

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

高 見 一 正[†] 新 津 善 弘[†]

ソフトウェア開発の上流工程において、要求仕様の形式的な記述による定義と仕様検証の機械的な支援を目指す技術が望まれている。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(以下 MSC と略す)で代表されるような手続き型の仕様記述法と if-then 型のルール集合による宣言型の仕様記述法が提案されている。しかし、それぞれの仕様記述法には長所短所があり、それぞれの利点を生かせるための仕様間の橋渡しを実現する必要がある。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。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から構成されるルールの集合によるサービス仕様、を定義する。次に、2つの仕様記述法における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、本論文で提案した方法に基づき、電話サービスの MSC 記述仕様を例にルール集合への変換を行い、その変換結果の評価と仕様確認と検証の観点から提案した変換法の有用性を考察する。

A Specification Description Conversion Method of Message Sequence Charts into Rules in Telecommunication Services

KAZUMASA TAKAMI[†] and YOSHIHIRO NIITSU[†]

In order to precisely acquire demands on software, specification description technique to formally define the demands and automatic verification support technique of the specification are desired especially on upper process of software development. On telecommunication software as well, two types of specification description method are proposed. One is a procedural specification description method of which message sequence chart (MSC) is a representative. Another is a declarative specification description method, for example, specification description method based on a set of if-then type rules. However, as each method has merits and demerits respectively, it is necessary to realize conversion between the specifications in order to utilize merits. This paper proposes a conversion method of specification described by MSC into specification expressed by a set of if-then type rules. The conversion method doesn't lose merits of MSC and besides can improve specification verification capabilities. Concretely, firstly we define two service specification methods. One is described as temporal message sequences between a terminal and a network. Another is described by a set of rules whose rule is consist of terminal states and a terminal input event. Secondly, we discuss equivalent relationship between two specification description primitives and propose a method that makes it possible to convert MSC description into rule description. Finally, we present application examples of the proposed conversion method to telecommunication services specifications. Furthermore, this paper describes evaluation of converted results and considerations about validity from the viewpoints of specification confirmation and verification.

1. はじめに

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

[†] NTT ネットワークサービスシステム研究所高機能処理研究部サービス制御方式研究グループ
Intelligent Network Systems Laboratory, NTT Network Service System Laboratories

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^{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。

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^{1)~3)}。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³⁾, マクロ化シーケンス検証, 遷移手順検証⁵⁾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^{6)~8)}。すなわち, 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⁹⁾。

そこで, 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。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。

以下, 本論文では, 2 章で MSC とルール集合による仕様を定義する。3 章では, 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。なお, 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, 4 章で具体的な適用例として, 文献 1) と 2) において IN (Intelli-

gent Network) のサービス仕様入力として提案されている LSDL (Layered Service Description Language) による仕様記述から文献 4) で提案されている STR (State Transition Rule) 言語への変換を示し, その適用例を用いて考察を行う。

2. 仕様記述

本章では, 変換元と変換先のサービス仕様として MSC とルール集合による仕様のそれぞれを定義する。

2.1 MSC による仕様

MSC によるサービス仕様 MSPC を以下のように定義する。

$$\text{MSPC} = \langle N, P_{\text{msc}}, C, \Sigma \rangle$$

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, パラメータ 1, ..., パラメータ k)

m_j = 記述要素名(始点, 終点, パラメータ 1, ..., パラメータ 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 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

表 1 MSC 記述要素における始点と終点の組み合わせ
Table 1 A combination of start point and end point on MSC description primitive.

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 2 MSC 記述要素のための制御文
Table 2 Control statements for MSC description.

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)	NW ◇ I
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	NW Y
end 文	制御の終了を示す	.end(NW, process)	NW ●
path 文	端末と通信網の接続構成を定義する	.path (A, NW)	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

```

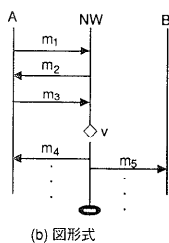


図 1 MSC による仕様記述例

Fig. 1 A specification description example based on MSC.

れる。

(3) C は表 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 1 に示す。図 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図が記述されている。図 1 (a) はテキスト形式での記述であり、図 1 (b) はその図形式による表現を示す。

2.2 ルール集合による仕様

ルール集合によるサービス仕様 RSPC は、端末の状態遷移と遷移の要因である端末に入力されるイベントから構成されるルールを記述することにより以下のよう

$$RSPC = \langle T, R, \Phi, s_0, f_0 \rangle$$

ここで、

a) T は端末の集合である。要素を t で表す。

b) R はルール集合を示す。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。なお、端末の状態は状態記述要素 p の集合として記述される。

$$r = cs : ns$$

cs : 端末の現状態を表す。

ns : 端末の次状態を表す。

e : 遷移の起因となる端末から入力されるイベントを示す。

c) 状態記述要素 p は以下の 2 種類の表記を持つ。

- $p_k(t)$: 端末 t 自身の状態を表記する時。
- $p_j(t, \gamma t)$: 端末 t の状態が、T に属する t 以外の端末と関係のある状態を表記する時。ここで、 γt は端末 t と関連がある任意の端末を示す。

d) イベント e は以下の 2 種類の表記がある。

- $e_k(t)$: 端末 t から入力され、端末 t の状態を変化させるイベントを示す。
- $e_j(t, \gamma t)$: 端末 t から入力され、端末 t とその端末と関連のある端末の状態も変化させるイベントを示す。

e) Φ 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 は端末の状態の集合であり、状態記述要素 p の集合として表現される。

$$r = M(s_i, e) :$$

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 $s_i \models cs$ を満足するルールを抽出する。サービス仕様が確定した時には、ルールの競合は既に解消されており唯一のルールのみが決定される。

$$s_j = W(r, s_i) :$$

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。

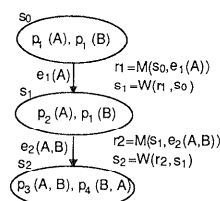
f) s_0 は端末の初期状態の集合である。

g) f_0 は端末の最終状態の集合である。

上記で示したルール集合による仕様記述例を図 2 に示す。図 2 には、2 端末が定義されており、図 2 (a) はルール集合の例を示している。図 2 (b) はそのルールを適用して得られる端末の状態の集合からなる状態 s と端末から入力されるイベント e からなる状態遷移表現を示している。

$$\begin{aligned}
r1) & p_1(A) : e_1(A) : p_2(A). \\
r2) & p_2(A), p_1(B) : e_2(A,B) : p_3(A,B), p_4(B,A).
\end{aligned}$$

(a) ルール集合



(b) ルール適用による状態遷移表現

図 2 ルール集合による仕様記述例

Fig. 2 A specification description example based on rule set.

3. MSC からルールへの変換法

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およ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。

(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およ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。

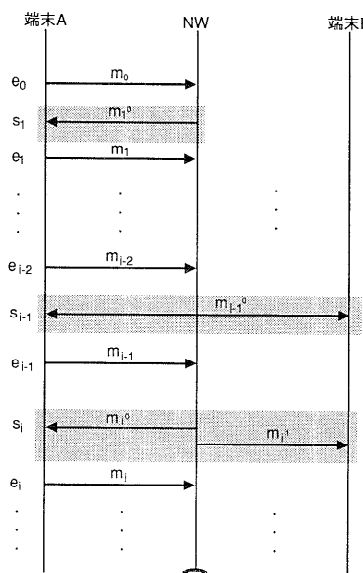
(3) 端末から通信網へのメッセージ記述要素を端末に入力されたイベントと対応させる。

(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。

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およ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、2 章の表 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。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に対応する。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 3 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 $m_0, m_1, \dots, m_{i-2}, m_{i-1}, m_i$ がイベント $e_0, e_1, \dots, e_{i-2}, e_{i-1}, e_i$ にそれぞれ対応している。また、端末の状態 $s_1, \dots, s_{i-1}, s_i$ はイベントに対応するメッセージに挟



(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 3 MSC 記述と端末状態およびイベントの関係

Fig. 3 A relationship between MSC description and terminal state or event.

まれる MSC 記述要素と対応する。

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 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 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 3 は、図 3 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 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、 $m_i^0(NW, x)$

表3 状態変換要素
Table 3 State conversion elements.

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 $P_{m_i^1}(y) \cdot P_0(x)$	
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		$P_0(x)$	$s_{i-1} \leftarrow \{s_{i-1} \cup P_i\}$
6		$P_{v_i^0}(\alpha), \dots, P_{v_i^k}(\alpha)$ $v = \langle v_i^0(\alpha), \dots, v_i^k(\alpha) \rangle$	
7		$P_{m_i^0}(\beta, \gamma \beta)$ $m_i^0(NW, NULL, \beta, \gamma \beta)$	$s_i \leftarrow \{s_i \cup P_i\}$ $s_{i-1} \leftarrow \{s_{i-1} \cup P_i\}$

(注) 1) P_0 : 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) $\gamma \beta$: β に設定された端末と関係をもつ端末識別子が入る。
5) $\cup$: 和集合を構成することを示す。

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 $p_{r0}(x)$ 、 $p_{mi^1}(y)$ 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 $p_{r0}(x)$ 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 $v = \langle v_i^0(a), \dots, v_i^k(a) \rangle$ 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 $p_{v_i^0}(a)$ 、 $\dots$ 、 $p_{v_i^k}(a)$ (ここで、 a 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。

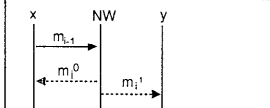
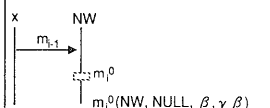
項番 7 は、網内処理記述要素の状態変換要素種別と状態記述要素への変換を示す。網内処理記述要素 $m_i^0(NW, NULL, \beta, \gamma \beta)$ は、端末から通信網へのメッセージ $m_{i-1}(x, NW)$ に対するイベントが入力できるための条件を示す状態および遷移後も維持されるサービスの起動条件等を定義するための状態であると解釈できる。従って、変換された状態記述要素 $p_{mi^0}(\beta, \gamma \beta)$ (ここで、 β は MSC の仕様記述において記述された端末識別子を示し、 $\gamma \beta$ は β と関連ある端末識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

表4 イベント変換要素
Table 4 Event conversion elements.

項番	変換要素種別	イベント e_{i-1}
1		$e_{i-1}(x, y)$
2		$e_{i-1}(x, \beta \text{ 又は } \gamma \beta)$
3	上記以外 (表3参照)	$e_{i-1}(x)$

(注) β 、 $\gamma \beta$ は表3の(注)3)、4)参照。

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 $e_{i-1}(x)$ として変換される。

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⁵⁾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 、または 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3のそれぞれの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

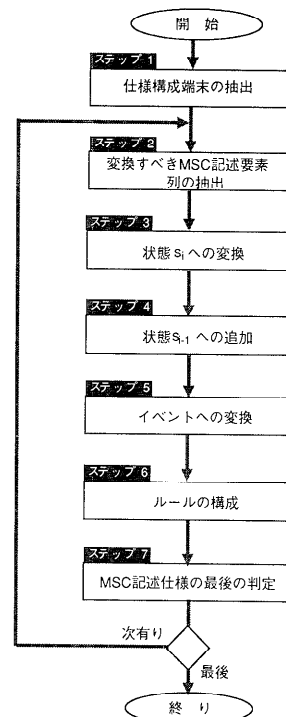


図4 変換処理フロー

Fig. 4 Conversion processing flowchart.

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_{s0} 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2参照)、同じメッセージで分岐ごとに複数種類のイベント変換要素に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 およびイベント e_{i-1} から以下のルールを構成する。

$s_{i-1} \quad e_{i-1} : \quad s_i$

ただし、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。なお、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に示す。

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。

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 LSDL (Layered Service Description Language) 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 (State Transition Rule) 言語による記述法を適用する。

LSDL のステップ1記述による POTS の仕様記述例を付録5(1)に示す。また、付録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5(2)に示す。ここで、LSDL 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、 $x=A$ または 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4参照) の基本仕様に関して LSDL 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5に示す(なお、記述量の差異については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。すなわち、それ

表5 記述量の比較

Table 5 Comparison of description volume.

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

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⁹⁾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で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。

4.2 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、か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ば、サービス要求仕様の確認支援として有効である。

表5には、LSDL 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 LSDL 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、LSDL 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 LSDL 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5の中から1つの例を示すと次のようになる。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 LSDL 記述のまとめりが、

```
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、LSDL 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、LSDL 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⁹⁾による LSDL 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。

4.3 サービス仕様競合検証への適用

LSDL 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 6) 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) を検出する方法が文献 7), 8) で提案されている。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 6)~8) 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 LSDL 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) LSDL 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) LSDL 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 6) 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 LSDL 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ば、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。

5. あとがき

本論文では、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およ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。すなわち、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

様確認およ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。

今後の課題としては、(1) 本提案に基づいた自動変換ツールの作成、(2) 更に多くの通信サービスへの適用実験による変換法の洗練化、(3) ルール集合による仕様検証で検出された矛盾の解消結果を変換元の仕様である MSC による記述仕様に反映するための方法、が残っている。

謝辞 本研究を進める上で、有益な議論と STR インタプリタを利用して頂いた ATR 通信システム研究所の皆様へ感謝します。

参考文献

- 1) 水野 修, 新津善弘: メッセージシーケンス図入力による通信サービス仕様設計方式, 電子情報通信学会論文誌, Vol. J74-B-I, No. 12, pp. 1042-1055 (1991).
- 2) Niitsu, Y. and Mizuno, O.: Computer-Aided Stepwise Service Creation for the Intelligent Network, *IEICE Trans. Commun.*, Vol.E75-B, No. 10 (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, Vol. 31, No. 1, pp. 2-81 (1990).
- 4) Hirakawa, Y. and Takenaka, T.: Telecommunication Service Description Using State Transition Rules, *Sixth Int. Workshop on Software Specification and Design*, Como, Italy (1991).
- 5) Okamoto, M. and Niitsu, Y.: A Verification Scheme for Service Specifications Described by Information Sequence Charts, *IEICE Trans. Commun.*, Vol. E75-B, No. 10 (1992).
- 6) 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).
- 7) Inoue, Y., Takami, K. and Ohta, T.: Automatic Detection of Service Interactions in Telecommunications Service Specifications, *IEEE Int. Conference on Communications (ICC'94)* (May 1994).
- 8) 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).
- 9) 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).

付 録

付録 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図5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 4 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 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図6に示す。図6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 3: ルールマッチング制約

1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

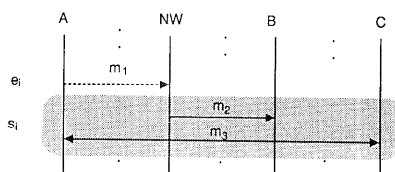


図5 2つの状態変換要素による端末の状態の構成例
Fig.5 An example of terminal state constructed by two state conversion elements.

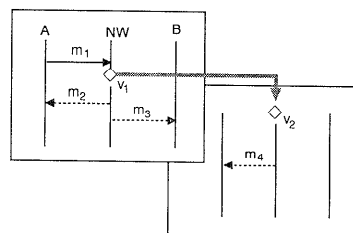


図6 2つのイベント変換要素に対応するメッセージの例
Fig.6 An example of a message with two event conversion elements.

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。

現状態: p1 (A, C), p2 (C), p3 (B)

↑ ↓ ↓
イベント: e (A, B):

付録4: 電話サービスの概要

(1) 話中着信 (Call-Waiting)

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、着側では、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。

(2) 着信転送 (Call-Forwarding Variable)

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、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。

(3) 着信拒否サービス (Terminal Call Screening)

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5: 基本電話サービス (POTS) の変換例

(1) LSDL による記述例

```
process (pots)
{
    .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) 本論文に基づき変換した 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).
```

(平成6年7月29日受付)

(平成6年11月17日採録)



高見 一正 (正会員)

1977年静岡大学工学部電気卒業。1979年同大学院修士課程修了。同年日本電信電話公社(現NTT)入社。以来、電話・パケット網間接続サービスの研究実用化、マルチメディアパケット通信方式、ATM制御方式と通信サービスの仕様記述法の研究に従事。現在、同社ネットワークサービスシステム研究所主任研究員。電子情報通信学会、IEEE各会員。



新津 善弘 (正会員)

1952年生。1976年東北大学工学部電気工学科卒業。1978年同大学院情報工学修士課程修了。同年NTT入社。以来、通信網構成、HMI高度化、仕様記述・検証、高度IN用サービス生成環境の研究開発に従事。工学博士。現在NTTネットワークサービスシステム研究所、主幹研究員。電子情報通信学会、IEEE会員。