サービス仕様における異常な状態への遷移の解消支援法”、信学春季全大 B-620、(1994)

(16) 井上、高見、太田：“通信サービス記述における動作競合の解消支援法”、信学技報 SSE92-7、(1992-05)

(17) 原田、高見、太田：“通信サービス課金処理競合検出方式の検討”、信学技報 AI92-58、(1992-09)

(18) 井上、高見、太田：“通信サービス仕様におけるサービス競合の検出法”、信学技報 SSE92-143、(1993-02)

(19) 原田、高見、太田：“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－”、 信学技報 SSE92-163、(1993-03)

(20) Hirakawa, Y., Takenaka, T.:“Telecommunication Service Description Using State Transition Rules,” Sixth Int. Workshop on Software Specification and Design, Como, Italy Oct. 1991.

(21) Takami, K., Harada, Y., Ohta, T. and Terashima, N.:“A Visual Design Support System for Telecommunications Services,” IEEE Phoenix Conference on Computers and Communications, March 1993.

(22) Shibata, K., Hirakawa, Y., Takenaka, T.:“Reachablity analysis for a behavior description independent of the number of processes,” 5th joint workshop on computer communication, July 1990.

(23) 佐藤：“広域状態遷移規則を用いた通信サービス仕様記述における可到達解析，”電子情報通信学会論文誌 B-I, Vol.J78-B-I No.6, pp.245-251 June 1995

(24) 松本、佐藤：“非単調論理と常識推論，”情報処理学会誌, Vol.30 No.6, June 1989.

(25) 人工知能学会：人工知能ハンドブック, 1990.

(26) 井上、高見、太田：“通信サービス仕様における異常な状態への遷移の解消支援法”、信学春季全大B-620,1994.

第4章

(1) 水野、新津：“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”、電子情報通信学会誌論文誌、Vol.J74-B-I,No.12,pp1042-1055,(1991).

(2) Niitsu,Y. and Mizuno,O. :“Computer-Aided Stepwise Service Creation for The Intelligent Network,” IEICE Trans. Commun. Vol.E75-B,No.10 Oct. 1992.

(3) 特集：通信システムの形式記述技法の標準化、情報処理学会誌、Vol.31 No.1 pp.2-81、(1990-01)

(4) Hirakawa,Y and Takenaka,T:“Telecommunication Service Description Using Sate Transition Rules,” Sixth Int. Workshop on Software Specification and Design, Como, Italy,(1991).

(5) 高見、新津：“通信サービス仕様におけるメッセージシーケンス図記述からルール記述への仕様変換法”，情報処理学会論文誌 Vol.36,No.5 pp.1081-1090,May 1995.

(6) 高見、新津：“メッセージシーケンス記述によるサービス仕様から状態遷移規則の獲得法”、情報処理学会第49回全国大会 6N-5、(1994-09)

(7) Okamoto,M. and Niitsu,Y. :“A Verification Scheme for Service Specifications Described by Information Sequence Charts,” IEICE Trans. Commun. Vol.E75-B,No.10 Oct. 1992.

(8) Takami,K., Harada,Y., Ohta,T., and Terashima,N. :“A Visual Design Support System for Telecommunication Services,” IEEE Phoenix Conference on Computers and Communications (IPCCC'93), (1993).

(9) Harada,Y., Hirakawa,Y., Takenaka,T. and Terashima,N. :“A Conflict Detection Support Method for Telecommunication Service Descriptions,” IEICE Trans. Commun., Vol.E75-B,No.10,(1992).

(10) Inoue,Y., Takami,K. and Ohta, T. :“Automatic Detection of Service Interactions in Telecommunications Service Specifications,” IEEE International Conference on Communications (ICC'94),May 1994.

(11) Ohta,T., Takami,K. and Takura,A. :“Acquisition of Service Specification in two-stage Detection/Resolution of Feature Interaction,” The 4th Telecommunications Information Networking Architecture Workshop (TINA'93), (1993).

第5章

(1) Gilmour, J. and Gove, R.D.:“Intelligent Network/2: -The Architecture -The Technical Challenges -The Opportunities,” IEEE Communications, 12, pp.8-11

(1988).
（2）高見、竹中：“仕様知識に基づく通信ソフトウェア設計支援方式の検討”、信学秋季全大 B-333、(1991)
（3）K. Genji, K. Takami, T. Takenaka:“Formalization of Experiental Knowledge for Message Sequence Design and Application”, in Proc. of the 4th Joint Conference on Communication, Networks and Switching System(JC-CNSS'91),1991.
（4）元治、高見、竹中：“信号系列の規則性を用いた通信ソフトウェアの設計支援”、信学秋季全大 B-353、(1991)
（5）高見、元治、竹中：“信号規則による通信サービスのメッセージシーケンス設計支援”、信学春季全大 B-521、(1992)
（6）K. Genji, K. Takami, T. Takenaka：“Visual Prototyping Design Support System Using Message Sequence Rules for Communication Network Services”, IEEE International Conference on Communications (ICC'92) 352.3 pp.1553-1557, June 1992.
（7）K. Genji, K. Takami, T. Takenaka：“Telecommunication Service Design Support System Using Message Sequence Rules”, IEICE TRANS. COMMUN., Vol.E75-B, No.8 pp.723-732 Aug. 1992.
（8）K. Takami, T. Ohta：“An Interactive Specification Synthesis Support System using Message Sequence Rules from Telecommunication Service Requirement Specification”, IEEE International Conference on Telecommunications(ICT'94) pp.101-104, Jan. 1994.
（9）和田、高見、太田：“通信サービス動作知識を用いた例外シーケンスの絞り込み法”、信学秋季全大 B-383、(1992)
(10) TR-TSY-000522 “Features Common to Residence and Business Customers III LATA Switching Systems Generic Requirements,” Bell Communications Research, Bellcore, (1989).
(11) Hayashi, K. and Nishizono, T.:“Evaluation Mechanism of Constraint Oriented Action Superposition Specification Description Method,” JSAI 3rd Annual Conf., July, (1989).

(12) Nishizono, T. and Akiyasu, Z.:“Visual Prototyping System for Designing Distributed Communication Software,” Nat. Conv. IPS Jpn., (1989).

第6章

（1）服部：“パーソナル通信の動向と展開”,電子情報通信学会誌 Vol.78 No.5 pp.506-509 (May 1995).
（2）中島：“パーソナル移動通信のためのインテリジェントネットワーク”、電子情報通信学会誌 Vol.78 No.2 pp.143-149 (Feb. 1995).
（3）鈴木：“高度インテリジェントネットワーク”、電子情報通信学会誌 Vol.77 No.4 pp.410-415 (Apr. 1994).
（4）K. Takami, H. Sekiguchi, T. Sugiyama, M. Honda, I. Ushijima :“An Application of Advanced Intelligent Network Technology to Personal Communications Services”, IEEE International Intelligent Network Workshop(IN'96), April 1996.
（5）T. Sugiyama, T. Omiya, K. Takami, S. Esaki: “A modeling and protocol of the out-channel interaction for PCS in Intelligent Network”, Special Issue on Personal Communications, IEICE Trans. Commun. Vol.E79-B, No.9 Sep. 1996.
（6）T. Sugiyama, T. Ohmiya, K. Takami, :“A Modeling and Protocol for the Out-channel User Interaction in Intelligent Network which is Applicable to the Personal Communication”, IEEE International Conference on Communications (ICC'96), June 1996.
（7）高見、関口：“移動通信サービスにおけるＶＬＲ型網間ローミング方式の検討”，信学秋季全大 B-620, (1995-09)
（8）本多、高見、牛嶌：“高度ＩＮサービス展開のためのネットワークアーキテクチャ”，情処第51回全大3E-07 (1995-09)
（9）佐野、本多、高見：“ 高度ＩＮにおけるパーソナル通信サービスのためのＳＬＰ構成法”,信学春季全大 B-705、(1996-03)
(10) 高見、関口、杉山、本多、牛島：“高度ＩＮにおけるパーソナル通信サービス制御

方式”、信学技報 SSE95-95 、(1995-10)
(11) 杉山、大宮、高見、江崎：“パーソナル通信に適用可能なＩＮにおけるアウトチャネルインタラクションのモデリングとプロトコル”，信学技報SSE95-109 、(1995-11)．
(12) 本多、高見、関口：“パーソナル通信のための高度ＩＮにおけるサービスポータビリティ制御法”，信学技報SSE95-132(1995-12).
(13) 田中、竹内、鈴木、両角：“PHS を応用したマルチメディア無線通信システム”, National Technical Report Vol.40 No.6 pp.56-61 (Dec. 1994).
(14) 飯田：“パーソナル移動通信のためのソフトウェア”、電子情報通信学会誌 Vol.78 No.2 pp.150-155 (Feb. 1995).
(15) 新津、岡本、高見：“高度ＩＮにおける統合サービス生成環境構成法”、信学技報 SSE94-173 (Jan. 1995).
(16) 浦田、水野：“高度ＩＮにおけるサービスソフトウェア間インタフェース設計法”、信学技報SSE94-255 (Mar. 1995).
(17) B. Jabbari, G. Colombo, A. Nakajima and J. Kulkarni:“Network Issues for Wireless Communications”, IEEE Communications Magazine pp.88-98 (Jan. 1995).

第７章

（１）K. Takami, T. Ohta:“A Design Method for Telecommunication Service Software and its Support System Using Specification Knowledge”, in Proc. of the 5th Joint Conference on Communication, Networks and Switching System(JC-CNSS'92),1992.
（２）鈴木：“高度インテリジェントネットワーク”、電子情報通信学会誌 Vol.77 No.4 pp.410-415 (Apr. 1994).
（３）高見、新津：“アイコニックな通信サービス仕様記述支援のための知識構成法”、信学秋季全大 B-544、(1994-09)
（４）牛嶌、高見：“視覚化に重点をおいたソフトウェア開発環境モデルとソフトウェア生産性の相関に関する一考察”，情処第51回全大、(1995-09)

（５）和田、高見、太田：“信号シーケンス設計支援法－サービス仕様からＭＳＣへの変換法－”、信学技報 SSE92-157、(1993-03)
（６）H. Sakai, K. Takami, Y. Niitsu:“A Service Logic Program Generation Method for Advanced Intelligent Network”, IEEE International Conference on Communications (ICC'96), June 1996.
（７）坂井、高見、新津：“高度ＩＮにおけるサービスソフトウェア生成法”、情報処理学会第49回全国大会 6N-1、(1994-09)
（８）坂井、高見、岡本、新津：“高度ＩＮにおけるＳＬＰへの準正常手順自動補完法の検討”、信学春季全大 B-667、(1995-03)
（９）新津、岡本、高見：“高度ＩＮにおける統合サービス生成環境構成法”，信学技報 SSE94-173(1995-01).
(10) 坂井、高見、新津：“高度ＩＮにおける通信サービスソフトウェア自動生成法”，信学技報SSE94-254(1995-03).
(11) 坂井、高見、新津：“高度ＩＮにおけるＳＬＰ自動生成ツール構成法の検討”，信学技報SSE95-151(1996-02).
(12) 水野、新津：“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, Vol.J74-B-I, No.12, pp.1042-1055, 1991.
(13) 岡本、新津：“高度ＩＮサービス生成環境におけるサービスソフトウェア検証法,” 信学技報, SSE94-85, 1994.
(14) 技術士ソフトウェア研究会編：“ソフトウェア生産工学ハンドブック,” 1991.
(15) NTT 技術参考資料：“ＩＮＳネットサービスのインタフェース第３分冊,” 1992.

付録

第２章

付録２．１：電話サービスの概要

（１）POTS（Plain Old Telephone Service:基本電話サービス）

２者間の通話を実現するサービスである。発信者が相手端末にダイヤルすると、相手端末が呼び出し音により呼び出され、相手が受話器を上げると通話ができるサービスである。

（２）CW（Call-Waiting:話中着信）

通話中の端末に対してダイヤルしたとする。この時、通話中の相手が話中着信サービスに加入している場合には、ダイヤルした端末は呼び返し音受信の状態となり、着側には、通話中に着信があったことが音により通知される。この状態で、着側がフラッシュフックにより、通話相手を切り替えることができ、最初に通話していた相手は保留される。

（３）CFV(Call-Forwarding Variable:着信転送)

利用者の端末が通話中で、その端末に着信があった場合、事前に転送先端末を登録しておくと、その登録されている端末に呼びを転送するサービスである。

（４）TWC(Three-Way Calling:３者通話)

３者による通話を提供するサービスである。２者通話をしている時に３者通話のコントローラの機能をもっている何方かの端末がフラッシュフックして、３者めの相手にダイヤルして呼び出す。呼び出された３者めの相手が応答し、２者通話となった状態で、コントローラの機能を持った端末がフラッシュフックすると３者通話状態となる。

（５）CCBS(Completion of Calls on Busy Subscribers:話中時再呼出し)

２者間での通話接続において、ダイヤルして呼び出した相手がビジーの場合に、本サービスを起動すると、相手が通話を終了して空き状態となった時に、双方を呼び出してくれるサービスである。

付録２．２：アニメーション時の絵シンボルとＳＴＲ言語の状態記述要素との対応表

symbol	状態記述要素
ts_1	idle(A)
ts_2	idle(A) 以外の状態記述要素
ts_3	ringing(B,A),r-path(A,B)
ts_4	cw-ringing(B,A),r-path(A,B)
ts_5	path(A,B),path(B,A)
ts_6	path-passive(A,B),path(B,A)

（注）ts_iは表２．３の絵シンボルを表す。

付録２．３：CWサービスのＳＴＲ言語によるテキスト形式記述例

r1)idle(A),path(B,C),path(C,B),m-cw(B)

dial(A,B):

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

r2)ringback(A,B),cw-ringing(B,A),r-path(A,B),path(B,C),path(C,B),m-cw(B)

flash(B):

path(A,B),path(B,A),path-passive(B,C),path(C,B),cw(B).

付録２．４：CWとCFVサービスの競合ルール（文献１９より）

以下のようなサービス競合前の状態において、

端末Ａ：dial-tone(A)

端末Ｂ：path(B,C),m-cw(B),m-cfv(B,D)

端末Ｃ：path(C,B)

端末Ｄ：idle(D)

以下のルールが競合する。

r_{cw} ＝dial-tone(A),path(B,C),path(C,B),m-cw(B)

dial(A,B):

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

r_{CFV} =dial-tone(A),not [idle(B)] ,m-cfv(B,C),idle(C)
dial(A,B):
ringback(A,C),ringing(C,A),r-path(A,C),m-cfv(B,C),not [idle(B)] .

付録2.5 ：POTS,CW,CFV,TWC,CCBSサービスの競合検出結果（文献19より抜粋。付録2.4以外のルール）

CW+CFVの時

r_{CW} =dial-tone(A),path(B,C),path(C,B),m-cw(B)
dial(A,B):
ringback(A,B),cw-ringing(B,A),m-cw(B),r-path(A,B),
path(B,C),path(C,B).

r_{CFV1}=dial-tone(A),path(B,C),path(C,B),m-cfv(B,C),not [idle(C)]
dial(A,B):
busy(A),m-cfv(B,C),not [idle(C)] .

TWC+CCBSの時

r_{TWC} =path-passive(A,B),m-3wc(A)
onhook(B): empty.

r_{CCBS}=m-CCBSed(A,B)
onhook(B): empty.

CW+CCBS の時

r_{CW1} =path(A,C),cw(A),path-passive(A,B)
onhook(B): path(A,C).

r_{CCBS}=m-CCBSed(A,B)
onhook(B): empty.

付録2.6 ：生成された解消ルール

CW+CFVの時

$r_{CW>CFV}$=dial-tone(A),not [idle(B)] ,path(B,C),path(C,B),m-cw(B),
m-cfv(B,D),idle(D)
dial(A,B):

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

$r_{CFV>CW}$=dial-tone(A),not [idle(B)] ,path(B,C),path(C,B),m-cw(B),
m-cfv(B,D),idle(D)
dial(A,B):
ringback(A,D),ringing(D,A),r-path(A,D),m-cw(B),m-cfv(B,D),
path(B,C),path(C,B),not [idle(B)] .

TWC+CCBSの時

$r_{TWC>CCBS}$=path-passive(A,B),m-3wc(A),m-CCBSed(A,B)
onhook(B): m-CCBSed(A,B).

$r_{CCBS>TWC}$=path-passive(A,B),m-3wc(A),m-CCBSed(A,B)
onhook(B):
path-passive(A,B),m-3wc(A).

CW+CCBS の時

$r_{CW1>CCBS}$=path(A,C),cw(A),path-passive(A,B),m-CCBSed(A,B)
onhook(B):
path(A,C),m-CCBSed(A,B).

$r_{CCBS>CW1}$=path(A,C),cw(A),path-passive(A,B),m-CCBSed(A,B)
onhook(B):
path(A,C),cw(A),path-passive(A,B).

第3章

付録3.1 ：ＳＴＲ言語によるサービス仕様の記述例

着信転送サービス(CFV) と着信拒否サービス(TCS) の基本仕様に関する記述例を以下に示す。実験では、このほかに2者間の通信サービス（基本電話サービス）仕様を記述した9個のルールを加えて行った。

（1）着信転送サービス(CFV)

1)dial-tone(A) cfv(A): cfv-tone(A).

2)dial-tone(A),m-cfv(A,B) cfv(A): cfv-tone(A).

3)cfv-tone(A),idle(B) dial(A,B): confirm-tone(A),m-cfv(A,B),idle(B).

4)cfv-tone(A),not[idle(B)] dial(A,B): confirm-tone(A),m-cfv(A,B).

5)cfv-tone(A) dial(A,A): busy-dial(A).

6)dial-tone(A),cond:idle(B),idle(C),cond:m-cfv(B,C)
dial(A,B): ringback(A,C),ringing(C,A).

7)dial-tone(A),not[idle(B)],idle(C),cond:m-cfv(B,C)
dial(A,B): ringback(A,C),ringing(C,A).

8)dial-tone(A),cond:idle(B),cond:m-cfv(B,C),not[idle(C)]
dial(A,B): busy-dial(A).

9)dial-tone(A),not[idle(B)],cond:m-cfv(B,C),not[idle(C)]
dial(A,B): busy-dial(A).

10)cfv-tone(A) onhook(A): idle(A).

11)confirm-tone(A) onhook(A): idle(A).

（2）着信拒否サービス(TCS)

1)dial-tone(A) tcs(A): tcs-tone(A).

2)dial-tone(A),m-tcs(A,B) tcs(A): tcs-tone(A).

3)tcs-tone(A),idle(B) dial(A,B): confirm-tone(A),m-tcs(A,B),idle(B).

4)tcs-tone(A),not[idle(B)] dial(A,B): confirm-tone(A),m-tcs(A,B).

5)tcs-tone(A) dial(A,A): busy-dial(A).

6)dial-tone(A),cond:m-tcs(B,A),idle(B) dial(A,B): busy-dial(A),idle(B).

7)dial-tone(A),cond:m-tcs(B,A),not[idle(B)] dial(A,B): busy-dial(A),idle(B).

8)confirm-tone(A) onhook(A): idle(A).

9)tcs-tone(A) onhook(A): idle(A).

第4章

付録4.1：複数の状態変換要素を含むＭＳＣ記述例

２つの状態変換要素が含まれる記述例を付図4.1に示す。端末Ａから通信網にメッセージm_1が送出され、端末Ｂに対してはメッセージm_2により通信網から指示され、端末Ａ

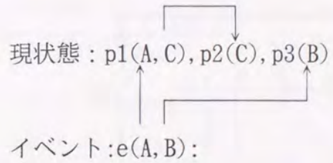
とＣは安定状態を示すメッセージm_3になる、記述を示している。このような記述は、CWサービス（付録2.1参照）のflash hookによる通話保留と通話の切替えの仕様を記述する時に使われる。

付録4.2：複数のイベント変換要素を含むＭＳＣ記述例

２つのイベント変換要素を含む記述例を付図4.2に示す。付図4.2には、端末Ａから通信網へのメッセージm_1が分岐値としてv_1とv_2を持つ場合を示している。それぞれの分岐後の通信網から端末へのメッセージの送出パターンが異なり、表4.4の項番１と項番３の変換要素種別が含まれている。このような記述は、端末がダイヤルした時に、相手端末が空き状態の場合と通話中状態の場合で着信するかビジー音返しにするかの仕様を記述する時に使われる。

付録4.3：ルールマッチング制約

一つのルールにおいて、現状態（状態記述要素の集合で定義）とイベントとに記述する端末識別子の関係は、イベントの端末識別子を元に、１）その識別子を第１引数に持つ状態記述要素を辿り、２）辿れた状態記述要素に第２引数がある場合は、その識別子を第１引数に持つ状態記述要素を辿り、現状態に記述した全ての識別子が辿れなければならない。例を以下に示す。



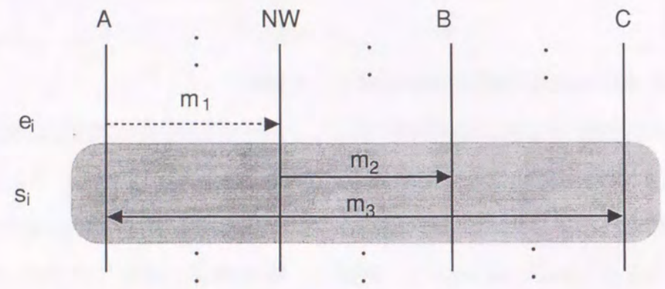
付録4.4：着信拒否サービス(Terminal Call Screening）の概要

事前に接続したくない端末の番号を登録しておく。登録している端末から着信があった場合、その端末との接続を拒否するサービスである。

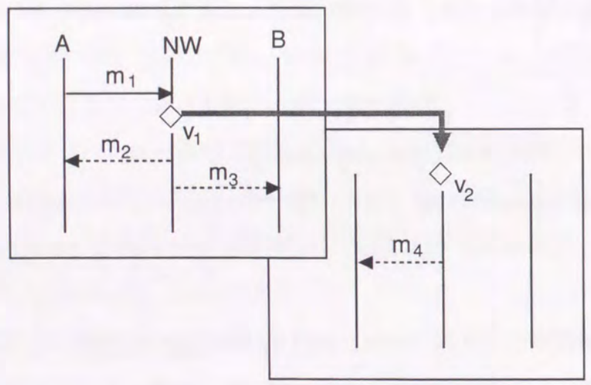
付録4.5：基本電話サービス(POTS)の変換例

（１）ＭＳＤＬによる記述例

process(pots)



付図4.1　2つの状態変換要素による端末の状態の構成例



付図4.2　2つのイベント変換要素に対応するメッセージの例


```
{
  .path(A,NW);
  .path(NW,B);
  offhook(A,NW);
  dial-tone(NW,A);
  dial(A,NW:v);
  .if(v=idle(B))  {
    ringback(NW,A);
    ringing(NW,B);
    offhook(B,NW);
    speaking(A,B);
    {
        onhook(B,NW);
        busy(NW,A);
        onhook(A,NW);
        .stop(NW,process);
      |
        onhook(A,NW);
        busy(NW,B);
        onhook(B,NW);
        .stop(NW,process);
    }
} ;
 .if(v=not[idle(B)]) {
        busy(NW,A);
        onhook(A,NW);
        .stop(NW,process);
      {
} ;
```

（２）本章にもとづき変換したＳＴＲ言語の例

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

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

3)ringback(A,B),ringing(B,A) offhook(B): speaking(A,B),speaking(B,A).

4)speaking(A,B),speaking(B,A) onhook(B): busy(A),idle(B).

5)busy(A) onhook(A): idle(A).

6)dial-tone(A),not[idle(B)] dial(A,B): busy(A).

第５章

付録５．１： Message Sequence Rule Examples

(1) Temporal relation of an input to outputs

(1-1) x?request -> x!notify | y!query

(1-2) x?query -> x!ack | x!reject

(1-3) x?ack -> y!notify

(1-4) x?reject -> y!notify

(1-5) x?complete -> y!complete | NONE

(2) Temporal relation of an output to subsequent inputs

(2-1) x!query -> x?ack U (x?ack | x?reject)

| x?rejectU (x?ack | x?reject)

(2-2) x!ack -> x?complete U (x?complete)

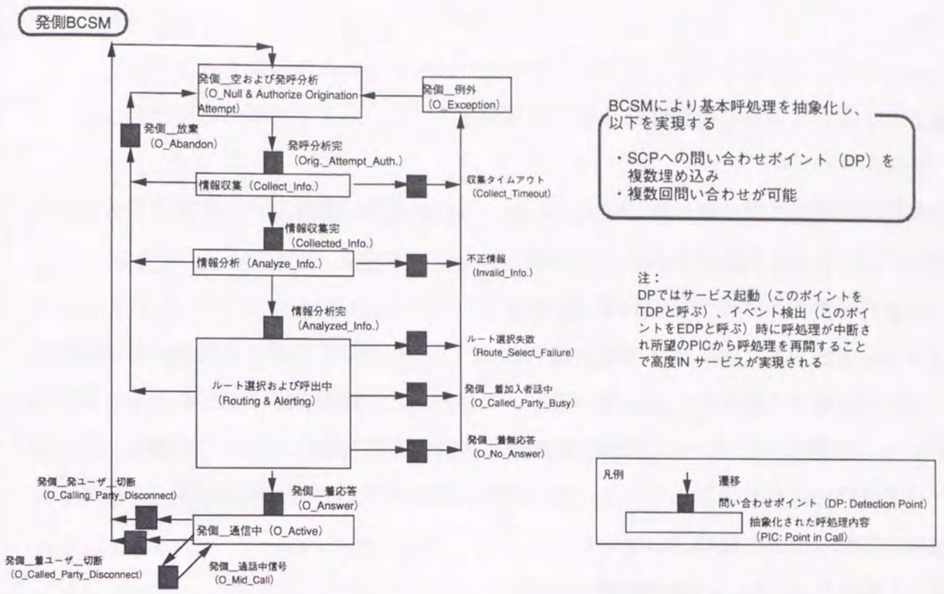
(3) Temporal relation of an input to subsequent inputs

(3-1) x?request -> x?complete U (x?complete)

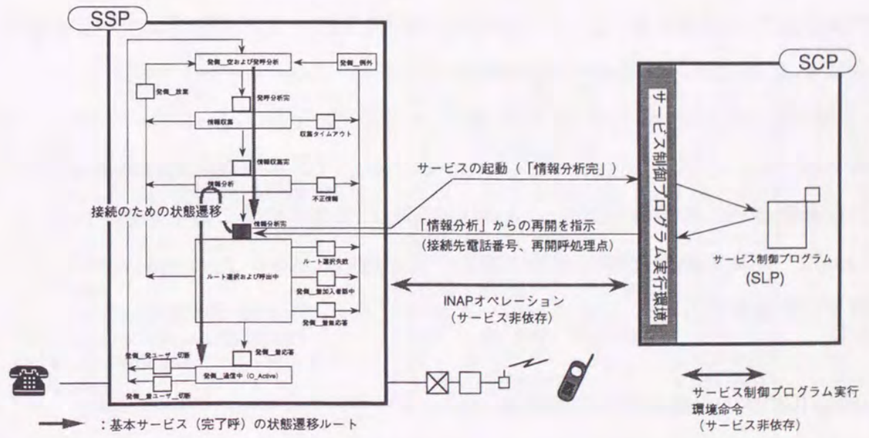
第６章

付録６．１：BCSMとＳＬＰによるサービス制御

発側BCSMを付図６．１に示す。BCSMとＳＬＰによるサービス制御例を付図６．２に示す。



付図6.1 BCSM （Basic Call State Model）による基本呼処理の抽象化



付図6.2 BCSM制御によるPCS基本接続の提供

付録7.１：要求仕様から機能概要仕様への変換法

1.　仕様記述法

要求仕様と機能概要仕様は第４章で示した、ＳＴＲ言語による記述法とＭＳＤＬによる記述法によりそれぞれ記述されているものとする。

2.　ユーザとネットワーク間のＭＳＣ生成法

ＭＳＣを生成するためには、端末の状態とイベントの時間的な関係を獲得する必要がある。ＳＴＲを適用することによって、縮退グローバル状態遷移図（RGSTG)を要求仕様から生成する。RGSTGには、サービス動作を表現するのに必要な複数の端末の状態からなるグローバル状態により構成されている。最小限の必要な端末数は着目する端末とその端末に関連のある端末により構成されている。

定義１：縮退グローバル状態遷移図（RGSTG）

RGSTGは、以下の５つの要素により構成される。

$RGSTG=<T, RGS, E, \Phi, rgs_0>$

ここで、

ａ）Ｔはサービス動作を表現するために最小限の必要な端末の集合である。

ｂ）RGS は縮退グローバル状態rgs_i の集合である。rgs_i は最小限の必要な端末の状態の集合として表現される。$\{S(t_i) \mid t_i \in T\}$

ｃ）Ｅはt_iに受付けられるイベントの集合である。

ｄ）Φはrgs_iからrgs_jへの遷移関数である。$\Phi(rgs_i, e_r/r) = rgs_j$、は$rgs_i$において、イベント$e_r$が選択され、ルールｒ（要求仕様記述に含まれるＳＴＲ）が適用され、次の縮退グローバル状態rgs_jが生成されることを示す。

ｅ）rgs_0は、初期の縮退グローバル状態を示す。$rgs_0 \in RGS$, $rgs_0 = S(t_i)$ でt_iは着目する端末である。

□

定義２：RGSTG生成のためのＳＴＲ適用法

あるrgs_i（$rgs_i \in RGS$）において、ルールｒが次の２つの条件を同時に満足する時、ルールｒが適用され、次の縮退グローバル状態rgs_jが生成される。

（１）あるrgs_iにおいて、$\{(s_i \supseteq cs_{ri}) \vee (cs_{ri} \supseteq s_i) \vee (s_i \cap cs_{ri} \neq \phi)\}$

が満足される。

（２）上記（１）を満足するルールｒの次状態記述の中に、着目する端末の状態を変化させる状態記述が存在する。

□

以下に示す変換ルールを使って、ネットワークと端末からなるＭＳＣ記述要素がRGSTGの中の縮退グローバル状態rgsとイベントe_rから生成される。

定義３：縮退グローバル状態rgsとイベントe_rのＭＳＣ記述要素への変換ルール

着目する端末とその端末と関連のある端末をそれぞれt_0，t_i（$i=0,1,\cdots k$）とする。ネットワークはNWと示す。縮退グローバル状態rgsが、状態記述要素ｐを使って表現されている時、ｐとe_rは以下のようにして変換される。

（１）$p(t_0) \rightarrow n_p(NW, t_0)$

（２）$p(t_0, t_i) \rightarrow n_p(NW, t_0)$

（３）$p(t_i, t_0) \rightarrow n_p(NW, t_i)$

（４）$e_r(t_0)$ 又は $e_r(t_0, t_i) \rightarrow n_{er}(t_0, NW)$

（５）$e_r(t_i)$ 又は $e_r(t_i, t_0) \rightarrow n_{er}(t_i, NW)$

ここで、n_z（$z=p, e_r$）はｚに関連するＭＳＣ記述要素名を示す。

□

付録7.２：主なINAPオペレーションと略号の意味

名称	略号	意　味	方　向	クラス等
Connect	CNCT	呼接続指示	SCF→SSF	CL2
PlayAnnouncement	PA	ガイダンス送出	SCF→SRF	CL1
PromptAndCollectUserInformation	PCUI	入力情報収集	SCF→SRF	CL1
ConnectToResource	CTR	特殊ﾘｿｰｽ接続指示	SCF→SSF	CL2
AnalyzeInformation,	AI	AI-PIC指示	SCF→SSF	CL2
SelectFacility,	SF	空き回線選択指示	SCF→SSF	CL2
Continue	CONT	処理一時停止解除	SCF→SSF	CL4
RequestReportBCSMEvent	RRBE	ｲﾍﾞﾝﾄ通知要求	SCF→SSF	CL2
SpecializedResourceReport	SRR	特殊ﾘｿｰｽ処理報告	SRF→SCF	成功応答
ReturnError(xx)	RE	xxエラー	SSF→SCF	ｴﾗｰ 応答
OAbandan	OABD	発側放棄	SSF→SCF	ｴﾗｰ 応答

（注）　CL1:RESULT,ERRORS を共に記述、CL2:ERRORSのみ記述

CL3:RESULTのみ記述、CL4:RESULT,ERRORS どちらも記述なし。

発表論文一覧（下線は関連発表論文を示す）

A：論文誌

（1）高見一正、井上泰彰、太田　理：“サービス競合における異常な状態への遷移の検出法”，情報処理学会論文誌 Vol.36,No.1,pp.93-104,Jan.1995.

（2）高見一正、太田　理：“通信サービスの視覚的な要求仕様化支援法”，情報処理学会論文誌 Vol.36,No.3 pp.728-741,Mar.1995.

（3）高見一正、新津善弘：“通信サービス仕様におけるメッセージシーケンス図記述からルール記述への仕様変換法”，情報処理学会論文誌 Vol.36,No.5 pp.1081-1090,May 1995.

（4）K. Genji, K. Takami, T. Takenaka：“Telecommunication Service Design Support System Using Message Sequence Rules”, IEICE TRANS. COMMUN., Vol.E75-B, No.8 pp.723-732 Aug. 1992.

（5）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”，情報処理学会論文誌 Vol.34 No.5 pp.1064-1073, May 1993.

（6）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”、情報処理学会論文誌 Vol.35 No.8 pp.1602-1613, Aug.1994.

（7）M. Nagasawa, Y. Hiramatsu, K. Takami：“Packet Switched Network Access Protocols for Multi-media Packet Communications”, The Journal of the Institution of Electronic and Radio Engineers, Vol.56 No.6/7 pp.243-247, June/July 1986.

（8）竹中豊文、赤池武志、高見一正：“非割り込み優先処理型ΣMi/G/1（ΣNi）システムの特性評価と近似解析手法”，電子情報通信学会論文誌B Vol. J70-B No.11 pp. 1338-1345, Nov. 1987.

（9）T. Sugiyama, T. Omiya, K. Takami, S. Esaki: “A modeling and protocol of the out-channel interaction for PCS in Intelligent Network”, Special Issue on Personal Communications, IEICE TRANS. COMMUN., Vol.E79-B, No.9 pp.1388-1393 Sep. 1996.

B　ＮＴＴ社内機関誌

（1）門田充弘、高見一正、中村雅夫、石塚義明：“電話・パケット網間接続制御プロトコル”ＮＴＴ研究実用化報告 Vol.32,No.5,pp.1083-1096,昭和58年11月.

（2）M. Monden, K. Takami, M. Nakamura:“DTE/DCE and Interexchange Interfaces for the Telephone-Network Packet-Network Interworking System”, REVIEW of the Electrical Communication Laboratories, Vol.31,No.6,pp.755-766,1983.

C：国際会議

（1）K. Takami, T. Takenaka: “Architectual and Functional Aspects of a Packet Switched Network for Integrated Services”, International Switching Symposuim 1987(ISS'87) B6.4.1 pp.470-474, March 1987.

（2）K. Takami, T. Ohta:“A Design Method for Telecommunication Service Software and its Support System Using Specification Knowledge”, in Proc. of the 5th Joint Conference on Communication, Networks and Switching System(JCCNSS'92),1992.

（3）K. Takami, Y. Harada, T. Ohta, N. Terashima：“A Visual Design Support System for Telecommunicaton Services”, IEEE Phoenix Conference Computers and Communications(IPCCC'93) pp.593-599, March 1993.

（4）K. Takami, T. Ohta, N. Terashima：“A Visual Programming for Telecommunication Services”, IEEE/CS Symposium on Visaul Languages(VL'93) pp.360-361, Aug. 1993.

（5）K. Takami, T. Ohta：“An Interactive Specification Synthesis Support System using Message Sequence Rules from Telecommunication Service Requirement Specification”, IEEE International Conference on Telecommunications(ICT'94) pp.101-104, Jan. 1994.

（6）K. Takami, H. Sekiguchi, T. Sugiyama, M. Honda, I. Ushijima：“An Application of Advanced Intelligent Network Technology to Personal Commu-

nications Services", IEEE International Intelligent Network Workshop(IN'96), April 1996.

(7) H. Yasuda, K. Takami : "Routing Functional Arrangements for A Distributed Switching System In B-ISDN", IEEE Global Telecommunication Conference(Globecom'89) 25.4.1 pp.893-897, Nov. 1989.

(8) K. Genji, K. Takami, T. Takenaka:"Formalization of Experiental Knowledge for Message Sequence Design and Application", in Proc. of the 4th Joint Conference on Communication, Networks and Switching System(JC-CNSS'91), 1991.

(9) K. Genji, K. Takami, T. Takenaka : "Visual Prototyping Design Support System Using Message Sequence Rules for Communication Network Services", IEEE International Conference on Communications (ICC'92) 352.3 pp.1553-1557, June 1992.

(10) Y. Inoue, K. Takami, T. Ohta : "Method for Supporting Detection and Elimination of Feature Interaction in a Telecommunication System", IEEE International Workshop on Feature Interactions In Telecommunications Systems pp.61-81, Dec. 1992.

(11) T. Ohta, K. Takami, A. Takura:"Acquisition of Service Specification in Two-stage Detection / Resolusion of Feature Interaction", Telecommunications Information Networking Architecture Workshop(TINA'93) pp.Ⅱ-173-187, Sept. 1993.

(12) Y. Inoue, K. Takami, T. Ohta : "Automatic Detection of Service Interactions in Telecommunications Service Specifications", IEEE International Conference on Communications (ICC'94) pp.1835-1841, May 1994.

(13) H. Enoki, K. Takami, Y. Kobayashi, T. Ohta:"Swiching Function Conceptual Model for Telecommunication Service Specification Design", IEEE Global Telecommunication Conference(Globecom'94) pp.818-822, Nov. 1994.

(14) H. Sakai, K. Takami, Y. Niitsu : "A Service Logic Program Generation Method for Advanced Intelligent Network", IEEE International Conference on Communications (ICC'96), June 1996.

(15) M. Nagasawa, Y. Hiramatsu, K. Takami : "Packet Switched Network Access Protocols for Multi-media Packet Communications", IERE Conference Networks and Electric Office Systems(NEOS'85) pp.115-120, Sept. 1985.

(16) M. Nakanura, Y. Inoue, K. Takami, T. Takenaka, T. Ohta : "Telecommunication Service Design Support Method Using Visual Service Description Language STR-GR", IEEE Global Telecommunication Conference(Globecom'92) pp.554-559, Dec. 1992.

(17) T. Sugiyama, T. Ohmiya, K. Takami, : "A Modeling and Protocol for the Out-channel User Interaction in Intelligent Network which is Applicable to the Personal Communication", IEEE International Conference on Communications (ICC'96), June 1996.

D　その他の機関誌及び解説記事

(1) 高見一正：“通信サービス記述・検証支援システム”、ＪＴＴ技術移転 Vol.15 No.6、(1992)

(2) 高見一正：“絵でプログラミング－アニメで確認－”、ＡＴＲジャーナル No.14、(1993)

(3) 高見一正、河原崎裕朗：“通信サービス仕様記述と視覚的プログラミング”、電子情報通信学会誌 Vol.78, No.12, pp.1260-1267, Dec. 1995.

(4) 市川晴久、高見一正：“通信ソフトウェアの自動生成技術”，電子情報通信学会誌 Vol.77, No.5 , pp.522-530, May 1994.

E　全国大会

(1) 高見一正、矢野　純、吉田健一郎：“パケット網・網間汎用接続方式の検討”，信学秋季全大No.363 、昭和56年10月

(2) 高見一正、平田　昭生：“低速パケット端末収容方式の一考察”，信学秋季全大No.369 、昭和56年10月

(3) 高見一正、一色耕治、矢野　純：“パケット集線局収容PTのトラヒック制御法”，

信学春季全大No.1637、昭和57年

（4）高見一正、竹中豊文、星田勝典：“電話・パケット網間信号方式の一考察”，信学秋季全大No.150、昭和57年

（5）高見一正、竹中豊文、木村重勝、田村好博：“電話網・パケット網・電話網接続に関する検討”、信学秋季全大No.1733、昭和58年

（6）高見一正、平松幸男、竹中豊文：“パケット交換におけるコネクションレスーX.25プロトコル変換の検討”、信学全大No.1862、昭和59年

（7）高見一正、赤池武志、竹中豊文：“オープンネットワークを指向したソフトウェア構成の一考察”、信学総大No.1864，昭和62年

（8）高見一正、竹中豊文、赤池武志、広重誠章：“画像情報のパケット転送制御方式の検討”、信学全大No.493 、昭和62年７月

（9）高見一正、竹中豊文：“仕様知識に基づく通信ソフトウェア設計支援方式の検討”、信学秋季全大 B-333、(1991)

(10) 高見一正、元治景朝、竹中豊文：“信号規則による通信サービスのメッセージシーケンス設計支援”、信学春季全大 B-521、(1992)

(11) 高見一正、原田良雄、太田　理：“通信サービス動作知識検証のための試験系列生成法”、信学秋季全大 B-386、(1992)

(12) 高見一正、太田　理：“状態遷移規則による通信サービス記述性評価”、信学春季全大 B-573、(1993)

(13) 高見一正、新津善弘：“アイコニックな通信サービス仕様記述支援のための知識構成法”、信学秋季全大 B-544、(1994-09)

(14) 高見一正、新津善弘：“メッセージシーケンス記述によるサービス仕様から状態遷移規則の獲得法”、情報処理学会第49回全国大会 6N-5、(1994-09)

(15) 高見一正、関口英生：“移動通信サービスにおけるVLR 型網間ローミング方式の検討”，信学秋季全大　B-620，(1995-09)

(16) 平松幸男、高見一正、永沢道夫：“パケット交換コネクションレスプロトコルの検討”、信学全大No.1861、昭和60年４月

(17) 赤池武志、高見一正、竹中豊文：“画像情報転送方式の検討”、信学全大シンポジュム(S6-5)、昭和61年６月

(18) 白石　智、高見一正：“統合交換用プロトコル交換方式の一考察”、情報処理学第39全大、平成２年７月

(19) 別所寿一、高見一正：“B-ISDNのための分散加入者アーキテクチャの検討”、信学春季全大、平成３年

(20) 元治景朝、高見一正、竹中豊文：“信号系列の規則性を用いた通信ソフトウェアの設計支援”、信学秋季全大 B-353、(1991)

(21) 中村光宏、高見一正、竹中豊文：“意味レベルの通信サービス仕様記述について”、信学秋季全大 B-337、(1991)

(22) 井上泰彰、高見一正、竹中豊文：“通信サービス仕様における機能要素を用いた設計支援”、信学秋季全大 B-335、(1991)

(23) 中村光宏、高見一正、竹中豊文：“通信サービス検索手法について”、信学春季全大 SB-5-4、(1992)

(24) 井上泰彰、高見一正、竹中豊文：“複数端末にわたる通信サービスの競合検出法”、信学春季全大 B-525、(1992)

(25) 原田良雄、高見一正、太田　理：“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法”、信学秋季全大 B-385、(1992)

(26) 井上泰彰、高見一正、太田　理：“処理の実行順序に着目したサービス競合解消法、信学秋季全大 B-380、(1992)

(27) 中村光宏、高見一正、太田　理：“プロダクションシステムに基づく通信サービス仕様記述法に関する一考察”、信学秋季全大 B-384、(1992)

(28) 和田　隆、高見一正、太田　理：“通信サービス動作知識を用いた例外シーケンスの絞り込み法”、信学秋季全大 B-383、(1992)

(29) 井上泰彰、高見一正、太田　理：“サービス競合における異常な状態遷移の検出法”、信学春季全大 B-578、(1993)

(30) 和田　隆、高見一正、太田　理：“ＳＴＲによるサービス仕様からＭＳＣへの仕様変換法”、信学春季全大 B-581、(1993)

(31) 原田良雄、高見一正、太田　理：“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”、信学春季全大 B-577、(1993)

(32) 榎木　浩、高見一正、太田　理：“仕様記述要素定義のための通信サービス概念モデルについて”、信学秋季全大　B-464 、(1993)

(33) 和田　隆、高見一正、太田　理：“通信サービスの日本語による仕様記述支援法の

一考察”、信学秋季全大　B-463 、(1993)
(34) 井上泰彰、高見一正、太田　理：“通信サービス仕様間の矛盾の自動検出法”、情報処理学第47回全国大会、(1993)
(35) 若林　学、高見一正、太田　理：“通信サービス仕様データベース構成法”、信学春季全大　B-621、(1994)
(36) 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”、信学春季全大　B-619、(1994)
(37) 井上泰彰、高見一正、太田　理：“通信サービス仕様における異常な状態への遷移の解消支援法”、信学春季全大　B-620、(1994)
(38) 坂井　博、高見一正、新津善弘：“高度ＩＮにおけるサービスソフトウェア生成法”、情報処理学会第49回全国大会 6N-1、(1994-09)
(39) 坂井　博、高見一正、岡本光浩、新津善弘：“高度ＩＮにおけるＳＬＰへの準正常手順自動補完法の検討”、信学春季全大　B-667、(1995-03)
(40) 本多麻理子、高見一正、牛嶌一郎：“高度ＩＮサービス展開のためのネットワークアーキテクチャ”，情処第51回全大（1995-09）
(41) 牛嶌一郎、高見一正：“視覚化に重点をおいたソフトウェア開発環境モデルとソフトウェア生産性の相関に関する一考察”，情処第51回全大（1995-09）
(42) 平松幸男、竹中豊文、高見一正：“パケット交換網におけるサービス統合化の一検討”、信学全大№231 、昭和59年10月
(43) 赤池武志、竹中豊文、高見一正：“多元パケット交換におけるオーバフロー特性の解析”、信学総大No1904，昭和61年12月
(44) 赤池武志、竹中豊文、高見一正：“多元パケット交換における優先処理方式の評価”、信学全大№496 、昭和62年７月
(45) 佐野直美、本多麻理子、高見一正：“ 高度ＩＮにおけるパーソナル通信サービスのためのＳＬＰ構成法”,信学春季全大　B-705、(1996-03)
(46) 坂井　博、西村正寿、高見一正、新津善弘：“高度ＩＮにおける通信サービス仕様補完法の検討”、信学秋季全大　B-629、(1996-09)

F　研究会

（1）高見一正、藤井敏孝、矢野　純：“パケット網・網間接続方式の検討”、信学交換研究会SE82-24 、昭和57年６月
（2）高見一正、平松幸男、永沢道夫：“マルチメディアに適合したパケット交換アクセスプロトコルの検討”、信学技法IN84-33、昭和59年８月
（3）高見一正、竹中豊文、野口清広：“高速パケット交換網アーキテクチャとその性能評価”、信学技報SE85-135、昭和61年１月
（4）高見一正、竹中豊文、赤池武志、広重誠章：“高速パケットにおけるメディア制御とそのプロトコル評価”、信学技報SE87-111、昭和62年11月
（5）高見一正、清野浩一、黒田憲一：“サービスの高度化を指向した交換プログラム構成法の検討”、信学技報SSE88-114、平成元年９月
（6）高見一正、平川　豊、河田慶三、竹中豊文：“ビジュアルプロトタイピングによる通信サービス仕様生成法”、信学技報 SSE91-103、(1991-11)
（7）高見一正、原田良雄、太田　理：“通信サービス記述仕様の確認支援法”、信学技報 SSE92-86 、(1992-11)
（8）高見一正：“非専門家のための通信サービス仕様記述法”、第１回通信ソフトウェア工学研究会（信学会第２種研究会）、(1993-07)
（9）高見一正：“プログラム自動合成に関する研究－通信サービスソフトウェアの自動生成－”、情報処理学会関西支部ソフトウェア研究会・講演会、(1995-06)
(10) 高見一正、関口英生、杉山武史、本多麻理子、牛島一朗：“高度ＩＮにおけるパーソナル通信サービス制御方式”、信学技報 SSE95-95 、(1995-10)
(11) 平松幸男、高見一正、永沢道夫：“LAN 相互接続を考慮したパケット交換コネクションレスサービスの実現法”、信学技報IN85-3，昭和60年４月
(12) 赤池武志、高見一正：“パケット交換網へのToken-Ring技術の適用に関する検討”、信学技報IN85-18,昭和60年６月
(13) 上村邦夫、高見一正：“LAN 技術の広域パケット交換網への適用”、信学技報 CS85-52、昭和60年８月
(14) 赤池武志、高見一正、竹中豊文：“高速パケットスイッチの構成とその性能評価”、信学技報SE85-149、昭和61年３月

(15) 平松幸男、高見一正、他　：“パケット交換網におけるLAN 相互接続”、マルチメディアと分散処理、昭和61年９月

(16) 井上泰彰、高見一正、竹中豊文：“仕様部品と規則による通信サービスのビジュアルプロトタイピングシステム”、信学技報 IN91-87 、(1991-09)

(17) 中村光宏、高見一正、竹中豊文：“分散協調アーキテクチャによる通信サービス仕様の生成法の検討”、第３回NAワークショップ、(1991)

(18) 中村光宏、高見一正、竹中豊文：“ビジュアルなサービス記述STR-GRによる通信サービス設計支援法”、信学技報 SSE91-179、(1992-03)

(19) 井上泰彰、高見一正、太田　理：“通信サービス記述における動作競合の解消支援法”、信学技報 SSE92-7、(1992-05)

(20) 原田良雄、高見一正、太田　理：“通信サービス課金処理競合検出方式の検討”、信学技報 AI92-58、(1992-09)

(21) 井上泰彰、高見一正、太田　理：“通信サービス仕様におけるサービス競合の検出法”、信学技報 SSE92-143、(1993-02)

(22) 原田良雄、高見一正、太田　理：“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－”、　信学技報 SSE92-163、(1993-03)

(23) 和田　隆、高見一正、太田　理：“信号シーケンス設計支援法－サービス仕様からＭＳＣへの変換法－”、信学技報 SSE92-157、(1993-03)

(24) 原田良雄、高見一正、太田　理：“通信サービス動作の修正影響解析支援法”、信学技報 SSE93-79 、(1993-10)

(25) 榎木　浩、高見一正、太田　理：“通信サービスの概念モデルの一考察”、信学技報 SSE93-77 、(1993-10)

(26) 和田　隆、高見一正、太田　理：“通信サービス仕様記述支援におけるガイダンス手法の一考察”、信学技報 SSE93-60 、(1993-09)

(27) 榎木　浩、高見一正、小林吉純、太田　理：“通信サービス概念モデルの応用に関する一考察”、信学技報 SSE93-126、(1993-12)

(28) 和田　隆、高見一正、太田　理：“非専門家を対象としたサービス仕様記述支援法”、信学技報 SSE93-138、(1994-01)

(29) 若林　学、高見一正、太田　理：“通信サービス仕様記述支援のためのデータベース構成法”、信学技報 SSE93-154、(1994-03)

(30) 坂井　博、高見一正、新津善弘：“高度ＩＮにおける通信サービスソフトウェア自動生成法”，信学技報SSE94-254 、(1995-03).

(31) 本多麻里子、高見一正、関口英生：“パーソナル通信のための高度ＩＮにおけるサービスポータビリティ制御法”，信学技報SSE95-132(1995-12).

(32) 坂井　博、高見一正、新津善弘：“高度ＩＮにおけるＳＬＰ自動生成ツール構成法の検討”，信学技報SSE95-151 、(1996-02).

(33) 平松幸男、竹中豊文、高見一正：“LAN 相互接続を考慮した公衆パケット交換サービスの検討”、分散処理システム研究会「LAN/マルチメディアの応用と分散処理」シンポジウム、昭和59年10月

(34) 赤池武志、竹中豊文、高見一正：“バースト的な情報源を持つ多元パケット優先転送方式の評価”、交換研究会、昭和62年６月

(35) 廣重誠章、安田洋史、高見一正：“分散型交換機におけるモジュール間通信方式の検討”、信学技報IN88-92 平成元年10月

(36) 新津善弘、岡本光浩、高見一正：“高度ＩＮにおける統合サービス生成環境構成法”，信学技報SSE94-173 、(1995-01)．

(37) 杉山武史、大宮知巳、高見一正、江崎修司：“パーソナル通信に適用可能なＩＮにおけるアウトチャネルインタラクションのモデリングとプロトコル”，信学技報SSE95-109 、(1995-11)．

(38) 佐野直美、杉山武志、高見一正、牛嶌一朗：“高度ＩＮにおけるカスタマコントロール提供法の検討”，信学技報SSE96-75、(1996-09)．

(39) 坂井　博、高見一正、新津善弘：“高度ＩＮにおける仕様補完技術を用いたＳＬＰ生成法”，信学技報SSE96-103 、(1996-10)．

