

〔公 開〕

TR-C-0129

対象システムに非依存な  
ソフトウェア生成と  
異種プロトコルへの整合

近藤 良樹  
Yoshiki KONDO

1 9 9 6 2 . 2 9

ATR通信システム研究所

対象システムに非依存なソフトウェア生成  
と異種プロトコルへの整合

近藤 良樹  
Yoshiki KONDO

1996.2.29

ATR 通信システム研究所

## もくじ

1	はじめに	1
2	自動生成ソフトウェア	1
2.1	概要 SDL	1
2.2	詳細 SDL	1
2.3	論理インタフェース	2
3	ソフトウェア構成	2
3.1	対象とするソフトウェア構成	2
4	インタフェース変換部	3
4.1	論理インタフェースの設定	3
4.2	概要 SDL と論理インタフェースの関係	6
4.3	詳細記述言語	6
4.4	詳細記述言語の拡張	6
4.5	インタフェース変換	7
4.5.1	変換部作成の自動化	7
4.6	ネットワーク制御タスク	10
5	評価	10
6	まとめ	11
7	今後の課題	11
A	付録	i
A.1	PBX における動作確認	i
A.1.1	機能構成	i
A.1.2	プログラム容量	ii

## 1 はじめに

端末の動作により記述された通信サービス仕様から1端末1プロセスとして動作可能となるソフトウェア仕様(概要SDL)が生成される。このソフトウェア仕様は端末の動作により記述されているため、ネットワーク内部を制御するタスクが存在しない。そのためソフトウェア仕様にタスクを挿入する必要がある。このとき、ソフトウェア仕様をシステムや端末のプロトコルに依存しないサービス制御ソフトウェア(詳細SDL)として生成するために、タスクを論理的な制御インタフェース(論理インタフェース)として提供する。

この論理インタフェースに基づいて自動生成したソフトウェアを対象のシステムで動作可能とするために、論理インタフェースからシステムが提供する制御インタフェースに変換するインタフェース変換部を設ける。

自動生成ソフトウェアを動作可能とするために、自動生成ソフトウェアとインタフェース変換部の2階層のソフトウェア構成を提案する。

以下、2章では自動生成ソフトウェアの概要について、3章で対象とするソフトウェア構成について述べ、4章でインタフェース変換部を作成するアルゴリズムについて述べる。5章ではインタフェース変換部のアルゴリズムをアナログ加入者線信号方式とデジタル加入者線信号方式に適用させ、その評価を示す。

## 2 自動生成ソフトウェア

自動生成ソフトウェアは非専門家が記述するサービス仕様と専門家が記述する詳細仕様の2つの仕様記述から作成される。

まず、非専門家が記述したサービス仕様から概要SDLが生成され、概要SDLに専門家が記述した詳細仕様によりプログラム仕様である詳細SDLが生成される。以下に概要SDL、詳細SDLについて説明する。

### 2.1 概要SDL

非専門家が端末の動作だけでネットワーク内部の処理をブラックボックスとしてサービス仕様を記述する。このとき、サービス仕様記述言語に自然言語(日本語)を用い、STR(State Transition Rule)という形式言語に変換される。STRから1端末1プロセスとして動作可能となるプロトコルを自動的に合成する。この合成の結果、概要SDLが生成される。概要SDLではSDLの構成要素である(1)始状態(2)入力信号(3)条件比較(4)送信信号(5)端末制御タスク(6)ネットワーク制御タスク(7)次状態のうち、端末制御タスクとネットワーク制御タスクが存在しない。これは、ネットワークをブラックボックスとしているためである。また、ネットワーク内部の処理は専門家の知識が必要である。このような理由により、不足しているSDL要素を補足するために、専門家がネットワークの内部の仕様を記述し、概要SDLに挿入して完全なSDLを生成する。

### 2.2 詳細SDL

概要SDLにネットワーク内部の制御を記述されているものが詳細SDLである。詳細SDLは概要SDLに専門家がネットワーク内部の詳細仕様をSTR/D言語を使って記述したもの

から自動生成される。このときのネットワーク内部処理を制御部品としてまとめ、これらはインタフェースとして提供する。

詳細 SDL の要求条件は、動作させる対象のシステムに依存しないものであること。従って、提供するインタフェースはシステムに依存しない論理的なインタフェース (以下、論理インタフェースと呼ぶ) である。この論理インタフェースに基づいて詳細 SDL が生成される。

### 2.3 論理インタフェース

論理インタフェースにはシステム非依存なインタフェースという意味の他に、次のような意味をも含ませることを試みた。

サービス仕様は非専門家がある種類の端末をイメージして端末の動作で記述される。たとえば最も端末動作として分かりやすいアナログ電話機がイメージされる。この端末をイメージして仕様を記述することは、非専門家は無意識のうち特定のプロトコルについて仕様を記述していると言える。このようなサービス仕様から自動生成したソフトウェアで、異なるプロトコルをもつ端末でも動作させたいという要求が起こる。この場合に、非専門家に端末毎に仕様を記述させることは避けたい。逆にプロトコルを知っている非専門家は少ない。そのため仕様記述はできない。従って、ある種類について記述された仕様から自動生成したソフトウェア 1 つだけで別の種類の端末も動作可能とすることを試みる。つまり端末制御タスクに関する論理インタフェースは端末毎にインタフェースを持つのではなく、共通のインタフェースとする必要がある。このように論理インタフェースはシステムに依存せず、端末の種類 (端末の信号方式) に依存しないという 2 つの意味を持ち合わせる。

## 3 ソフトウェア構成

対象のシステムに依存しないものとして自動生成したソフトウェアを動作させるためのソフトウェア構成を示す。

### 3.1 対象とするソフトウェア構成

一般的なシステム構成は大きく分けてハードウェア、OS、アプリケーションの 3 階層構成になる (図 1 左)。

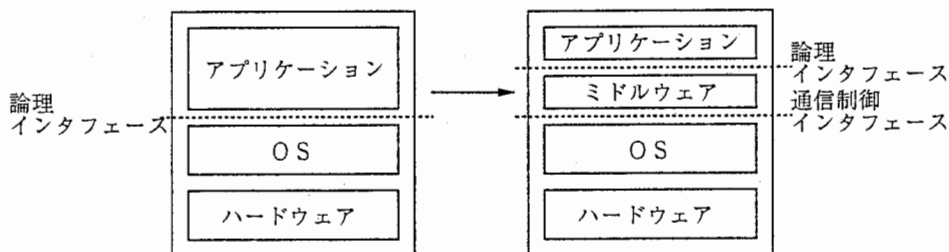


図 1: システム構成

自動生成するソフトウェアがアプリケーション部分 (図 1 左) に相当する。

アプリケーションは OS 上で動作するものである。従って、アプリケーションをシステム非依存にするためには、対象とする OS を CTROM, IROS のような共通 OS と限定すれば自動生成ソフトウェアはシステムに依存せず 1 つで済むことになる。このとき、論理インタフェースは OS が提供するインタフェースとなる。

しかし、OS が提供するインタフェースでは、前述の論理インタフェースの条件である端末種別に依存しない共通な制御インタフェースという条件に合わない場合がある。また、最近の傾向としてベンダ間の競争を激しくし、高性能システムが提供されることを期待して OS 部分はベンダ独自のものとなっている。つまりマルチベンダ指向になっている。従って、ベンダに依存しない共通のアプリケーションとして動作させるためには、ミドルウェアあるいはプラットフォームと呼ばれるものが OS とアプリケーションの間に置かれるようになった(図 1 右)。このときの論理インタフェースはミドルウェアが提供するインタフェースとなり、OS が提供する制御インタフェース(以下通信制御インタフェースと呼ぶ)がある。

つまり、課題は (1) システムと端末種別に依存しない論理インタフェースの提供、(2) 論理インタフェースから通信制御インタフェースに変換するミドルウェア(以下インタフェース変換部と呼ぶ)の構築である。これらの課題に取り組んだ結果について以下にまとめる。

## 4 インタフェース変換部

### 4.1 論理インタフェースの設定

ここでは、論理インタフェースとして提供される中の端末制御に関するインタフェースについて述べる。

インタフェースの要求条件は、システム非依存でかつ端末非依存である。そのため、インタフェース設定に際してインタフェースを一般化するために OSI 参照モデルを参考にする。

OSI 参照モデルでは  $n+1$  層に対して  $n$  層は  $n$  サービスを提供する。このサービスをインタフェースとして提供する。インタフェースの種類として指示、要求、応答、確認の 4 種類あり、これらのインタフェースの関係は図 2 のようになる。信号方式としては、要求信号に対して応答信号を送信する要求・応答型だけでなく、応答信号を返さない通知型もある。

今回対象とする自動生成ソフトウェアを動作させるシステムの階層構成を図 3 に示す。ここで、上で述べた通信制御インタフェースは Layer3 サービスインタフェースに相当する。従って論理インタフェースは Layer4 以上のインタフェースになる。

例えば、OSI 参照モデルにおける Layer4 サービスでは、以下のプリミティブが提供される。

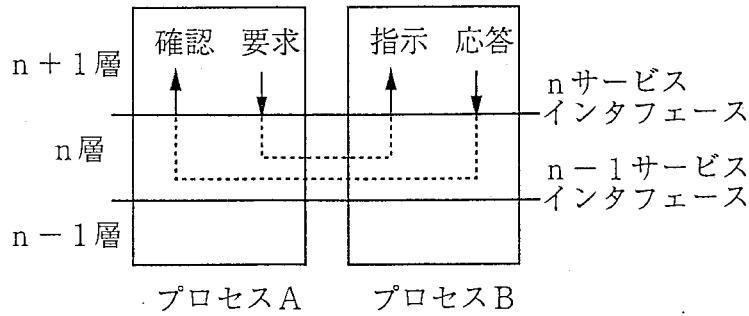


図 2: OSI 参照モデル

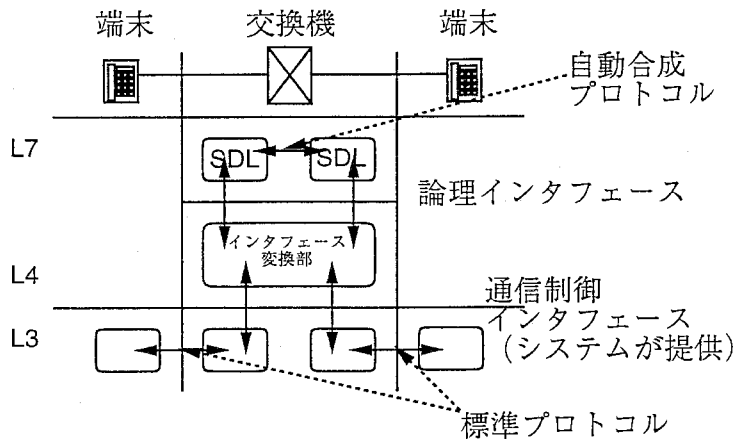


図 3: 自動生成ソフトウェアの階層構成

サービス名	プリミティブ	パラメータ
コネクション確立	T-CONNECT 要求	発アドレス, 着アドレスなど
	T-CONNECT 指示	発アドレス, 着アドレスなど
	T-CONNECT 応答	応答アドレスなど
	T-CONNECT 確認	応答アドレスなど
コネクション解放	T-DISCONNECT 要求	ユーザデータ
	T-DISCONNECT 指示	切断理由, ユーザデータ
データ伝送	T-DATA 要求	ユーザデータ
	T-DATA 指示	ユーザデータ
	T-EXPEDITED-DATA 要求	ユーザデータ
	T-EXPEDITED-DATA 指示	ユーザデータ
コネクションレス型	T-UNITDATA 要求	
	T-UNITDATA 指示	

このプリミティブを電話端末を使った音声通信に適用して論理インターフェースを設定してみる。CONNECT 指示は、端末の動作では受話器を上げる (offhook) になる。CONNECT

要求は相手端末への呼出信号になる。CONNECT 確認は着信端末が応答 (offhook) したときになる。CONNECT 応答は発信端末への接続信号になる。DISCONNECT 指示は端末が受話器を戻す (onhook) になり、DISCONNECT 要求は相手が切断したことを告げる信号となる (図 4実線)。以上の基本信号を実現するためには、各信号に必要なパラメータを取得する必要がある。次に、パラメータ取得のための信号手順を例にとって説明する。

まず、基本サービスを行うときの相手アドレス (相手電話番号) を取得するため、または付加サービス、高機能サービスを行うかのサービス選択を取得するために、サービス選択要求信号を端末に対して送る。それに対して端末は選択数字として返送する。また、相手アドレスが決定し、CONNECT 要求を送信したあと、相手端末が呼出状態になったかどうかを告げるための信号が発信端末に送信される。さらに、コネクションが完全に解放されたかどうかの確認のための信号などが追加される。(図 4点線)。

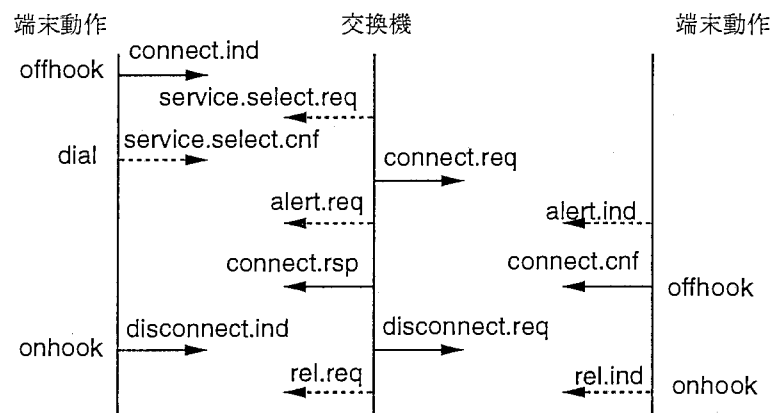


図 4: OSI プリミティブと論理インタフェースへの対応

以上のような観点から論理インタフェースをまとめたのが表 1である。音声ガイダンスや PB 音受信などの端末操作は端末制御インタフェースとして設定せず、ネットワーク制御インタフェースとして扱う。

表 1: 端末制御インタフェース一覧

コネクション確立	接続指示 選択要求	接続確認 選択確認	接続応答	接続要求
コネクション解放	切断指示	切断要求	解放指示	解放要求
呼出	呼出指示	呼出要求		
保留	保留要求	保留確認	保留拒否	
保留解除	保留解除要求	保留解除確認	保留解除拒否	
サービス活性	サービス活性指示	サービス活性応答		



## 4.2 概要 SDL と論理インタフェースの関係

ここでは、概要 SDL と論理インタフェースの関係について考察する。概要 SDL は先に述べたように入力信号は構成要素として既に存在する。この入力信号はサービス仕様にある端末からのイベントとして記述される。

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

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

(規則 3) dial-tone(A) uptreq(A): ident(A).

上の規則の offhook, dial, uptreq が入力信号となる。offhook が論理インタフェースの接続指示に対応し, dial, uptreq が選択確認に対応する。dial, uptreq から分かるように設定した論理インタフェースは抽象的な選択確認としているが、実際の信号は仕様記述に依存してしまふ。dial, uptreq と区別がつくのは電話機からダイヤル数字を幾つか受信し、数字分析を行って信号名が決定した結果である。従って、インタフェース変換部で数字分析処理を行うことで信号名を決定する。

逆に、設定した論理インタフェースが概要 SDL で表現される信号と対応しない場合もある。サービス仕様はネットワークの外部から認識できる端末の動作で記述されるため端末の変化が認識できる範囲の信号は論理インタフェースと対応付く。しかし、ネットワーク内部でやりとりされる信号、例えば呼出指示などは内部的な信号のためこの論理インタフェースに対応する入力信号がない。従って、対応付かなかった論理インタフェースの入力信号を受信するように、概要 SDL に信号受信を挿入する必要がある。

一方、出力信号はもともと概要 SDL には存在していないため、信号送信を挿入する必要がある。これら不足している信号を概要 SDL に挿入するために詳細記述言語 STR/D を用いる。

## 4.3 詳細記述言語

詳細記述言語は概要 SDL で表現される特定の位置にタスクを挿入する規則を記述する。記述規則は以下の通りで、STR/D の記述対象者は専門家である。

位置指定 { タスク指定 };

出力信号は信号を送信するタスクとして提供される。この STR/D 規則では、出力信号を送信するタスクをタスク指定に記述することで概要 SDL に挿入することが出来る。

対応付かない入力信号受信を概要 SDL に挿入するには、次の問題がある。(1) 入力信号を受信する状態が既に存在する場合はその状態が位置指定される。しかし、存在しない場合は状態を新たに生成する必要がある。(2) 入力信号を受信した結果、次に遷移する状態、遷移間のタスクを指定する必要がある。これらの問題は、従来の詳細記述言語では対処できない。そのため、従来の詳細記述言語を拡張する。

## 4.4 詳細記述言語の拡張

上で述べた問題を解決するために、新たな状態や入力信号を挿入することができるように仕様記述を次のように拡張する。

位置指定 (生成状態)[ 入力信号 ] { タスク指定 };

位置指定はどの位置に挿入するかを指定するもの。生成状態は、新たに状態を生成したいときに状態名を記述する。入力信号は位置指定にある状態で受信する信号名を記述する。タスク指定は生成状態から次状態へ遷移するときの実行するタスクを記述する。このときのタスクは他プロセスへのプロセス間信号も記述される。

概要SDLと拡張した詳細仕様記述言語により詳細SDLを生成するが、生成するときの課題は次の通りである。

1. 概要SDLで示されるプロセス間通信の信号処理を新状態に追加する必要がある。

生成した新状態で指定した入力信号だけでは、他プロセスからのプロセス間通信の信号を処理できず、送信プロセスがデッドロックを起こしてしまうため。

2. 準正常処理(途中切断など)を新状態に追加する必要がある。

新状態で途中切断処理がないと、自プロセスがデッドロックを起こしてしまうため。

3. 状態管理処理を変更、追加する必要がある。

遷移途中に新たに状態を追加した場合、もとの遷移状態が変わるため。

上記課題を解決することにより、概要SDLから論理インタフェースに基づいた詳細SDLを生成することができる。このときの詳細SDLが目的とするシステム非依存で端末非依存なアプリケーションとなる。

#### 4.5 インタフェース変換

論理インタフェースに基づいて自動生成したソフトウェアを実際のシステムで動作させるためには、論理インタフェースからシステムが提供する通信制御インタフェースに変換するインタフェース変換部が必要となる。

インタフェース変換部で行う処理は、通信制御インタフェースからの信号を受信し、それを詳細SDLで表現される入力信号(論理インタフェース)に変換し、送信する。あるいは、交換機からの出力信号(論理インタフェース)を受信し、通信制御インタフェースに変換することである。

通信制御インタフェースは制御する端末毎に存在することから、インタフェース変換部も端末毎に作成する。

##### 4.5.1 変換部作成の自動化

インタフェース変換部の作成を専門家がすべて作成するのではなく、ある程度の自動化を行い、専門家の作業の削減を目指す。自動化のための作成手順を次に示す。(図はデジタル加入者線信号方式を例にしている)

1. 変換部のプログラム仕様の雛形(状態遷移)を通信制御インタフェースにより作成する。(図5)
2. 信号の変換であることから、通信制御インタフェースによる信号と論理インタフェースによる信号の意味的に同じものを対応つける。(表2)

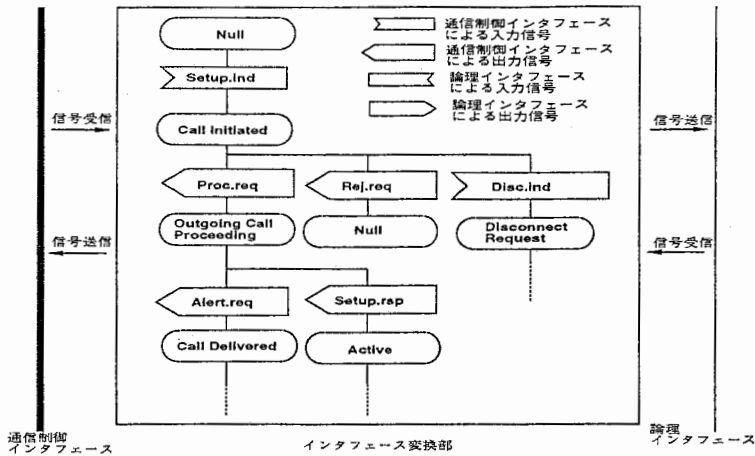


図 5: 通信制御インタフェースによる状態遷移

表 2: インタフェース対応例

通信制御インタフェース	論理インタフェース	意味
stup.ind	offhook	接続指示
alert.req	start-ringbacktone	呼出要求
disc.ind	onhook	切断指示

3. 作成した状態遷移に対応関係にある論理インタフェースによる信号を挿入する。このとき、交換部に論理インタフェースによる信号を受信すれば通信制御インタフェースによる信号を送信し、通信制御インタフェースによる信号を受信すれば論理インタフェースによる信号を送信するといった処理になるよう挿入する。(図6)

4. 対応つかないインタフェースの処理を次に示す。

対応つかないインタフェースのパターンと処理方法を示す。(図7)

(a) 受信信号の論理インタフェースに対する通信制御インタフェースがない

この信号を受信するためには新たに安定状態を作成し、この安定状態で論理インタフェースによる信号を受信して次の安定状態に遷移する処理を追加する。

このとき自動化を行うには挿入位置決定の問題がある。信号手順を与えないでインタフェースだけで受信信号の挿入位置を決定させることを考える。適切な位置に信号を挿入するためには、インタフェースの意味で挿入位置を決定する。インタフェースの意味は OSI 参照モデルにおける 4つの種別に分類される。これらは相互に関係があるため、挿入するインタフェースの前後のインタフェースの意味を参照することにより挿入位置候補が上がってくる。候補の中から適切な位置を指定することにより安定状態と信号を挿入する。

(b) 送信信号の論理インタフェースに対する通信制御インタフェースがない

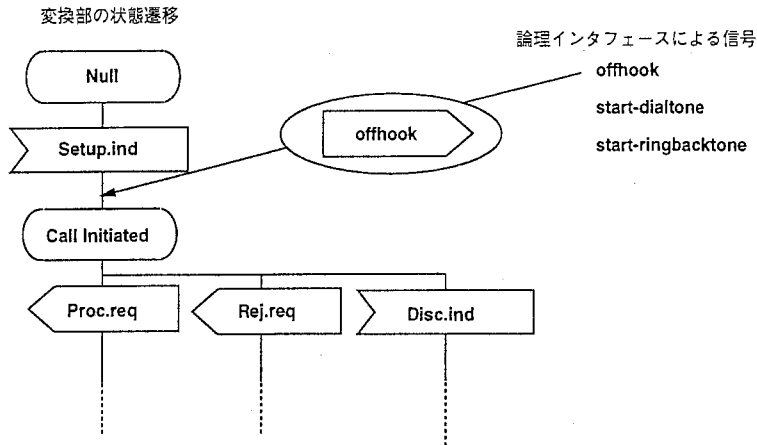
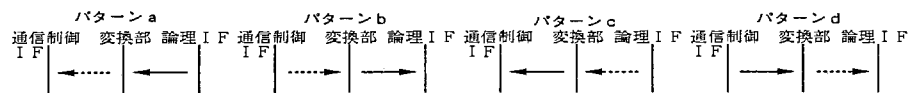


図 6: 論理インターフェースによる信号挿入

信号を送信するため、挿入位置を決定する必要がある。挿入位置決定は、パターン a と同様にインターフェースの意味を参照し、挿入位置候補を上げ、適切な位置を指定することにより送信信号を挿入する。

- (c) 送信通信制御インターフェースに対する論理インターフェースがない  
この信号を送信するためには、論理インターフェースによる信号を受信するための安定状態を削除する必要がある。対応するインターフェースがないため状態遷移はこの信号を送信する直前の安定状態で待ち続けてしまう。従って、このパターンの通信制御インターフェースの直前の安定状態を削除する。
- (d) 受信通信制御インターフェースに対する論理インターフェースがない  
通信制御インターフェースによる受信信号を変換する必要がない信号だと判断し特に処理を施さず、次の安定状態に遷移する。

◎対応つかない4つのパターン



◎対応つかない4つのパターンの処理方法

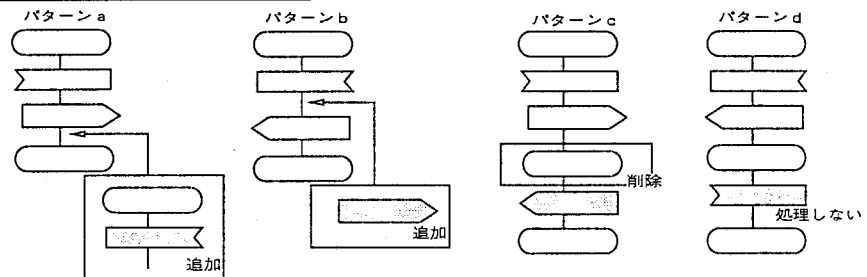


図 7: 対応つかないインターフェースのパターンと処理方法

以上の手順により変換部の状態遷移の信号処理が作成できる。完全な状態遷移とするためには、信号処理だけでなくネットワーク制御処理も必要であり、これらの処理を状態遷移に挿入する必要がある。

#### 4.6 ネットワーク制御タスク

詳細 SDL は信号処理に関する部分だけでなく、交換機内部のネットワークリソース制御に関しても記述される。これらはネットワーク制御タスクまたは制御部品と呼ばれる。これらのタスクは、システムに依存しない論理的なインタフェースとして提供される。ネットワーク制御インタフェースには次のようなものが存在する。

- 音制御 (サービス音源：完了音，不完了音，保留音など)
- ダイヤル数字分析
- 通話路制御
- 特殊リソース制御 (ガイダンス装置制御，PB 受信装置，各種トランク装置)

これらのインタフェースも変換部で論理的なインタフェースから実際のシステムが提供するインタフェースに変換する。

ネットワーク制御タスクで問題となるのは、これらのタスクは特に交換機リソースを制御するために、有限のリソースを制御することになる。従って、この種のタスクは、実行結果が返ってきて、その結果により遷移する状態を変更する必要がある。実行結果により遷移先を指定するのは詳細記述言語 STR/D により行う。

## 5 評価

システムや端末に依存しない論理インタフェースに基づいて自動生成したソフトウェアと論理インタフェースから対象とするシステムが提供するインタフェースに変換するインタフェース変換部の2階層のソフトウェア構成を提案した。

2階層ソフトウェア構成により異なるプロトコルに適用させ動作可能かを評価する。評価に用いたプロトコルはアナログ加入者線信号方式とデジタル加入者線信号方式 (DSS) である。サービス仕様は二者通話 (pots) を用いた。それぞれのインタフェース変換部は提案したアルゴリズムで作成することができる。

表3から次のことが分かる。自動生成したソフトウェアは1つのため論理インタフェースの数は同じである。システムが提供する通信制御インタフェースの数はそれぞれアナログが16、DSSが13で、このとき対応したインタフェースの数がそれぞれ8と10である。アナログよりDSSの方が対応した数が多いことが分かる。これは論理インタフェースがOSI参照モデルに基づいていて、DSSも同じくOSI参照モデルを基にしているためである。つまり、自動生成したソフトウェアが動作可能となるプロトコルの適用範囲はOSI参照モデルを基にして作成されたプロトコルをもつ端末では動作可能であると推測できる。具体的なプロトコルでは、X.25やNo.7信号方式などがある。アナログはOSI参照モデルに基づいていないのに適用可能であった。これはアナログ加入者線信号方式はDSSの一部分と見ることができるためで、不足している信号を追加することにより適用可能であった。

表 3: プロトコル適用例

プロトコル種別	アナログ	DSS
論理インタフェースの数	12	12
通信制御インタフェースの数	16	13
対応したインタフェース数	8	10
パターン a の数	1	1
パターン b の数	3	1
パターン c の数	6	2
パターン d の数	2	1

## 6 まとめ

端末の動作により記述された通信サービス仕様から 1 端末 1 プロセスとして動作可能となるソフトウェア仕様が生産される。このソフトウェア仕様をシステムや端末のプロトコルに依存しないサービス制御ソフトウェアとして生成するために、論理インタフェースを提供した。この論理インタフェースに基づいて自動生成したソフトウェアを対象のシステムで動作可能で、異なるプロトコルに適用可能するためのインタフェース変換部を作成するアルゴリズムを提案した。このアルゴリズムをアナログ加入者線信号方式とデジタル加入者線信号方式に適用させ、1つのサービス制御ソフトウェアで動作可能であることを確認した。

## 7 今後の課題

今回の手法では、OSI 参照モデルに基づいたプロトコルであれば端末と交換機間の信号方式だけでなく、交換機間の信号方式にも適用可能といえる。交換機間の信号方式には OSI 参照モデルに基づかない個別信号線方式などのプロトコルが存在する。これらのプロトコルでは、外部で認識できる端末の状態と自動生成ソフトウェアの状態を一致させることが困難となる。個別信号線方式を例にすると、dial 後、交換機内部の状態は端末とトランクの通話路を接続し、相手交換機からのリングバックトーンあるいはジジートーンを端末に聞かせる。つまり、交換機内部の端末の状態はトランク通話中状態である。しかし、外部から認識できる端末の状態は ringback、あるいは busy のため、状態が一致しない。

これらのプロトコルにも自動生成ソフトウェアで適用可能とするためには、自動生成ソフトウェアの状態を dial-tone の状態からインタフェース変換部により ringback、path 状態へと無条件に状態を遷移させる。つまり、交換機内部の状態は path 状態で、外部から認識できる端末の状態は ringback あるいは busy である。このように遷移させれば任意のプロトコルに適用可能と考えられる。

今後の課題として任意のプロトコルで動作可能とするために、外部から認識できる端末の状態と交換機内部の状態の不一致が起こる。この手法が適切であるか、別の手法を確立するか検討が必要である。

## A 付録

### A.1 PBX における動作確認

自動生成されたソフトウェアを実際のシステムに適用させる。適用サービスは UPT サービスとスクリーニングサービスである。

#### A.1.1 機能構成

システムの構成は PBX と WS。機能配備を IN で用いられる用語を使って説明する。IN で用いられる機能にはサービス制御機能 (SCF)、交換制御機能 (SSF)、特殊リソース制御機能 (SRF)、データベース制御機能 (SDF) がある。

これらの機能は PBX に配備されるが、特に UPT サービスに必要なデータベース (SDF) とガイダンス処理 (SRF)、スクリーニング条件 (SDF) を WS に配備した。

- PBX のプロセス構成

- MAINP

- ハードウェアからのイベントを受信して、呼対応にイベントを分配する。

- TMP

- 加入者データを管理する。呼毎のプロセスも同様に管理する。

- STRP

- システムからの信号を受信してインタフェース変換しサービス制御プログラムに信号を送信する。その逆にサービス制御プログラムから信号を受信してインタフェース変換しシステムへ信号を送信する。また、サービス制御を行う。以上 3 つの機能を 1 つのプロセスで行う。

- TIMERP

- 時間管理を行う。タイムオーバーなど。

- RSCP

- ネットワークリソースを管理する。リソースの空き・塞がりを管理する。また、WS とのリンク状況を管理する。

- WS のプロセス構成

- UPT データベース

- UPT サービスに必要な UPT 番号、暗証番号、ホーム端末、着信端末を管理する。

- スクリーニング条件

- スクリーニングを行う端末、条件などを管理する。

- ガイダンス機能

- 音声を UNIX ツールを使って符号化しデータベース化したものをガイダンスとして再生する。

- 保守機能

通信サービス実行に必要な局データ, UPT データベース, スクリーニング条件などを追加, 変更するためのマシンインタフェース.

A.1.2 プログラム容量

デモシステム (UPT + スクリーニング) における PBX 側の機能別のプログラム容量を示す.

表 4: デモシステムにおけるプログラム容量

プロセス名	ステップ数	バイト数
MAINP	5,234	122,142
STRP	17,087	385,004
(自動生成)	133,442	3,118,182
TIMERP	1,177	24,771
RSCP	2,232	45,046
total	159,172	3,695,145