

二段階通信サービス仕様記述とプログラム仕様への自動変換

田 倉 昭† 太 田 理†

ユーザの要求する通信サービスの早期提供という社会的ニーズに応えるためには、ユーザ（非専門家）による要求記述とそれに基づくプログラム仕様の自動生成が効果的である。本論文では、通信サービス仕様を端末の状態変化を表す規則の集合として記述したサービス仕様とそのサービス仕様を通信システム上で動作させるソフトウェアへの変換法を示し、PBXソフトウェア開発への適用により有効性を示す。本手法では、外部から認識可能な端末の動作を記述する仕様記述言語 STR と STR で記述されたサービス仕様を通信システム上で実現するのに必要な詳細仕様記述言語 STR/D を用いる。これら二段階からなる仕様記述から通信ソフトウェアへの自動変換を行う。個々のベンダの通信システムに依存しないソフトウェア自動生成を可能とするために、論理的な通信システム制御インタフェースを設けた。本インタフェースを用いることにより、ベンダが異なる二つの PBX 上で走行するソフトウェアが自動生成できることを確認した。提案する二段階仕様記述法を用いることにより、通信システムに関する非専門家による新サービスの定義、およびそのソフトウェアの自動生成が可能となる。

Two-Layered Communications Service Specification Descriptions and Program Specification Generation

AKIRA TAKURA † and TADASHI OHTA †

A two-layered communications service specification description method is proposed as well as a communications software generation method from service specifications. In this method, a declarative specification language of observable terminal behaviors, STR, and supplementary specifications to implement STR specifications, STR/D, are used, and they are automatically transformed into a communications software. The generated software is used for controlling services on a PBX. A logical interface for controlling a communications system is introduced to describe STR/D specifications which are independent of both individual service specifications described with STR and a target communications system. The application results show that the proposed method is feasible and useful.

1. はじめに

通信サービスに対する要求として、種々のサービスを迅速に提供する必要性が高まっている¹⁾。このような要求に対して、通信システムの専門家がユーザの要求を聞いて個別に対応する方法では、十分に与えられない状況になっている。このため、通信システムの非専門家によるサービス仕様の記述の必要性が生じつつある。

サービス仕様を正確に記述するためには形式的仕様記述言語が必要となる。通信サービスに対する仕様記述言語として、SDL²⁾や LOTOS^{3)~5)}など多くの提案がある。しかし、これらの言語を用いて通信サービス仕様を記述する場合、サービスを実現するためのソフトウェア構造やプロセス間の同期をとるための仕様など

通信システムや通信ソフトウェアに関する専門知識が必要となる。このため、これらの仕様記述言語は非専門家には使えない。

このため、ATR ではネットワーク内部をブラックボックスとみなし、端末の動作を記述することによりサービス仕様を記述する言語 STR (State Transition Rules)⁶⁾を開発した。STR を用いることにより、通信システムや通信ソフトウェアの詳細知識を持たない通信システムの非専門家でも通信サービス仕様を記述することができる。STR で記述された通信サービス仕様は端末で起こったイベントに対して、端末がどのように状態遷移するかという通信システムのユーザから認識可能な仕様だけしか記述しない。すなわち、従来の仕様記述言語において記述していた信号送受などの詳細仕様を記述することはない。このため、STR で記述された通信サービス仕様からその仕様を実現するソフトウェアを生成するためには、上記の詳細仕様をどの

† ATR 通信システム研究所

ATR Communication Systems Research Laboratories

ように補うかが課題となる。

本論文では、通信サービス仕様を端末から見た要求仕様として STR を用いて非専門家が記述し、通信システムを制御するための詳細仕様を知識として詳細仕様記述言語 STR/D を用いて専門家が記述する二段階の仕様記述法を提案する。このように、サービス仕様とそれを実現するための通信システム制御仕様を別々に記述する方式を実現するためには、これら二段階の仕様記述からプログラム仕様への変換と通信システム制御仕様を個別の通信サービスや通信システムに依存しない形の知識として用意することが課題である。

STR で記述されたサービス仕様からそれを実現する状態遷移形式のプログラム仕様を自動生成する方式²⁾を利用することにより、STR と STR/D の二段階仕様記述から、最終的なプログラムに対する仕様を導出することができる。生成されるプログラム仕様を表現するのに SDL²⁾を用いる。SDL から各種プログラム言語で記述されるプログラムへは容易に変換することができる。STR では、端末の状態を表すための状態記述要素と端末で起こるイベントを基本要素としてサービス仕様を記述する。これらの基本要素だけに依存する形で STR/D 記述を行うことにより、個別の通信サービスに依存しない形で通信システム制御仕様を記述することが可能である。また、仮想的な通信システムを仮定した論理的な通信システム制御インタフェースを提供することにより、通信システム制御仕様を個々の通信システムに依存しないようにすることができる。以上の方式を用いることにより、STR 記述において既存の状態記述要素とイベントを用いて記述できるサービスについては、STR/D で記述された通信システム制御仕様を知識として再利用することができる。新たな状態記述要素やイベントを使って記述されたサービス仕様を実現するのに必要な通信システム制御仕様は専門家が追加記述する。

提案する手法を用いて自動生成されたプログラムを二つの異なる PBX 上で走行させ、手法の確認実験を行った。その結果、本手法により自動生成されたプログラムが十分実用的な性能を持つことを確認した。

以下、2章では非専門家を対象とした二段階仕様記述法を述べ、3章でプログラム仕様への自動変換法を示す。4章、5章で、PBX サービスへの適用法およびその評価を示す。

2. 二段階仕様記述

2.1 サービス仕様記述言語 STR

STR では、通信サービス仕様を、通信サービスの進

行にもなって外部から認識可能な端末の状態がどう変化するかを宣言型規則の集合として記述する。一つの STR 規則は次の形で書かれる。

$$p_1, \dots, p_m \quad e : \quad q_1, \dots, q_n.$$

上の規則において、 p_1, \dots, p_m を現状態、 e をイベント、 q_1, \dots, q_n を次状態と呼ぶ。現状態、次状態の構成要素である $p_i (1 \leq i \leq m)$ 、 $q_j (1 \leq j \leq n)$ を状態記述要素と呼ぶ。状態記述要素は一つあるいは二つの引数を有し、第一引数で指定される端末が第二引数の端末との間で、その状態記述要素で指定される状態にあることを表す。イベントも一つあるいは二つの引数を有し、第一引数で指定される端末においてそのイベントが発生したことを表す。イベントの第二引数は、そのイベントにおいて入力された他端末の情報を表す。上記の規則はイベントが発生したときに、そのイベントにより端末の状態変化が起こる端末群の状態が現状態から次状態に遷移することを表す。一つの規則内で同じ端末変数は同一の端末を表す。

例 1 図 1 に、通信サービスの動作を STR で記述した例を示す。この例では、A、B はそれぞれ端末を表す変数である。この規則は、ダイヤルトーン (dial-tone) 受信中の端末 A が、空き (idle) 状態である端末 B にダイヤル (dial) したとき、端末 A が呼返音 (ring-back) 受信状態へ遷移し、端末 B が呼出音 (ringing) の鳴っている状態に遷移することを規定する。

端末の状態を定義した後で、規則の適用法を述べる。 t を端末、状態記述要素 p_1, \dots, p_l の第一引数が t であると仮定する。このとき、 t がもつ状態記述要素が p_1, \dots, p_l であるとき、 t は $p_i (i=1, \dots, l)$ を満足する、あるいは t は状態 $p_i (i=1, \dots, l)$ にあると定義する。ある規則 r がある端末群 t_1, \dots, t_n に対して適用できる条件は、 r に含まれるイベントの発生端末が t_1, \dots, t_n のどれかであり、 t_1, \dots, t_n が規則 r の現状態を満足するときである。

例 2 端末 t_1 と t_2 が、dial-tone(t_1)、idle(t_2) という状態にあるときに、端末 t_1 において端末 t_2 にダイヤルするイベント dial(t_1, t_2) が起こった場合を考える。このとき、例 1 の規則は適用可能であり、規則適用後に t_1, t_2 はそれぞれ、ring-back(t_1)、ringing(t_2) という状態に遷移する。

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

図 1 STR 規則例

Fig. 1 An example STR rule.

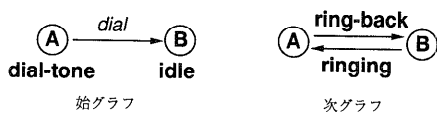


図2 STR規則のグラフ表現

Fig. 2 Graph representation of an STR rule.

一般に一つのイベントが生じたときに、上で述べた適用条件を満足する規則が一つに定まるとは限らない。規則適用の優先順位をSTR規則をグラフ表現することにより定義する。

通信システムに属するすべての端末の状態を次の対応関係により一つの有向グラフ（連結とは限らない）として表現することができる。このグラフをシステムグラフと呼ぶ。

グラフ表現法 端末を頂点、1引数の状態記述要素およびイベントを頂点のラベル、2引数の状態記述要素およびイベントを第一引数から第二引数へのラベル付き有向辺とする。有向辺の第一引数に対応する頂点を始点、第二引数に対応する頂点を終点と呼ぶ。

上の対応関係により、一つのSTR規則を、現状態とイベントのグラフ表現である始グラフと次状態のグラフ表現である次グラフからなる二つの有向グラフの組として表現することができる。図2に例1の規則のグラフ表現を示す。

このときSTR規則の適用は条件を満足する部分グラフの探索問題として定式化することができる。すなわち、システムグラフ中に規則の始グラフが存在すれば、その始グラフをもつ規則は適用条件を満足する。二つのSTR規則 r_1, r_2 がともに適用条件を満足し、 r_2 の始グラフが r_1 の始グラフの部分グラフとなるとき、 r_1 のほうが r_2 より適用の優先順位が高い。イベントが起こったときに、より優先順位の高い規則がない規則が適用される。これは規則の適用条件を満足する規則が複数ある場合には、より詳細に適用条件（現状態）を規定している規則を適用することを表す。この適用法を用いて、STRの記述能力が十分実用的であることが確認されている⁹⁾。

例3 端末 t_1, t_2, t_3 があり、それらの状態が次に示す場合を考える； t_1 の状態：dial-tone(t_1)、 t_2 の状態：idle(t_2)、m-cfv(t_2, t_3)、 t_3 の状態：idle(t_3)。ここで、m-cfv(t_2, t_3)は端末 t_2 から端末 t_3 へ転送を登録してあることを表す。このとき、端末 t_1 において端末 t_2 にダイヤルするイベントdial(t_1, t_2)が起こったときに、適用条件を満足する規則として、例1の規則と次の規則がある場合を考える。

dial-tone(A), idle(B), m-cfv(B, C), idle(C)
dial(A, B) :
ring-back(A, B), idle(B), m-cfv(B, C),
ringing(C, A).

この転送登録をしてある規則の始グラフが例1の規則の始グラフを含んでいるので、この転送登録をしてある規則が優先して適用される。

2.2 詳細仕様記述言語 STR/D

STRで記述された端末の状態変化を通信システムに接続された実際の端末上の変化として実現するために、通信システムあるいは端末を制御するための仕様を詳細仕様記述言語STR/Dで記述する。STR/Dは、STRにおける端末ごとの状態遷移時に実行する通信システム制御動作を指定する規則の集合である。通信システム制御動作をタスクと呼ぶ。次の形式で定義されるSTR/D規則は、ある端末がどのような状態遷移をしたときに、どのようなタスクを実行するかを、位置指定とタスク指定の組で与える。

位置指定 {タスク指定}；

位置指定には、次の8種類の指定法がある。それぞれの指定法は、予約語の後の“(”と“)”で囲まれた中に、タスク指定で指定されるタスクを実行する位置を状態記述要素あるいはイベントを用いて記述する。

(1) 始点指定 (予約語 start)：プロセスの初期化動作を規定する。

(2) 終点指定 (予約語 term)：プロセスの終了動作を規定する。

(3) 部分状態指定 (予約語 primitive)：ある状態を構成する状態記述要素の一部を指定し、指定された状態記述要素を含む状態への遷移の直前にタスクを実行する。

(4) 状態指定 (予約語 state)：ある状態を構成する状態記述要素すべてを指定し、指定された状態への遷移の直前にタスクを実行する。

(5) 入力指定 (予約語 input)：指定された入力の直後にタスクを実行する。

(6) 部分状態入力指定 (予約語 primitive-input)：指定された部分状態で指定された入力を受けた直後にタスクを実行する。

(7) 状態入力指定 (予約語 state-input)：指定された状態で指定された入力を受けた直後にタスクを実行する。

(8) 状態遷移指定 (予約語 transition)：二つの状態間の状態記述要素の変化を指定し、指定された状態遷移が完了する直前にタスクを実行する。

通信ソフトウェアの仕様⁹⁾では、ある状態において

```

primitive(dial-tone(A))
{ tone_on(DLTONE); }           (r1)

state-input(dial-tone(A) & (event dial(A, B)))
{ tone_on(RBTONE); }         (r2)

transition(-(idle(B)) +(ringing(A, B)))
{ tone_on(RGTONE); }         (r3)

```

図3 STR/D規則例

Fig. 3 An example STR/D rule.

入力を受けとった後、次の状態に遷移するまでの間にタスクを実行することで通信サービスを実現することができる。この位置は状態入力指定を用いることにより特定することができる。したがって、これら8種類の位置指定法を用いることにより生成される状態遷移形式のプロセス仕様上でタスクを実行すべき任意の位置を指定することが可能である。ただし、STR/Dで記述する仕様を知識として再利用するために、これら8種類の位置指定法で十分であるかどうかについては、更に今後の検討が必要である。5章の評価で述べる通信サービスを実現する上では、十分な記述能力を有している。

図3に、図1の規則を端末上の動作として実現する仕様のSTR/D記述例を示す。規則r1は、dial-tone(A)を含む状態へ遷移する直前に、tone_on(DLTONE)を実行することを表す。規則r2は、状態dial-tone(A)でイベントdial(A, B)が起こった直後にtone_on(RBTONE)を実行することを表す。規則r3は、端末AにおいてSTR規則の現状態から次状態に遷移するときに、状態記述要素idle(A)が減少(-)して、状態記述要素ringing(A, B)が増加(+))したら、tone_on(RGTONE)を実行することを表す。

3. 通信プログラム生成法

3.1 通信サービス制御モデル

STRで記述された通信サービス仕様からプログラム仕様を生成するための通信サービス制御モデルを示す。このモデルは図4に示す階層化アーキテクチャ¹⁰⁾と一致する。

(1) 端末ごとにその端末を制御するプロセスが一つ対応する。各プロセスは通信システム内で唯一の識別子pidを持つ。

(2) 各端末制御プロセスは他の端末制御プロセスの状態を該当端末制御プロセス間の通信によってのみ知ることができる。プロセス間通信機能は下位レイヤの機能として提供される。

(3) プロセスは、制御端末で生じたイベントあるいは他プロセスからの信号を受けとったら、必要な

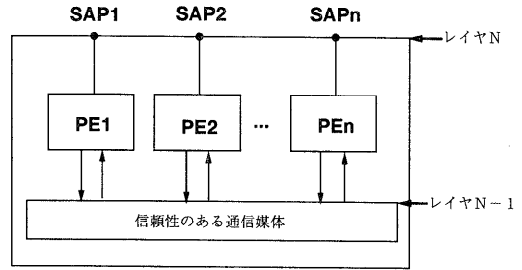


図4 階層化アーキテクチャ

Fig. 4 Layered architecture.

通信を他プロセスと行った後で、次の状態に遷移する状態遷移機械である。

3.2 プログラム生成法

STR, STR/Dで記述された仕様から前節で定義したモデルに基づくプログラム仕様を生成することができる⁷⁾。端末は通信システムへのアクセス点であり、図ではSAPとして表現されている。各SAPごとにプロセスPEが対応する。

STR規則のグラフ表現における始グラフに含まれる頂点のうち、ラベルをもつ頂点あるいは辺の始点にあたる頂点のすべてをちょうど一回だけ通過する経路が唯一に存在するものに制限する。この経路を幹と呼ぶ。この制限を設けても、通信サービスに対する十分な仕様記述能力を有する⁹⁾。更に効率のよいプログラム仕様を生成することができる。幹を持たないSTR規則に対しても仮想的に幹を持つグラフに変換するアルゴリズムを開発している。したがって、この制限はSTRの記述能力に制限を与えるものではない。

幹上の頂点は、STR規則の現状態に現れる端末のうち状態が記述されている端末、またはイベントが生じる端末である。したがって、幹に沿って通信を行い状態を調べることにより、STR規則の適用条件を判定することができる。また、幹には分岐がないので幹の終端に到達した段階で適用条件の判定が完了する。

合成は次の4ステップからなる。

ステップ1: STR規則の分類

入力されたSTR規則の集合をRとする。Rを同一のイベントを持つ規則ごとの集合 C_1, \dots, C_k に分類する。各 $C_j(j=1, \dots, k)$ ごとに一つの規則分類木を作る。規則分類木の頂点には規則の集合が対応し、辺には幹を構成する頂点の近傍が対応する。ここで、ある頂点 v の近傍とは、 v と v を始点とする辺およびそれらに付いているラベルである。

あるイベント e をもつ規則集合 C_j に対する規則分類木Tの作り方を述べる。T上の頂点 v に対して、T

の根からの距離 D を、根から v に至る経路上にある辺の数として定義する。この距離 D を使って、帰納的に T の作り方を示す。

(1) $D=0$ のとき、 T の根にはイベント e をもつ規則すべてからなる集合を対応させる。 T の根を $v_{0,0}$ 、対応する規則集合を $R_{0,0}(=C_j)$ と表す。

(2) $D=n$ までの T ができていると仮定する。このとき、 $D=n+1$ の頂点のとり方およびその頂点に至る辺およびそのラベルを定義する。 $D=n$ のある頂点 u に対応する規則の集合が $R_{n,i}$ であるとする。 $R_{n,i}$ の要素 r に対して、 r の始グラフに含まれる幹上の頂点 v に、イベント e が生じた頂点 (イベント生起点と呼ぶ) からの距離 d を、イベント生起点から v に至る経路上にある辺の数として定義する。 $R_{n,i}$ の要素をイベント生起点からの距離 d が n である幹上の頂点の近傍 N が同一である規則ごとの集合に分類する。この分類された集合ごとに T の新たな頂点を作り、それぞれの頂点に分類された規則の集合を対応させる。この新しく作られた頂点に u からの有向辺を作り、その有向辺のラベルとして、近傍 N を割り付ける。 $D=n+1$ である頂点を $v_{n+1,i}(i=0, \dots, i_{n+1}), v_{n+1,i}$ に対応する規則集合を $R_{n+1,i}$ と表す。作成した k 個の規則分類木のすべての辺に互いに異なる状態通知信号名を割り付ける。

この作り方より、幹の終端に当たる頂点の近傍は規則分類木には現れない。また、すべての規則には幹がただ一つだけ存在するので、規則分類木は必ず木になる。二つの規則 $r1$ と $r2$ があるとき、 $r1$ と $r2$ の幹上の頂点の近傍がイベント生起点から n 番目の頂点まで同じで、 $n+1$ 番目ではじめて異なるとき、規則分類木の根からの距離が n までの頂点では $r1$ と $r2$ はつねに同時に存在する。このため、 $r1$ と $r2$ の幹 $t1$ と $t2$ のイベント生起点からの距離が n までの頂点に対して

は、同一の通信で規則の適用条件判定が可能となる。ステップ3で合成される概要プロセス仕様は、規則分類木の根から深さ優先で、辺のラベルとして割り付けられている近傍を調べるように、規則上の幹に沿って通信を行う。

図5に、イベント e をもつ規則が $r1$ と $r2$ だけの場合における規則分類木を示す。図では、頂点に割り付けられた規則集合と辺に割り付けられた近傍だけを表してある。

ステップ2: プロセス動作パターンの生成

規則分類木およびSTR規則からプロセス動作パターン(PBPと略す)を生成する。はじめに記号と用語を定義する。

$v(r, n)$: 規則 r に含まれる幹上の頂点のうちイベント生起点からの距離が n である頂点とする。

$N(r, n)$: $v(r, n)$ の近傍とする。

$PN(r, n)$: $N(r, n)$ からイベントを除いたものであり、準近傍と呼ぶ。

L : R から得られる各規則に対する幹のうち最長である幹の辺数。

以下の処理を $n=0, \dots, L$ に対して行う。

$R_{n,i}(i=1, \dots, k_n)$ を、規則分類木上で $D=n$ である頂点に対応する規則集合とする。 $R_{n,i}$ の各要素 r を、 $PN(r, n)$ が同一である規則ごとの集合に分類する。この分類された集合ごとにPBPを作成する。PBPの基本的な形を図6に示す。 $R_{n,i}$ に対する、図中のPBPの構成要素(1), ..., (11)の作成法を示す。 $r \in R_{n,i}$ とする。

(1) 始状態を $PN(r, n)$ とする。

(2) $n=0$ のときは、イベントを受信する信号受信を作成する。 $n \neq 0$ のときは、ステップ1において $v(r, n-1)$ を始点とし、 $v(r, n)$ を終点とする辺に割り付けられた状態通知信号名をもつ信号を受信する。あるイベントまたは状態通知信号 s には、既に適用条件を満足している規則のうち適用優先順位が最も高い規則を表す確定規則、適用条件判定を更に行う必要がある規則の集合を表す規則集合、規則分類木上の根から s に至る経路上にある辺のラベルとして割り付けられた近傍すべてを表す近傍情報を含んでいる。

(3) r の幹上にない辺のうち、 $v(r, i)(i=0, \dots, n-1)$ を始点とする辺の終点が、 $v(r, n)$ と一致するかまたは $v(r, n)$ を始点とする辺の終点に一致するか否かの同型判定を行う。この同型判定を行う理由は、幹に沿った通信により幹上の各頂点の準近傍を使った規則適用条件の判定を行っただけでは、システムグラフに含まれる規則の始グラフを完全に特定することがで

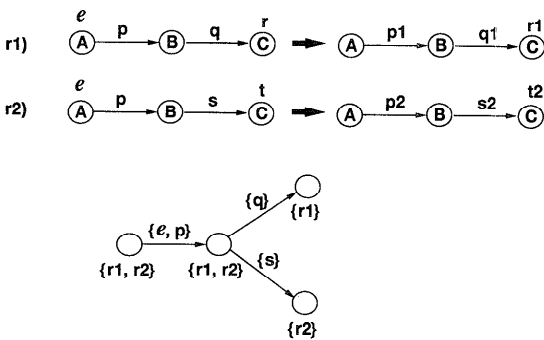


図5 規則分類木の例
Fig. 5 A rule classification tree.

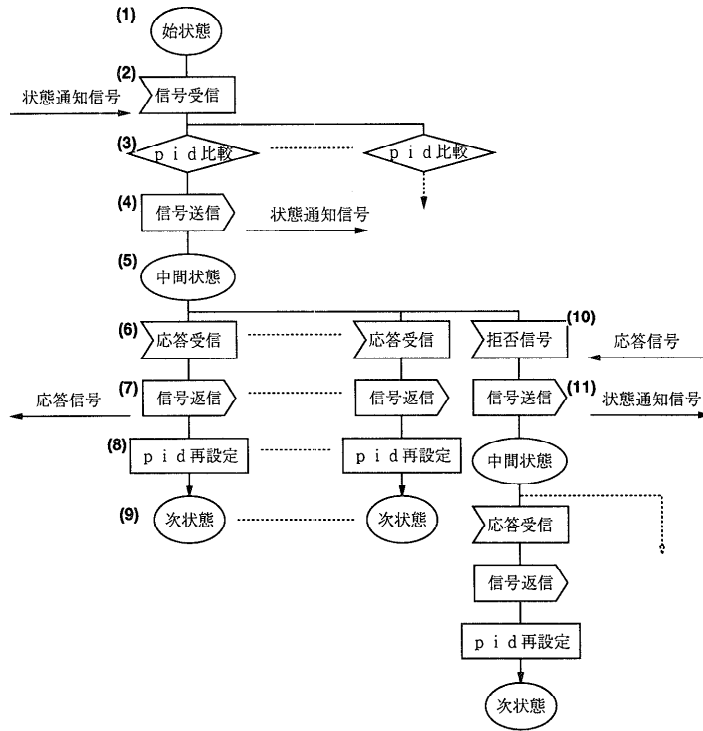


図6 PBPの基本形
Fig. 6 Process behavior pattern sample.

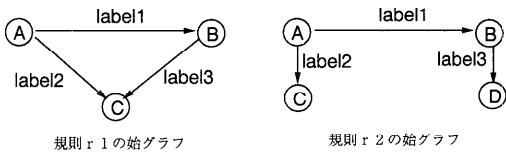


図7 pid比較による同型判定
Fig. 7 Two graphs distinguished by comparing pids.

きないからである。

図7に同型判定が必要な例を示す。規則 r_1 , r_2 の始グラフのみを表す。どちらにおいても、プロセス A からプロセス B に至る経路が幹である。プロセス A でイベントが起きると、プロセス B に状態通知信号が送られ、プロセス B で規則 r_1 と規則 r_2 の判定を行う。B は label3 で隣接しているプロセスが A と label2 で隣接しているかどうかを判定する。この判定はプロセス B の近傍情報とプロセス B が受け取った状態通知信号とにより判定することができる。

次に、 $R_{n,i}$ のすべての要素 r に対して $v(r, n+1)$ を決定する。 $v(r, n+1)$ は $v(r, n)$ を始点とする辺 f のうち、ラベルが r の幹における $v(r, n)$ を始点とし $v(r, n+1)$ を終点とする辺のラベルと一致しており、かつ

f の終点の pid が $v(r, i) (i=1, \dots, n)$ と異なるものとして特定することができる。

(4) (2) で受けとった信号を元に始状態 PN (r, n) における確定規則、規則集合、近傍情報の更新を行う。上記 (3) で $v(r, n+1)$ が決定し、かつ (2) で受けとった信号に含まれる規則情報と $R_{n,i}$ の共通集合を rules としたとき、rules が空集合でない場合には、(3) で決定した $v(r, n+1)$ にステップ1で割り付けられた状態通知信号名をもち、更新した3種類の情報を含む信号を送信する。このとき、 $v(r, n+1)$ が複数存在する場合には、任意の一つを選ぶ。残りの頂点は、分岐情報として記憶する。作成した近傍情報も記憶する。

上記 (3) で $v(r, n+1)$ が見つからなかった場合、または rules が空集合のときには、更新した確定規則を表す応答信号または拒否信号を $v(r, n)$ に返す。応答信号には更新した近傍情報を含める。

(5) 応答信号または拒否信号を受信するための中間状態を作成する。

(6) $v(r, n+1)$ からの応答信号を受信する。

(7) $v(r, n-1)$ に (6) の応答信号を送信する。

(8) (6) で受けとった応答信号から次グラフで規定される近傍に状態を置き換える。

(9) 次グラフにおける近傍を状態として表現する。

(10) 拒否信号を受けとる。

(11) (4)で分岐情報が記憶されていれば、同時に記憶した近傍情報を使って分岐情報内の任意の頂点に(4)と同様に状態通知信号を送る。残りの分岐情報があれば再度記憶する。

(4)で分岐情報が記憶されていない場合、(4)で記憶された近傍情報に確定規則が含まれていれば、その確定規則を応答信号とし、確定規則が含まれていなければ拒否信号を、 $v(r, n-1)$ に送る。

ステップ3：概要プロセス仕様の合成

ステップ2で得られたPBPを合成して一つのプロセス仕様を合成する。合成されるプロセス仕様は、状態遷移形式の表現となり、STRで記述されたサービス仕様を実現する状態遷移に必要なプロセス間通信をすべて含んでいる。これを概要プロセス仕様と呼ぶ。

合成は初期状態であるidleから始める。すなわち、idleを始状態とするPBPをすべて集めて、idleから始まる状態遷移をすべて合成する。合成されたプロセス仕様に新たに出現した状態を、次の合成対象の状態とする。合成対象の状態に含まれる始状態をもつPBPをすべて集めて、新たな状態遷移を合成する。途中まで作成された概要プロセス仕様の状態Sに、あるPBPが合成可能である条件は、そのPBPの始状態をsとするとき、 $s \subseteq S$ であるときである。合成後、元のPBPの次状態 t_1, \dots, t_j は、それぞれ $(S-s) \cup t_1, \dots, (S-s) \cup t_j$ となる。この処理を新たな状態ができなくなるまで続ける。

ステップ4：タスク挿入

STR/D規則に記述されたタスクをステップ3で生成された概要プロセス仕様に挿入する。挿入位置はSTR/D規則の位置指定により決定する。タスク挿入後に得られるプロセス仕様を詳細プロセス仕様と呼ぶ。この詳細プロセス仕様をSDL⁹⁾で表現されるプログラムの仕様である。

生成された詳細プロセス仕様は機械的に通信プログラムに変換することができる。得られた通信プログラムは、STR/Dで記述されたタスクを実際の通信システム上で実現するための通信システム制御部品と組み合わせる動作させることができる。

上記の合成法に基づき作成した自動生成システムの概略を図8に示す。上記ステップ1, 2, 3をSTRコンパイラが行い、ステップ4をSTR/Dコンパイラが行う。詳細プロセス仕様からプログラムへの変換をコード変換モジュールが行う。

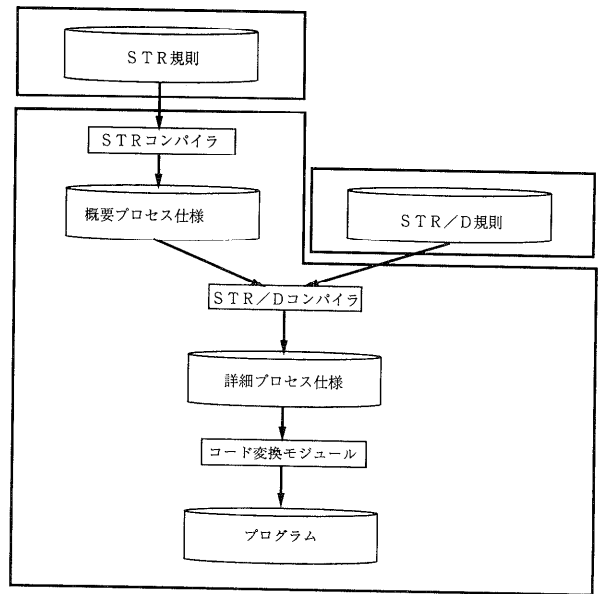


図8 プログラム生成システム概略
Fig. 8 An overview of program generation system.

4. 適用例

二段階通信サービス仕様記述から自動生成されるプログラム仕様を使ってPBXサービスを実現する適用例を示す。通信システム制御インタフェースを有する実行機構を二種類のPBX上に作成することにより、通常のPBXに対しては個別のシステムに依存しない論理インタフェースを定義できることを確認した。

4.1 プロセス実行機構の構成

本実行機構内に存在するプロセスとその役割を説明する。図9に実行機構の構成を示す。

システム共通 システム立ち上げ時に起動。

1. MAIN 発呼があると発呼した端末に端末対

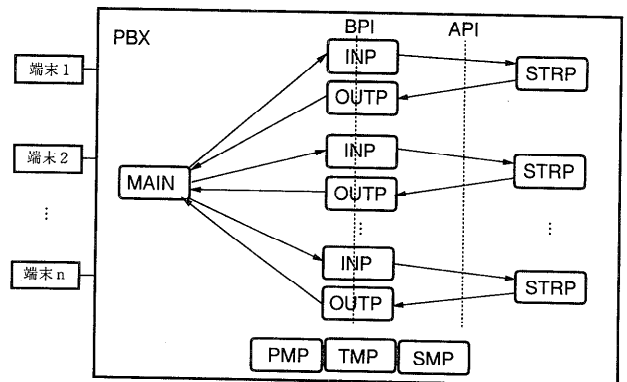


図9 プロセス構成
Fig. 9 System structure.

応プロセスを割り付ける機能を有する。

2. PMP 呼対応, 端末対応に生成されるプロセスの生成, 削除機能を提供する。

3. TMP 端末制御プロセスと端末との対応関係, プロセスと呼の対応関係管理, 加入者データテーブルの管理機能を有する。

4. SMP イベント, プロセス間通信の履歴管理機能を有する。実行状況を表示するために用いる。

端末対応 発呼時および空き端末への着呼時に起動。

1. STRP 二段階仕様記述から生成されるプロセスである。

2. INP 端末からの入力をSTRで定義されるイベントに変換する機能を有する。

3. OUTP STRPからの出力をハードウェア制御オーダに変換する機能を有する。

INP, OUTPについては, 端末対応に置かずに, システム共通プロセスとすることも可能であるが, 端末ごとに入出力をバッファリングする機能を容易に実現するために端末対応のプロセスとした。

4.2 通信システム制御インタフェース

本実行機構は, 汎用性を確保するため, 個別のハードウェアに依存しない通信サービス制御のための基本機能を提供する基本プリミティブ層と, 本実行機構の上で動作するSTRPに合わせたインタフェースを提供するアプリケーションプリミティブ層の2階層からなる。前者と後者の間のインタフェースを基本インタフェースBPI, 後者とSTRPとの間のインタフェースをアプリケーションインタフェースAPIと呼ぶ。図9に示すとおり, これらの階層インタフェースをINP, OUTPで実現している。

基本インタフェース

STRおよびSTR/Dで使うイベントやタスクなどの入出力プリミティブが新たに必要になったときに, ハードウェアインタフェースを意識せずに, 新たなイベント, タスクの追加が可能となるように本インタフェースを設定した。

入力については, ダイアル数字を一つのイベントとする以外は, ハードウェアやタイマからの入力単位ごとに一つのイベントを設定している。言い換えると, 今後のサービス追加を考慮して, それ以上分解する必要がない単位で入力信号列を一つのイベントとして設定する。この他に, APIを作成しやすくするために, 複数の状態変化を一つの入出力プリミティブで制御できる形に設定したものがある。

ハードウェアあるいはOSへの出力については, 目

的別に一つのプリミティブを設定する。現在設定している目的の例としては, 接続管理(connect, disconnect, switch, hold), 音制御(tone_on, tone_off), プロセス管理(start, stop), タイマ管理(set_timer, cancel_timer), リソース管理(hunt, free), プロセス間通信(send, receive), 加入者データ管理がある。接続管理, 音制御については, 通話路を使うか会議トランクを使うかの判断, どの音源を使うか等の実現手段を意識する必要はない。

BPIでは, 端末の識別は端末が接続されているPBX上の物理位置で行う。APIでは端末を電話番号で識別する。このため, 上記目的対応プリミティブの他に端末位置と電話番号の相互変換プリミティブ等の共通プリミティブがある。

アプリケーションインタフェース

APIはSTRまたはSTR/Dで記述される仕様に基づくインタフェースである。入力についてはサービス上の意味で表現する。出力については予約サービス等を除いて, ネットワーク内部のリソースを意識しないでネットワークを制御することを基本とする。APIの実現は, BPIで提供するプリミティブを使って, 端末を電話番号で識別できるようにしている。二者通話, 会議通話, 着信転送, コールウェイトイング, キャンプオン, UPT(Universal Personal Telecommunication)は, BPIで提供しているプリミティブを物理リソースを指定することなく使えるようにしたAPIで実現できる。プロセス起動は, プロセス間通信プリミティブであるsendの中で行っている。

5. 評 価

Bellcoreで仕様を規定している23サービス¹⁾をSTRを用いて記述することにより, STRの通信サービス仕様記述能力については既に確認した⁹⁾。記述量としては人手で状態遷移仕様を記述した場合と比べて約50%の工数削減ができることが分かっている。

本論文では, 代表的な通信サービスである二者通話, 会議通話, 着信転送, コールウェイトイング, キャンプオン, UPTについて, 実際にプログラムを自動生成しPBX上で動作を確認した結果を評価する。性能については人手で作成したプログラムと比較して10%の劣化が見られた。これは今後のプロセッサの高速化を考えると実用上問題にならない。

これら6つのサービスを記述するのに必要となったSTRの規則数は234, STR/D規則数は110である。STRに比べてSTR/Dの規則数は半分以下となっており, 一つのSTR/D規則が複数回使われている。非専

門家が記述したサービス仕様を通信システム上で実現するのに専門家がかわる割合は、規則数の比較で半分以下であることが分かる。これより、STR/Dを知識として充実させておくことにより、サービス追加における専門家の作業を多くは必要とせずに、サービス開発が可能となることが分かる。

ベンダの異なる二種類のPBXに対して共通の制御インタフェースを設定したことにより、詳細仕様を個別のハードウェアに依存しない知識として記述できることを確認した。設定したインタフェースは通常のPBXが持つと想定される機能のみを使用しているので、他PBXにも同じインタフェースが設定できると考える。

6. ま と め

外部から認識可能な端末の状態変化を宣言的に仕様として記述する言語STRおよびSTR/Dからなる二段階通信サービス仕様記述法を実際のPBXのソフトウェア開発に適用することにより、STRの記述能力と合わせてその有効性を示した。自動生成された通信サービス制御プログラムの性能測定により、実行時間についても十分な性能が得られることを確認した。

今後の課題として以下がある。STRおよびSTR/Dで記述された仕様からプログラム仕様への自動変換には、実行時間をあげるための最適化の余地が残っている。今回適用に用いた通信システムはPBXであり、アーキテクチャに対する制限がない。実際の通信網では、与えられたアーキテクチャに合わせたソフトウェアを自動生成することが必要である。これについては、論理的な機能モデル¹²⁾を定義し、そのモデルに合わせてソフトウェア生成を行う手法が有効であると考えられる。

謝辞 本研究を進めるにあたり、有益な御助言を頂いた葉原耕平博士、寺島信義博士に感謝いたします。有益な御討論をして頂いた通信ソフトウェア研究室の皆様へ感謝いたします。本論文に対して有益な御助言を頂いた査読者の方に感謝いたします。

参 考 文 献

- 1) Garrahan, J. J., Russo, P. A., Kitami, K. and Kung, R.: Intelligent Network Overview, *IEEE Comm. Mag.*, Vol. 31, No. 3, pp. 30-36 (1993).
- 2) CCITT, Revised Recommendation Z.100: CCITT Specification and Description Language (SDL) (1992).
- 3) Bolognesi, T. and Brinksma, E.: Introduction to the ISO Specification Language LOTOS, *Comput. Networks and ISDN Syst.*, Vol. 14, pp.

25-59 (1987).

- 4) Faci, M., Logrippo, L. and Stepien, B.: Formal Specification of Telephone Systems in LOTOS: the Constraint-Oriented Style Approach, *Comput. Networks and ISDN Syst.*, Vol. 21, pp. 53-67 (1991).
- 5) Drayton, L., Chetwynd, A. and Blair, G.: Introduction to LOTOS through a Worked Example, *Computer Comm.*, Vol. 15, No. 2, pp. 70-85 (1992).
- 6) Hirakawa, Y. and Takenaka, T.: Telecommunication Service Description Using State Transition Rules, *Int. Workshop on Software Specification and Design*, pp. 140-147 (1991).
- 7) Kawata, K., Takura, A. and Ohta, T.: On a Communication Software Generation Method from Communication Service Specifications Described by a Declarative Language, *Proc. Fifth Int. Conference on Comput. and Information*, Sudbury, Canada, pp. 116-122 (1993).
- 8) Takami, K., Harada, Y., Ohta, T. and Terashima, N.: A Visual Design Support System for Telecommunications Services, *IEEE Int. Phoenix Conference on Computers and Communications*, pp. 593-599 (1993).
- 9) 日本電信電話株式会社: 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(レイヤ3回線交換編), 第3版 (1992).
- 10) Liu, M. T.: *Protocol Engineering, Advances in Computers*, Vol. 29, Academic Press (1989).
- 11) Bellcore: Features Common to Residence and Business Customers 2, 3, LSSGR, Technical Reference TR-TSY-000521-000522, Issue 2 (1987).
- 12) Sera, T., Takura, A. and Ohta, T.: Distributed Functional Model for Telecommunication Service Specifications Described with Terminal Behaviors, *JC-CNSS'94* (1994).

(平成6年8月23日受付)

(平成7年1月12日採録)

**田倉 昭 (正会員)**

1957年生, 1979年東京工業大学理学部卒業, 1981年同大学院修士課程修了, 同年電電公社(現NTT)入社, プロトコル設計支援およびその適用に関する研究に従事, 1992年2月にATRに出向, 通信ソフトウェアの自動作成の研究に従事, 電子情報通信学会, IEEE各会員.

**太田 理 (正会員)**

1945年生, 1968年九州大学工学部電子卒業, 1970年同大学院修士課程修了, 同年電電公社(現NTT)入社, 電子交換機のソフトウェア, 共通OSの研究開発に従事, 1992年2月にATRに出向, 通信ソフトウェアの自動作成, セキュリティの研究に従事, 通信ソフトウェア研究室長, 工学博士, 電子情報通信学会, IEEE各会員.