

分散処理型通信ソフトウェアのための仕様化方式

西園 敏弘 竹中 豊文

ATR通信システム研究所

高度通信サービスを柔軟に設計するための仕様化方式を提案する。この方式の基本的考え方は、各プロセスの仕様を機能部と相互作用部に分離し、機能部を他プロセスの仕様から独立とすることにある。設計者は、相互作用部の仕様化のみに専念できるので、分散した機能群の協調処理による通信サービスが従来方式に比べ、より簡単に設計できる。本方式による機能部の再利用性は、協調するプロセス数が多い高度通信サービスの仕様化において、特に有効である。また、相互作用部の設計を簡易化するために、並列処理と仕様の可視化を特長とする設計支援環境を提案する。コールウェイティングと複合サービスの例を用いて、本方式の利点を示す。

Specification System for Distributed Processing Communication Software

Toshihiro Nishizono and Toyofumi Takenaka

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

A specification system for flexibly designing advanced telecommunication services is proposed. The essential idea in the system is separation of the functional and interaction parts of each process to obtain functional part independence from other processes. Because designers can concentrate their design efforts on the interaction part of the processes, advanced services accomplished by cooperation of multiple distributed service functions are more easily designed than the conventional way. This capability is effective when the number of cooperative processes is large. To facilitate specifying interaction parts, a design support environment employing parallel processing and specification visualization is proposed. Using call waiting and composite service examples, the advantages of the specification system are illustrated.

1. まえがき

情報化社会の進展に伴い、通信サービスに対する要望は益々複雑化・高度化しつつある。通信サービスは、一般にネットワーク内に分散配置されたサービス機能群の協調処理により成される。例えば、図1に示すように、ひとつの高度サービスは、複数の基本呼機能と付加サービス機能の協調処理として実現される。即ち、通信ソフトウェアの仕様は、全体のサービスを実現するために必要とされるサービス機能を実行する複数プロセスと、それらの間の相互作用(interaction)の形でモデル化される。

将来のネットワークでは、複数の端末が同時に取り扱われ、それらの間の通信において、複数のサービス機能が有機的に組み合わせられたサービスが提供されていくことが予想される。従って、プロセス間の相互作用が増加し、より複雑となっていくことが推測される。このようなサービスを提供する通信ソフトウェア仕様を簡易化するためには、分散環境下にあるプロセス間の相互作用に適合した新しい仕様化方式が望まれる。

新サービス設計の立場からは、その生産性向上も仕様化方式に対する重要な要件である。複数サービス機能の協調による新サービスを速やかに実現するためには、サービス機能仕様の再利用性向上が求められる。また、ラピッドプロトタイピング技術として、仕様全体の確認のためのサービス実行過程の可視化と、新しい機能の仕様化支援のための各プロセスの論理構造の可視化も重要な効用として挙げられる。

分散型通信ソフトウェアの仕様化におけるこれらの要求条件を満たすために、本資料は、プロセス仕

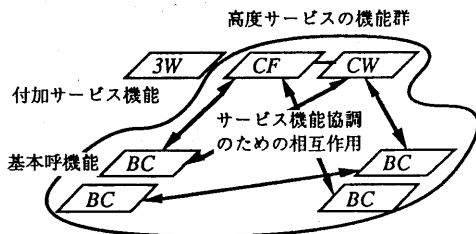


図1 高度サービスにおける機能の協調

様を機能部と相互作用部に分割する並列合成ソフトウェア構造 (Parallel Composing Software Architecture、以下PCS構造と呼ぶ)⁽¹⁾に基づく仕様化方式を提案する。まず、PCS構造の基本概念を概説した後、その構造に基づく仕様を実行するために必要な機構を明確化する。次に、コールウェイティングと複合サービスを例に、PCS構造に基づくサービス仕様とその有効性を示す。さらに、設計支援環境の構成と機能を提案する。最後に、本仕様化方式を基本として、新サービスのユーザプログラマビリティに向けた展開を議論する。

2. PCS構造とその実行機構

通信ソフトウェアの仕様では、ユーザからの入力イベントが複数のプロセスに係わるので、プロセス間の相互作用が必要となる。例えば、図2に示すように、通常の基本呼サービス (Plain Old Telephony Service、POTS) においては、発端末からの発呼イベント (Setup) に対して、着端末の状態が IDLEかACTIVEかに応じて、呼び出し音かビジー音かの何れかを応答する必要がある。このため発端末側の基本呼プロセス (BC-b) は、着端末側の基本呼プロセス (BC-a) に状態問い合わせの意味も含めた呼設定要求信号 (setupInd) を送信し、その応答結果に基づき上述の何れかをユーザに出力する。この場合、各プロセスの仕様は、機能部と相互作用部の2つの部分からなると見做せる⁽²⁾。機能部は、各プロセスの状態に基づき所要の機能を行うものであり⁽³⁾、相互作用部は、入力イベントに対する応答出力を決めるための状態問い合わせ等、プロセス間の情報交換を行うものである。

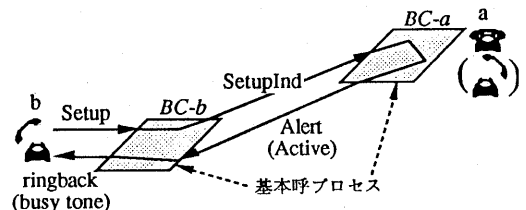


図2 従来構造によるPOTS仕様の例

従来のソフトウェア構造では、各プロセスの仕様の中に機能部と相互作用部が混在していた（図3（a））。入力イベントを契機として、各プロセスは、逐次的にその機能を実行する（図2）。入力イベントに対する出力動作は、このような機能と相互作用の逐次的実行により定められていた。

上述のソフトウェア構造と比べ、PCS構造では、プロセスを機能部と相互作用部に分割する（図3（b））。図4のように相互作用部は、まず、入力イベントに関連するプロセスの機能部に分配する。各機能部は、要求されるサービスに部分的に寄与する部分動作を行う。サービスに係わる出力動作は、再び相互作用部により、各プロセスの機能部が並列かつ独立に行なった部分動作の重畳により定まる。

図5に示すように、PCS構造の仕様を実行するための動作重畳機構は、以下の3機能を持つ。

(1) 入力分解

入力イベントをイベント要素に分解し、関連するプロセスに分配する。

(2) 出力重畳

各プロセスからの部分動作の待ち合わせ（同期）を行ない、それらを出力動作に変換する。

(3) 方向付け（図5（c）下線部）

関連プロセスアドレスの識別のための相互作用情報を用いて、入力イベント送信元のチェックと出力動作の送信先決定を行う。

図5（c）に示すように、プロセスBC-bの相互作用部は、端末bからパラメータDA（宛先アドレス、BC-a）を伴ったSetupイベントを受信すると、そのイベントを2つのイベント要素（SetupおよびSetupInd）に分解し、自らの機能部（fp）とプロセスBC-aに分配する。これと同時に、宛先アドレスを相互作用情報@calleeに記憶し、部分動作（AlertまたはActive）の送信元の識別に備える。この後、部分動作の待ち合わせに入る。自らの機能部の部分動作SetupAckとプロセスBC-a（相互作用部情報@calleeの値）からのAlertを受信した場合は、端末に呼び出し音を出力し、それ以外の場合はビジー音を出力する。

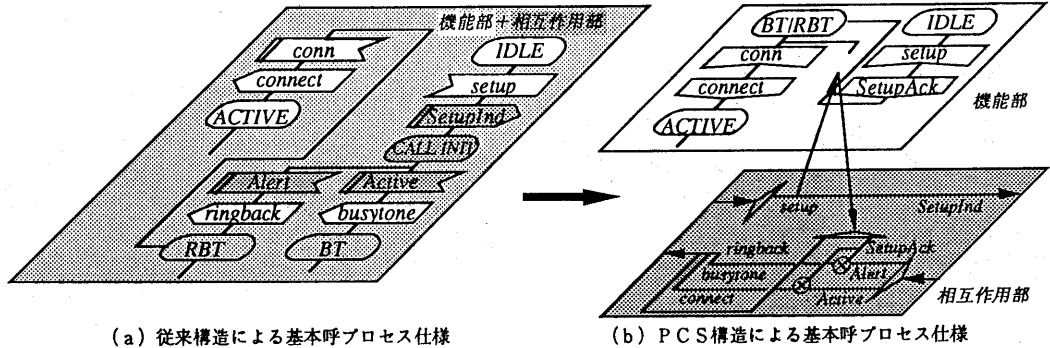


図3 従来構造とPCS構造の比較

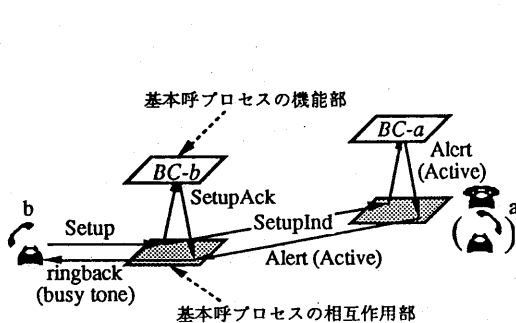
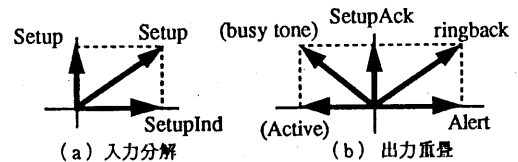


図4 PCS構造によるPOTS仕様例



Setup [item][DA] → Setup [fp] ∧ SetupInd [DA] ∧ @callee := DA
 SetupAck [fp] ∧ Alert [@callee] → ringback [item]
 SetupAck [fp] ∧ Active [@callee] → busytone [item]

Event[source][parameter] → output[destination]
 @callee: interaction information named callee

(c) 方向付けと仕様記述言語

図5 動作重畳機構の機能と仕様記述言語

3. PCS構造に基づくサービス仕様

従来のソフトウェア構造によるコールウェイトイングサービス仕様の例を図6に示す。図2と比べると、コールウェイトイングプロセス(CW)が、このサービスを制御するために追加されている。基本呼プロセスBC-aの仕様は、プロセスCWとの相互作用のために変更されている。変更内容は、プロセスCWからの応答を待ち合わせるための状態、その応答に対応した出力動作に伴う遷移、およびプロセスCWに関連するイベントを転送するための相互作用情報の追加である。プロセスBC-aの元の仕様は、全般的に、上述の変更の影響を受けることになる。更に、既存のサービス機能群を組み合わせた複合サービスを新たに提供するためには、各サービスプロセスの仕様もまた、他のサービスプロセスとの相互作用のために変更する必要がある。

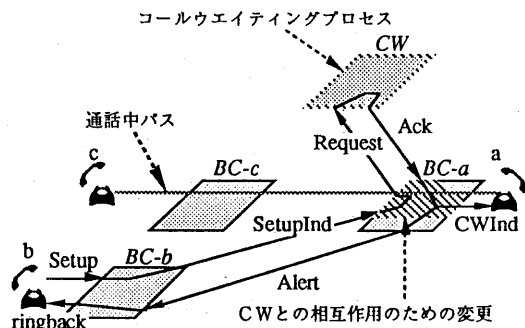


図6 従来構造によるコールウェイトイング仕様例

図7に示すように、既存プロセス(A、B)の変更を避けるためには、新しいサービスプロセス(A+B)を用いることとなる。しかし、この方法は、サービスプロセス数の増大という問題がある。

一方、PCS構造においては(図8)、機能部が、他プロセスと直接の相互作用を行なわないので、上述の変更はプロセスBC-aの相互作用部に閉じたものとなる。状態、遷移および相互作用情報の追加は、動作重畳機構の機能により仕様化できる。従って、機能部は他サービスプロセスから独立となる。更に、図9に示すように、機能(A、B)を組み合わせた複合サービスは、3プロセス(A、BおよびBC-a)の相互作用として仕様化できる。

このように、機能部は、簡単に相互の組み合わせができる。これは、機能部が、他プロセスに影響される相互作用部から分離されていることに起因する。従って、各サービスプロセスの機能部は、再利用可能な部品としてモジュール化できる。

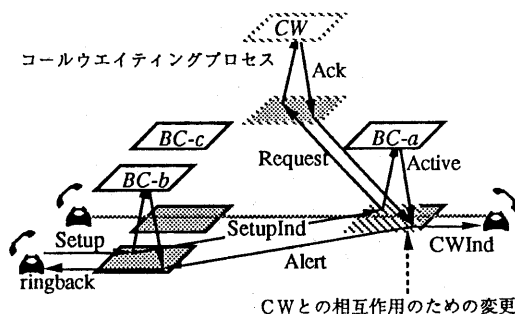


図8 PCS構造によるコールウェイトイング仕様例

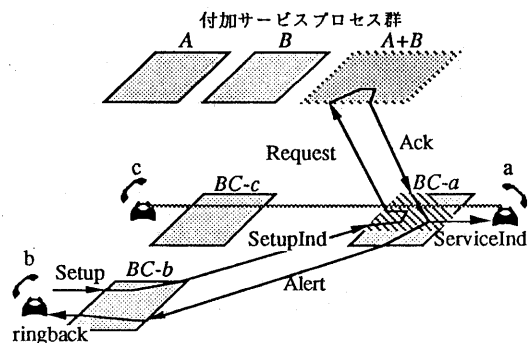


図7 従来構造による複合サービス仕様例

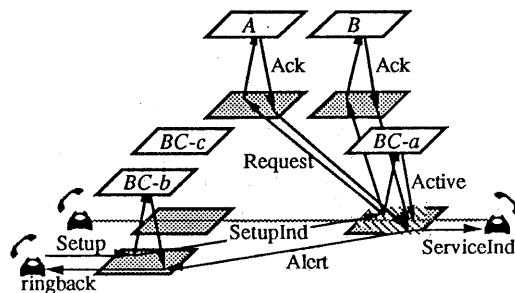


図9 PCS構造による複合サービス仕様例

新しい通信サービスをサービスを柔軟に提供する
方法としてインテリジェントネットワークが提案さ
れ、研究されている⁽⁵⁾。その基本的構成は、ネッ
トワーク内にサービス制御ポイント (SCP) と呼
ぶコンピュータとデータベースのノードを設け、そ
こに置いたサービス論理により、交換機を制御する
ものである⁽⁶⁾。この場合、SCP-交換機間は通
信バスの設定、端末状態変化のモニタ等の交換処理
基本動作で定義した Functional Component (FC)
と呼ばれる標準インタフェースで制御のやりとりを
行う。

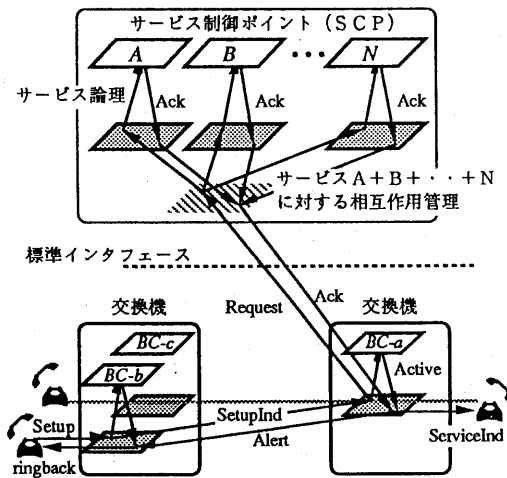


図10 インテリジェントネットワークへのPCS構造の適用

従来のソフトウェア構造では、既存サービス機能
を組み合わせただけのものであっても、図7と同様にサー
ビス毎に新しくサービス論理を記述する必要がある。
即ち、FCの時系列的組み合わせとその実行結果に
よる条件分岐で記述したサービス論理を各サービス
毎に用意することになる。

PCS構造をインテリジェントネットワークに適
用した例を図10に示す。この例では、サービス機
能(A、B、・・・、N)を組み合わせるための相互
作用管理を基本呼プロセスBC-aから分離し、標準イ
ンタフェースFCを通してBC-aと相互作用させるも
のである。PCS構造を用いたことによる最大の利
点は、複合サービスの仕様化において、FCを組み
合わせたサービス論理の再利用ができる点にある。
新たな複合サービスは、相互作用管理によるサー
ビス論理の並列的かつ協調的実行の形で仕様化する
ことができる。このような、サービス機能の再利用性
は、組み合わせるサービス機能の数が多い高度サー
ビスの仕様化において、特に有効である。

4. 設計支援環境

現在、PCS構造に基づく仕様の設計と確認を行
なうために、図11に示す支援環境の開発を進めて
いる。この支援環境は、PCS構造の仕様実行のた
めのマルチワークステーションを用いた並列処理と
設計活動支援のための仕様の可視化を特長とする。
これらは、以下に示すシステムにより実現される。

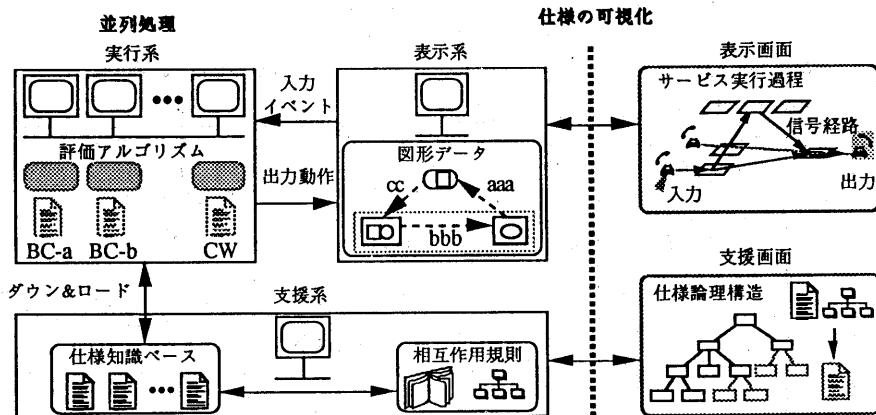


図11 設計支援環境の構成

(1) 実行系

このシステムでは、各ワークステーションは、ネットワーク内に分散配置された各プロセスの動作を行う。各ワークステーションに置かれた評価アルゴリズムは、図5(c)に例示した言語で記述された仕様知識を解釈することにより、入力イベントから出力動作を得るものである。仕様知識は、評価アルゴリズムから分離されているので、新サービス設計において、相互作用部は、容易に仕様化・変更できる。各機能部の仕様部品は、ここで仕様化された相互作用部を通して相互に組み合わせることができる。

(2) 表示系

このシステムは、入力イベントと出力動作による全体のサービスイメージおよびプロセス間の信号経路による協調処理の表示を通して、処理系に対する図的マンマシンインタフェースを提供する。基本的には、表示系の図形データをもとに、表示ウィンドウにおける入力イベントの識別と各プロセスの出力動作による図形上の表面処理を施すことにより表示される。このインタフェースにより、ユーザは要求したサービスの理解が可能となり、設計者は、サービス仕様の確認ができる。

(3) 支援系

このシステムは、相互作用部の変更にあたっての適用規則を用いた設計支援を行なう。例えば、複合サービスを設計する場合、設計者は、既存サービスプロセスの機能部仕様を再利用し、それらのプロセス間に新しい信号を追加することにより、相互作用部を変更することになる。

この相互作用規則は、新しい信号に対応した部分動作重畳の組み合わせを提示するためのものである。この時、知識ベース中の既存仕様を含めて、サービスを行うための機能部と相互作用部の論理構造を可視化することにより、本支援がより有効となる。

5. ユーザプログラマビリティへの方法論

効率的な新サービス提供のためには、ユーザの視点からのサービス定義を可能とするユーザプログラマビリティが必須である。このための基本的方法論は、ユーザが記述したサービス要求仕様を実行可能な設計仕様に変換することになる。

図12に示すように、要求仕様は、全体のサービス定義とサービス間の関係からなるサービスの大局的記述により定義される。全体のサービス定義は、サービス開始から終了までに、ユーザにより送受される全信号の時間的順序関係を記述する。また、この定義では、通信バスの接続状態や着端末の呼び出し状態などのユーザが認識できるサービス状態も併せて記述することとなる。サービス間の関係は、サービスの両立性等のサービス間の相互関係を規定する。この関係では、例えば、コールウェイティング要求信号は、第三者からの呼設定信号が契機になる等、各サービスで用いられる信号の概念的関係の記述も必要である。このようなユーザ要求を仕様化するため、ユーザに直接関係するサービス状態を示す大局的プリミティブを用いたサービス定義法⁽⁶⁾と、E-R (Entity Relationship) モデルを基本とするサービスおよび信号間の概念構造の整理⁽⁷⁾の研究を進めている。

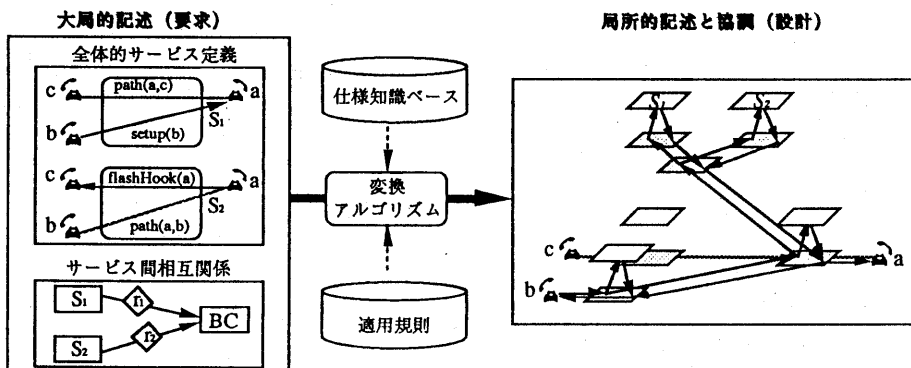


図12 ユーザプログラマビリティへの展開

大局的記述は、P C S構造に基づき、局所的記述である各プロセスの機能とその間の協調処理の仕様に交換され、本資料で提案した設計支援環境により実行・確認される。変換アルゴリズムは、仕様知識と適用規則を用いて、全体のサービス定義を各プロセスにおける局所的仕様に交換し、サービス間の関係をプロセス間の相互作用に変換する。この変換を実現するために、既存の通信ソフトウェア仕様の分析を進めている。この分析における基本的方針は、他プロセスとの相互作用に係わるイベントと状態を抽出することである⁽⁸⁾。この分析においては、仕様知識と相互作用規則を一般化するために、様々なサービス仕様中の機能に関連する信号の分類も必要である。

6. あとがき

本資料は、各プロセスの仕様を機能部と相互作用部に分ける並列合成ソフトウェア構造(P C S構造)に基づく仕様化方式を提案した。また、コールウェイトティングサービスと複合サービスを例に、本仕様化方式の有効性を示した。これらの例では、各プロセス仕様の機能部を変更することなしにサービスの仕様化が可能であった。各プロセスは、自由に他プロセスと協調できるので、本仕様化方式は、複数サービス機能を同時に結合させた新サービス仕様を効率的に設計できる。本資料は、さらに、P C S構造の仕様を実行するための並列処理機能と、ラビッドプロトタイピングによる新サービス設計効率化のための仕様の可視化機能を提供する設計支援環境を提案した。この環境では、仕様知識と相互作用規則を用いて、新サービスを提供するための相互作用部を容易に仕様化できる。

ユーザによる新サービスの定義を可能とするには、ユーザの視点からの大局的サービス記述が必要である。本資料では、ユーザプログラマビリティを実現するためのサービス生成方法論についてふれた。その基本的考え方は、全体のサービスとサービス間の関係を、プロセス仕様の機能部と相互作用部の夫々に交換することにある。

現在のところ、マルチワークステーション上にP C S構造仕様の実行系と仕様実行経過を視覚化する表示系の試作が完了した。今後は、新サービス設計支援機能とユーザ要求からP C S構造仕様への変換法を明確化するために、種々の通信サービス仕様分析を進める予定である。

謝辞

本研究を進める上で、ご指導と励ましをいただいた国際電気通信基礎技術研究所葉原耕平副社長に深く感謝いたします。また、有益な議論を頂いたA T R通信システム研究所の皆様にも感謝します。

参考文献

- (1) Nishizono, T., Takekaka, T. and Monden, M.: "Parallel Composing Software Architecture for Advanced Telecommunication Services", GLOBECOM '89, Dallas, 13.2 (1989).
- (2) Manna, Z. and Wolper, P.: "Synthesis of Communicating Process from Temporal Logic Specification", ACM TOPLAS, 6, 1, pp.68-93 (1984).
- (3) Kawashima, H., Futami, K. and Kano, S.: "Functional Specification of Call Processing by State Transition Diagram", IEEE Trans. Commun., COM 19, 5 (1971).
- (4) Gilmour, J. and Gove, R. D.: "Intelligent Network/2: -the Architecture -The Technical Challenges -The Opportunities", IEEE Communications Magazine, 12, pp.8-11 (1988).
- (5) Homayoon, S. and Singh, H.: "Methods of Addressing the Interactions of Intelligent Network Services with Embedded Switch Services", IEEE Communications Magazine, 12, pp.42-46 (1988).
- (6) Hirakawa, Y., Harada, Y. and Takenaka, T.: "Behavior Description for a System Which Consist of Infinite number of Processes", BILCON '90, Turkey (1990) (to be published)
- (7) Okamoto, K. and Hashimoto, M.: "Program Synthesis by Specification Parts Technique Using A Constraint-based Conceptual Model", SS89-18 (1989).
- (8) 元治、西園、竹中: "分散型通信ソフトウェアの図的表現法"、第40回情処全大(1990)