

通信ソフトウェアに対する要求理解の一手法

横田 政憲、島 健一

ATR 通信システム研究所

本報告では、ユーザの通信サービスに対する曖昧で不完全な要求を理解し、ソフトウェア設計に必要な情報を付加し、完全な要求を導き出す通信サービスの要求理解の一手法について述べる。

一般に、通信サービスを規定するためには、目標状態の情報だけではなく、目標状態に到るための動作、状態、及び、これらの順序関係に関する情報が必要である。即ち、通信サービスは、それを構成する動作、及び、状態の順序系列と捉えることができる。この観点から、ユーザの通信サービスに対する要求の曖昧性、不完全性を、要求されているサービスと類似の既知のサービスを検索し、要求されたサービスの情報の不足を補う以下の方法により解消する。

- (1) 要求中の動作と知識ベース中の動作とのマッチングによる動作の規定
- (2) 動作をキーとした類似サービスの検索
- (3) 類似サービスとの差分情報を用いた、サービス構成動作の補完

A Requirement Understanding method for Communications Software

Masanori Yokota Ken-ichi Shima

ATR Communication Systems Research Laboratories

Sanpeidani, Inuidani, Seika-cho, Soraku-gun, Kyoto 619-02, Japan

This paper proposes a requirement understanding method for communications software to understand a ambiguous or incomplete communications software requirements. By this method, the communications service is added a sufficient information for software design in a communication domain and the complete service is derived.

A description of communication software requirements consists of 3 basic components: action, state, sequence of action or state transition.

From the point of view, communication services are understood by matching the analogy of communication actions.

The proposed requirement understanding method consists of the following 3 steps:

- (1) match the action sequences between the requirement and the knowledge base
- (2) retrieve the analogous service from the knowledge base by the matched action
- (3) supplement insufficient action sequences by differential action of analogous action.

1 はじめに

通信ソフトウェア自動作成における重要なステップの一つとして、ユーザの通信サービスに対する曖昧で不完全な要求を理解することがあげられる。我々は、曖昧で不完全な要求から、ソフトウェア設計に必要な情報を付加し、完全な要求を導き出すことを、要求理解と捉え、その手法について研究を行っている [1]。本論文ではこの観点からの通信サービスの要求理解の一手法について述べる。

一般に、通信サービスを規定するためには、目標状態の情報だけでなく、目標状態に到るための動作、状態、及び、これらの順序関係に関する情報が必要である。即ち、通信サービスは、それを構成する動作、及び、状態の順序系列と捉えることができる。一方、ユーザの通信サービスに対する要求の曖昧性、不完全性は、サービスを構成する動作や状態が、充分定義されていないことと、サービスを構成する動作や状態に抜けがあることに起因する。このような情報の不足を知識によって補完し、完全な要求を導くことが要求理解であると考えている。

ここでは、理解の一手法として、要求されているサービスと類似の既知のサービスを検索し、要求されたサービスの情報の不足を補う方法を提案する。具体的な手順としては、

- (1) 要求中の動作と知識ベース中の動作のマッチングによる動作の規定
- (2) 動作をキーとした類似サービスの検索
- (3) 類似サービスとの差分情報を用いた、サービス構成動作の補完

の3つからなる。

(1) を行うために、知識表現としてフレーム表現を採用し、動作を表すフレームのスロットとその値、及び、上位一下位関係の階層を用いて、最も適合するフレームとマッチングをとることで、動作を規定するという手法を用いた。

(2)、(3) の過程は、ジャンクが提案したスクリプト [5] の参照による、欠落動作の推定手法に似ているが、通信サービスという領域の特性により、次のような特徴がある。

動作系列の表現 — 通信サービスの実行手順は、プロトコルに従い定められた順序で動作を行う場合や、順序がどちらでもよい場合、動作が並列に行われる場合などがあり、構成動作間に様々な順序関係を持った動作系列とみなすことができる。動作系列を表現するために、スクリプトでは制御構造を明示した時間順のリストで表現しているが、ここでは、動作系列とそれを構成する動作間の関係を、時間順を明示しない単なる部分-全体関係として定義し、順序関係は、動作が起こるための条件となる状態、動作を行うことにより変化する状態を個々の動作に定義し、動作による変化をシミュレートすることで計算するようにした。

起動条件 — 通信サービスの構成動作は、様々なサービスに共通して使われる動作が多く、一つの動作、状態によって、そのサービスを特定することは、困難である。スクリプトでは、一つの動作や状態によってヘッダとマッチングをとることで起動をかけるが、このようにすると、数多くのサービスに起動がかかってしまう。そこで、類似サービスの検索というフェーズを設け、複数の動作をキーとして最も適したサービスを検索し、この動作について起動をかけるようにした。

本稿では、2章で通信サービスの知識表現について、3章で要求理解の手法について述べる。

2 知識表現

要求理解システムにおける知識表現として、フレーム表現を採用した。ここで扱うフレームは、通信サービスの世界の概念を表したもので、

動作フレーム — 動作の概念を表す

状態フレーム — 状態の概念を表す

ものフレーム — ものの概念を表す

関係フレーム (スロット) — 関係の概念を表す

に分類した。ここで、関係フレームは、概念間の関係を表すフレームで、スロットとして用いる。スロットをフレームとして表現した理由は、関係フレーム間の関係を表現するためである。スロットの詳細な分類については、2.1～2.4節で述べる。

動作に関する知識表現としては、動作を規定するための知識、類似サービスを構成する動作系列に関する知識を表現できることが必要である。動作を規定するための知識については、自然言語の格に対応するスロットと、動作による状態の変化を規定するためのスロット(これらを総称し格スロットと呼ぶ。2.2節参照)を導入し、格スロットとその値、及び、他の動作との上位一下位関係によって表現した。

サービスとその構成動作に関する知識については、部分-全体関係によって動作間の構成関係を表現し、サービスを一つの動作とみなすことで、この知識を表現した。例えば、知識ベースの一部を示した図1において、部分-全体の階層から、「電話をかける」という動作(= 発信サービス)が「発呼する」「ダイヤルする」などの動作から構成され、「発呼する」という動作が「アイドル受話器を上げる」「受話器を上げられる」などの動作から構成されていることがわかる。

<p>電話をかける 上位：複合動作、かける 部分：発呼する、ダイヤルする、 主体：発信者 条件：（発信者、所有、発信電話機）、 （発信者、状態、人アイドル）、</p> <p>事前：（発信者、状態、人アイドル）、 （発信電話機、状態、電話機アイドル）、</p> <p>事後：（発信者、状態、人通話中）、 （発信電話機、状態、電話機オフック）、</p> <p>発呼する 上位：複合動作 部分：アイドル受話器を上げる、受話器を上げられる、 電話機機能1、ループを繋ぐ、… 主体：発信者 条件：（発信者、所有、発信電話機）、 （発信者、状態、人アイドル）、</p> <p>事前：（発信者、状態、人アイドル）、 （発信電話機、状態、電話機アイドル）、</p> <p>事後：（発信者、状態、人発音を聞く）、 （発信電話機、状態、電話機オフック）、</p>	<p>アイドル受話器を上げる 上位：受話器を上げる 主体：発信者 条件：（発信者、所有、発信電話機）、 （発信者、状態、人アイドル）、 事前：（発信者、状態、人アイドル）、</p> <p>始点：発信者 終点：発信電話機</p> <p>受話器を上げられる 上位：受信動作、上げられる 逆動作：受話器を上げる 主体：電話機 対象：受話器 条件：（電話機、状態、電話機オンック）、 （人、所有、電話機） 事前：（電話機、状態、電話機オンック）、 事後：（電話機、状態、電話機オンック）、</p> <p>受話器を上げる 上位：送信動作、上げる 逆動作：受話器を上げられる 主体：人 対象：受話器</p> <p>電話機機能1 上位：電話機機能 入力動作：受話器を上げられる 出力動作：ループを繋ぐ</p>
--	--

図 1: 知識ベースの一部

構成動作間の順序関係の表現は、次のような考え方で行った。通信におけるプリミティブな動作を情報の移動という観点からみると、通信サービスは情報の送信と受信の連鎖とみなすことができ、通信機器の機能は、情報を受けることで、他の機器や人間に情報を送る動作とみなすことができる。また、情報の送信に対する受信の関係と、通信機器の機能を定義することにより、動作間の局所的な順序関係を表現することができる。そこで、我々は通信の動作を、次の4つに分類した。

能動動作 — 情報の送信と見なせる動作、

受動動作 — 情報の受信と見なせる動作、

機能動作 — 通信機器の機能、

複合動作 — 他の動作によって構成される動作。

この分類を用い、能動動作と対応する受動動作を逆動作スロットで関係付けることで、また、機能動作の入力動作スロットに受動動作を、出力動作スロットに能動動作を記述することで、動作間の局所的な順序関係を表現した。局所的な順序関係が表現できない場合は、動作による状態変化の知識を用いて、動作の実行をシミュレートし、動作がおこるための条件についての知識を用いて、次の動作を検索することで動作の順序を計算する。

例えば、図1において、「発呼する」という動作がおきるためには、発信者の状態がアイドルであるとか、発信者が電話機を持っているなどの条件が必要であり、これらの状態を記録する。この状態でおきる動作は、「アイドル受話器をあげる」という動作だけであり、この動作が最初に行われることが計算できる。また、「アイドル受話器をあげる」が能動動作であり、逆動作スロットが、上位のフレーム「受話器をあげる」という動作から「受話器を上げられる」という値を継承していることより、次の動作が「受話器を上げられる」であることがわかる。また、「電話機機能1」の入力動作スロットが「受話器を上げられる」であり、出力動作スロットが「ループを繋ぐ」であることより、これらの動作間の順序関係がわかる。

以下の節では、スロットの詳細な分類について述べる。

2.1 概念関係スロット

フレームの意味的關係を表現するためのスロットである。概念関係スロットには、次のスロットがある。

上位、下位 — 概念の抽象—具体関係を表す。下位のフレームは、上位のフレームのスロット、値を継承する。

部分、全体 — 概念の部分—全体関係を表す。動作フレームに用いる場合は、動作と、その動作を構成する動作系列中の動作との関係を表す。

逆関係 — 上位と下位、部分と全体のように、互いに逆の関係フレームを表す。

逆動作 — 能動動作に対する受動動作、受動動作に対する能動動作の関係を表す。

入力動作、出力動作 — 機能動作の入力、出力の動作を表す。

2.2 格スロット

動作を規定するために、動作フレームと他のフレームの関係を表現するためのスロットである。格スロットには、次のスロットがある。

< 全ての動作共通のスロット >

主体 — 動作の主体を表す。

対象 — 動作の対象を表す。

条件、事前、事後 — 動作が起こるための条件となる関係、起こったために消滅する関係、生じる関係を表す。これにより動作による状態変化の知識が表現できる。これらのスロットの値は(フレーム、スロット、値)の3つ組である。

< 情報伝達動作(能動動作、受動動作)のためのスロット >

起点、終点、意図 — 情報伝達の起点、終点、伝達される情報の意図を表す。これらのスロットは動作の類似をとるためのキーとなる属性である。

メディア、形式 — 伝達される情報のメディア(例えば、音声、音声信号)、形式(例えば、電話番号、名前)を表す。

2.3 属性スロット

ものフレームを規定するためのスロットである。属性スロットは、数多くあるのでその一部を示す。

所有 — ものの所有関係を表す。

状態 — ものの状態を表す。このスロットの値は、状態フレームである。

役割 — 送り手、受け手のように、ものの役割を表す。

2.4 制御スロット

要求理解過程において、入力された要求を制御するために用いるスロットである。これらのスロットは、入力フレームに用いられる。

前動作、後動作 — 要求サービスを構成する入力動作間の順序関係を表す。

マッチ — 入力フレームがマッチングしている知識ベース中のフレームを表す。入力フレームはマッチスロットの値のフレームから情報を継承する。

照合、類似 — 入力動作と対応する、類似サービス中の動作フレームを表す。照合は動作フレームと同じ動作と見なせるが、類似は動作フレームと違う動作である。(3.4節参照)

3 要求理解の手法

本節では、まず、要求理解システムの構成について述べ、次に、システム構成に従って、要求理解の手法を示す。入力は、自然言語で書かれた要求であるが、現在のところ処理を簡単にするために、サービスの動作を時間順に箇条書きにしたものという制約を加えている。なお、本節の説明を行う上で、ハンドフリーダイヤルサービス(受話器を置いたまま、ダイヤルできるサービス)を取り上げ、次に示す要求を例に上げて、説明を進めていく。

[入力文] スピーカボタンを押す。

発信音をスピーカから聞く。

ダイヤルする。

相手が応答すると、ハンドセットを上げ通話する。

3.1 システム構成

要求理解システムの構成を図2に示す。

入力された要求はパーザによって、格文法に基づいたフレーム構造に変換される。このフレーム構造は、マッチング部において、知識ベースのより具体的な(下位の)フレーム構造とマッチングがとられる。このマッチング情報とサービスの構成動作の順序関係をもとに、類似サービス検索部において類似サービスが検索される。最後に、類似サービスと入力フレーム構造の差分から、サービス構成動作が補完され、完全な要求サービスの動作系列が得られる。

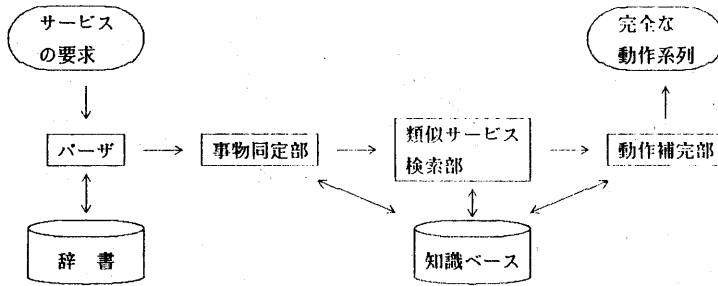


図 2: システム構成

3.2 パーザ

入力文をパーズし、格文法に基づくフレーム表現に変換する。それぞれのフレームは、辞書によって知識ベース中のフレームと、「マッチ」スロットで関係付ける。また、動作を表すフレームの間には、前動作、後動作スロットで、順序関係を付ける。

知識ベース中に対応するフレームが無い場合は、知識ベース中に対応するフレームを作成し、ユーザに質問をして、そのフレームの概念関係スロットを埋める。

例えば、上記の入力文中の「スピーカボタン」や「スピーカ」は、新しい概念なので、「スピーカボタン(スピーカ)とは何ですか?」、「スピーカボタン(スピーカ)は何の一部ですか?」といった質問をユーザに行い、「スピーカボタン」は「ボタン」の下位であり、「スピーカボタン」は「スピーカ電話」の一部である、という情報を得る。

図 3 に入力文のパーズ結果を示す。

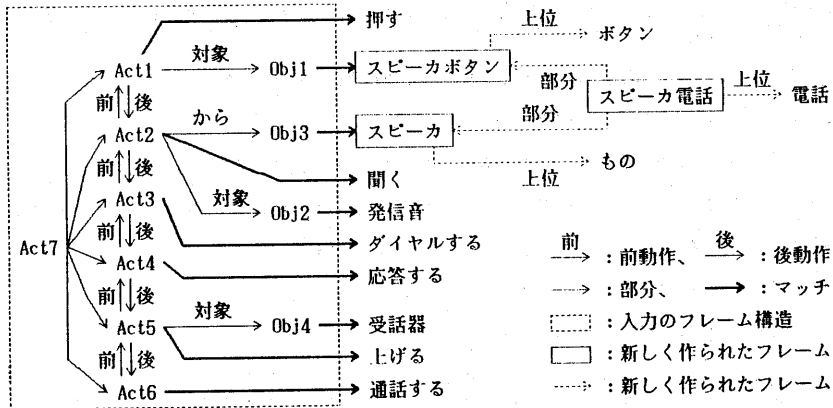


図 3: パーズ結果

3.3 マッチング部

フレーム構造に変換された入力を、知識ベース中のより具体的な(下位の)フレームとマッチングさせることで、入力を具体化する。マッチングは、格スロットや属性スロットで結ばれた2つの入力フレーム間の関係、即ち、3つ組(フレーム, スロット, 値)をキーとする部分マッチングの繰り返しによって行う [2]。

まず、部分マッチングがどのように行われるかを、図 4(a) で示すような簡単な知識ベースを例に考えてみる。

- (ケース 1) X が鳥にマッチし、Y が白い羽にマッチし、(X, 所有, Y) である場合、X は白鳥ではないかと考え、X を白鳥にマッチさせる。(図 4(b))
- (ケース 2) X が白鳥にマッチし、Y が羽にマッチし、(X, 所有, Y) である場合、Y は白い羽ではないかと考え、Y を白い羽にマッチさせる。(図 4(c))
- (ケース 3) X が鳥にマッチし、Y が羽にマッチし、(X, 所有, Y) である場合、(白鳥, 所有, 白い羽) という関係が知識ベース中にあるが、具体化しすぎると考え、X、Y のマッチは共に変化させない。このような場合に過度の具体化と呼ぶ。(図 5(d))
- (ケース 4) X が鳥にマッチし、Y が羽にマッチし、(X, 属性, Y) である場合、所有という関係が属性という関係の下位であれば、X と Y の間の関係が所有ではないかと考え、(X, 属性, Y) を (X, 所有, Y) に変える。(図 5(e))

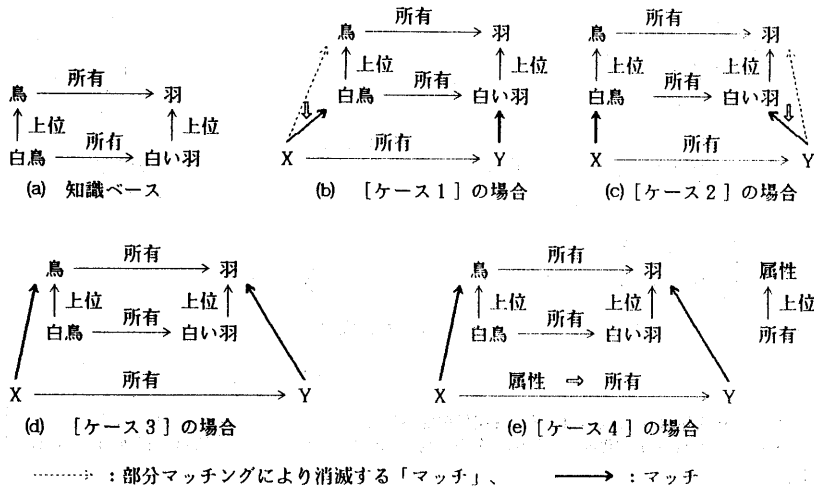


図4: 簡単な部分マッチングの例

このような様々なケースをまとめると、基本的な部分マッチングの実行条件と実行処理は以下に示すような形でまとめることができる。

入力3組(IF, IS, IV)は、IF、IVがマッチしている知識ベース中のフレームをMF、MVとしたとき、条件1～条件3を全て満たす知識ベース中の3組(F, S, V)の中で、条件4(過度の具体化になる条件)を満たさない、最も具体的な(下位の)フレームで構成されている3組と部分マッチングが行われる。但し、条件4を満たすような3組しかない場合は、最も抽象的な(上位の)フレームで構成されている3組と部分マッチングが行われる。部分マッチングが行われたときは、処理1～処理3に示す処理が行われる。

(条件1) Fは、MFと等しい、またはMFの上位である、またはMFの下位である。

(条件2) Sは、ISと等しい、またはISの上位である、またはISの下位である。

(条件3) Vは、IVと等しい、またはIVの上位である、またはIVの下位である。

(条件4) FがMFの下位である、かつ、VがMVの下位である、かつ、SがISと等しいかSの下位である。

(処理1) FがMFの下位であれば、IFのマッチ先をMFからFに変える。

(処理2) SがISの下位であれば、IFのISスロットをSスロットに変える。

(処理3) VがMVの下位であれば、IVのマッチ先をMVからVに変える。

上記の(条件1)はFがマッチしているフレームを中心とした上位-下位階層中のフレームとしか、部分マッチが行われえないという条件を表している。(条件2)(条件3)についても同様である。また(処理1)は、より下位のフレームとマッチしたときは、マッチ先を変えることによって具体化するという条件を表している。(処理2)(処理3)についても同様である。このような処理は、従来のフレーム継承によるマッチングとは異なり、より適切なマッチングとなる。

さて、入力例に戻って、最初の文「スピーカボタンを押す。」がどのように具体化されるかを考えてみる。図3のこの文のフレーム構造より、入力3組は(Act1, 対象, Obj1)であり、Act1、Obj1のマッチ先は、それぞれ、「押す」、「スピーカボタン」である。条件1～条件3を満たす知識ベース中の3組は(ボタンを押す, 対象, ボタン)しかないのので、入力3組はこの3組と部分マッチングが行われる。また、「ボタンを押す」が「押す」の下位であるので、(処理3)から、Act1のマッチ先が「ボタンを押す」に変わる。この文に対する部分マッチングの様子を図5に示す。

次に、2つめの文「発信音をスピーカから聞く」について考えてみる。図3のこの文のフレーム構造より、入力3組は、(Act2, 対象, Obj2)と(Act2, から, Obj3)である。(Act2, 対象, Obj2)に対しては、条件1～条件3を満たす3組は(聞く, 対象, 音)と(発信音を聞く, 対象, 発信音)の2つがあるが、より下位のフレームで構成されている(発信音を聞く, 対象, 発信音)と部分マッチングし、Act2のマッチ先が「発信音を聞く」に変わる。しかし、3組(Act2, から, Obj2)については、条件1～条件3を満たす3組は知識ベース中に存在しない。

我々は、このような場合に対してより柔軟なマッチングを行うため、部分マッチング時に対する、次のような特別な規則を導入している。

(規則1) Sが「事前」「事後」「起点」「終点」の場合、VがMVの部分であれば、VはMVの下位とみなす。逆に、MVがVの部分であれば、MVはVの下位とみなす。

(規則2) Sが「条件」「事前」「事後」のとき、Vは3組となるが、この3組の要素の一つV'に対し、V'とMVが等しければMVはVと等しいとみなし、V'がMVの上位(下位)であれば、V'はMVの上位(下位)であるとみなす。

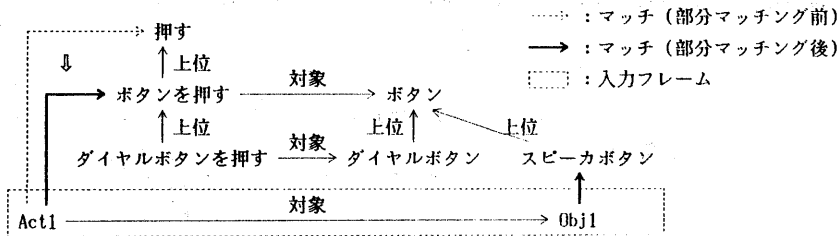


図5: 「ダイヤルボタンを押す」に対する部分マッチング

さて、このような規則を導入すると、3つ組 (Act2, から, Obj3) の部分マッチングにおいて、スピーカがスピーカ電話の部分であり、スピーカ電話が電話の下位であることより、知識ベース中の3つ組 (発信音を聞く, 始点, 電話) が検索され、からスロットが始点スロットに変化する。(図6参照)

このようにして、部分マッチングにより、入力フレームはより具体的なフレームとマッチングされる。

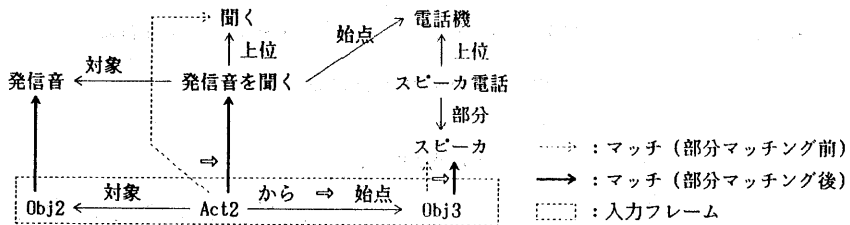


図6: 「スピーカから発信音を聞く」に対する部分マッチング

3.4 類似サービス検索部

具体化した入力動作をキーとして、入力サービスの動作系列と、知識ベース中の既知のサービス (動作) との比較することにより、類似サービスを検索する。類似サービスは、次の条件 (1)、(2)、(3) を満たすような知識ベース中のサービス (動作) である。

- (1) 入力の動作系列の個々の動作 A と、サービスを構成する動作系列の個々の動作 B との照合の数が最大である。ここで、照合とは、A のマッチ先の動作が B と等しい、または B の上位である、または B の下位である場合を言う。
- (2) 入力の動作系列の個々の動作 A と、サービスを構成する動作系列の個々の動作 B との類似の数が最大である。ここで、類似とは、A の「起点」「終点」「意図」スロットの値が、それぞれ、値がない、または B のスロット値 V と等しい、または V の上位である、または V の下位である、または V の部分である、または V の全体である場合を言う。
- (3) 照合、あるいは類似した個々の動作間の順序関係は、同じでなければならない。

手順としては、動作の部分-全体関係による動作と構成動作との関係を利用して、まず、照合の数が最大で、(3) の条件を満たすようなサービスを検索する。このとき、照合の情報を残すために、入力動作の照合スロットに、照合する動作をいれる。次に、照合しなかった入力動作に対して、順序関係を満たすような、サービスの構成動作の中から、類似

動作を検索する。ここで、類似動作が検索されたときは、類似スロットに検索された動作を入れる。類似の数が最大のものを類似サービスとする。

例に戻って考えると、3.3節での具体化により、Act1,Act2,Act3,Act4,Act5,Act6のマッチ先は、それぞれ、「ボタンを押す」「発信音を聞く」「ダイヤルする」「応答する」「受話器を上げる」「通話する」になっている。このとき、照合の数が最大になるのは、「電話をかける」というサービス（動作）に対してであり、照合の数は4である。そのときの照合しない要求中の動作は、Act1とAct4の場合と、Act1とAct5の場合がある。

次に、照合しない要求中の動作について、類似の動作がサービス中に有るかどうか調べる。Act1については、Act1は「意図」スロットに値をもたず、「起点」「終点」スロットの値が、「アイドル受話器を上げる」という動作の各スロットの値と、上位-下位関係にあるので、Act1の「類似」スロットに「アイドル受話器を上げる」を入れる。Act4,Act5に関しては、類似動作が類似サービス中にないので、類似ファクターは1となる。図7に類似サービスの検索結果を示す。

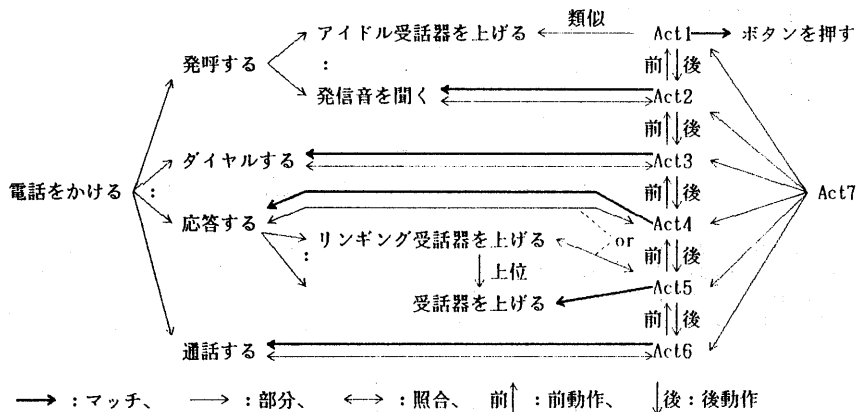


図7: 類似サービスの検索結果

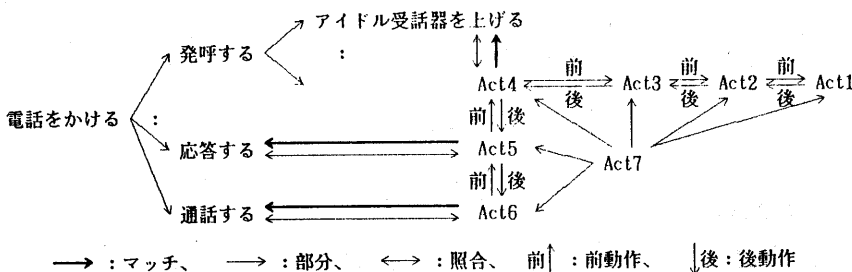


図8: Act4が「アイドル受話器を上げる」に照合すると仮定したときの照合じ状態

さて、ここで、Act4が「アイドル受話器を上げる」に照合したと仮定すると、Act1、Act2、Act3の照合先がなくなり、照合の数が3になってしまいます(図8)。よって、この場合は、照合の数が最大にならないので、検索は失敗する。

最後に、類似サービス検索のために用いたアルゴリズムについて述べる。このアルゴリズムは、多くの仮定を発生させながら進んでいくため、仮説世界を導入した。以下にこのアルゴリズムを示す。

< 照合の数が最大のサービスの検索 >

(STEP1) 入力動作のマッチ先を部分とするようなサービス（動作）に対し、仮説世界を生成し、そのサービスが類似サービスであると仮定する。ここで、マッチ先が部分となるかどうかは、上位-下位階層、部分-全体階層をたどることによって、導く。初期値として、マッチ先が類似サービスの部分となるような入力動作に対し、照合スロットの値に類似サービスを入れ、このような入力動作の数を照合の数とする。

(STEP2) 照合スロットの値が、マッチ先の動作と、等しくなく、かつ、上位でも下位でもない(即ち、照合が終わっていない)入力動作の照合スロットの値である動作Xに対し、部分スロットの値を用いて、部分動作に分解する。このとき、「条件」「事前」「事後」「逆動作」「入力動作」「出力動作」スロットの情報から、部分動作間に、順序関係を付ける。複数の順序が導かれるときは、それぞれ違う世界として分割する。次に、Xを照合スロットの値とするような入力動作に対し、分解した部分動作の中から、マッチ先の動作が等しいか、上位であるか、下位であるか、または、部分となるような動作を探し、照合スロットの値をこの動作に変える。複数の動作が見つかった時は、それぞれ、違う世界として分割する。

(STEP3) 入力動作間の順序関係が、照合スロットの値である部分動作間の順序関係を満たさない場合、その原因となる入力動作の照合スロットの値を削除し、照合の数の値を減らす。この場合、原因となる入力動作は、複数通りあるので、それぞれの場合について、違う世界として分割する。

(STEP4) 全ての入力動作の照合スロットの値が、マッチ先の動作と等しいか、上位であるか、下位であるか、または、空のとき、この世界に対する検索は成功する。

(STEP5) 検索が成功した世界の照合の数より少ない照合の数しか持たない検索中の世界を破棄する。

(STEP6) 検索中の世界が無ければ終了。あれば、その中で照合の数が最大の世界についてSTEP2に戻る。

< 類似ファクターが最大のサービスの検索 >

(STEP1) 照合スロットに値を持たない入力動作について、順序関係のみを満たすような類似の候補をサービスの構成動作より検索する。

(STEP2) 類似動作を候補から検索し、類似フレームにその値を入れる。複数の動作が検索された場合は、それぞれ違う世界として分割する。また、検索が成功してもしなくても、類似フレームに値を入れない世界を作っておく。

(STEP3) 全ての照合スロットに値を持たない入力動作について、STEP1、STEP2が完了すれば、類似フレームに値を持つ入力動作の数が最も大きい世界以外を破棄して終了する。

3.5 動作補完部

動作補完部は、類似サービスとの差分情報によって、抜けのある動作フレームの生成、動作フレームの修正を行い、完全な要求サービスの生成を行う。

動作フレームの生成および修正は、次のような手順によって行われる。

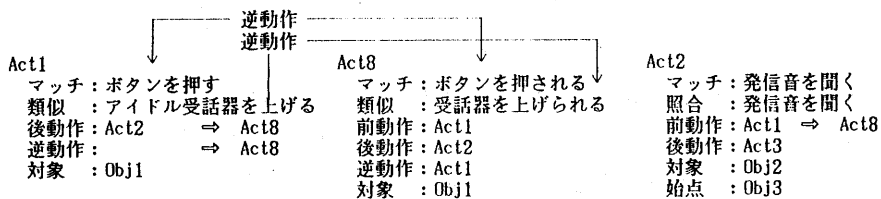
(1) 逆動作の生成

入力動作の中で、能動動作、受動動作であるような動作 A に対し、その逆動作になるような動作 B を生成し、A の主体スロットと制御スロット以外の各々のスロットとその値をコピーする。また、A がマッチ、照合、類似スロットに値を持つ場合は、この値の逆動作スロットの値を B の対応するスロットに入れる。更に、A と B を互いに、逆動作スロットで結び、前動作、後動作スロットの値を修正する。

(2) 要求サービスのシミュレートによる動作の補完と修正

要求サービスをシミュレートし、その過程で、不足の動作を類似サービスを参照して補完し、矛盾のある部分を修正する。また、修正が不可能であるような場合は、要求サービスが理解できなかったことになり、他の類似サービスの候補があれば、そちらについて動作の補完を試みる。実際にどのような修正を行うかについては、例を通して説明する。

3.4 節の類似サービス検索部での検索結果は図 7 に示されている。(1)の逆動作の補完手順に従うと、能動動作、受動動作であるような入力動作は、Act1,Act2,Act5 であり、逆動作となるような Act8,Act9,Act10 が生成され、各スロットが埋められ、値が修正される。例えば、Act8 の生成により、Act1,Act2,Act8 は、図 9 のようになる。

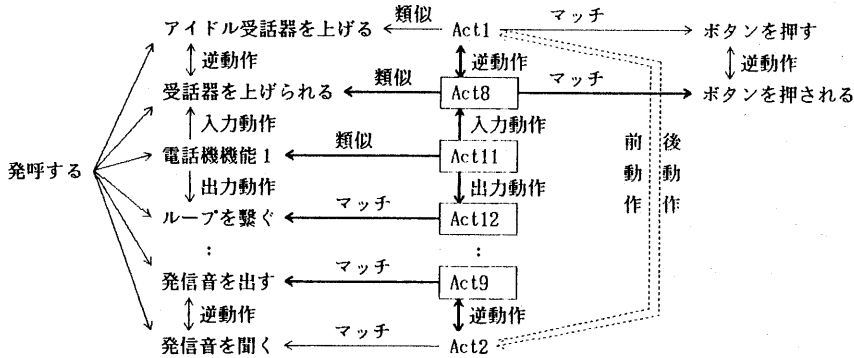


⇒：スロット値の変化、Act8はすべて新しい値なので省略

図 9: 逆動作の生成

次に、要求サービスのシミュレートが行われる。まず、類似サービス「電話をかける」の条件スロットの値が、初期状態としてセットされる。最初の動作は Act1 であるので、Act1 のマッチ先の「ボタンを押す」から継承された事前、事後スロットの値により、Act1 がシミュレートされる。Act1 の次の動作は Act8 であり、同様に Act8 がシミュレートされる。今、Act8 に対応する類似サービスの動作は、「受話器を上げられる」であり、その次の動作「電話機機能 1」に対応する動作は要求中にないので、Act11 が対応する動作として生成される。このとき、Act11 の入力動作スロットの値は Act8 になり、Act8 と「受話器を上げられる」は類似スロットで結ばれているので違う動作であることより、Act11 と「電話機機能 1」も違う動作と見なされ、これらが類似スロットで結ばれる。類似サービスの次の動作「ループを閉じる」については、それと対応して動作 Act12 が生成され、この場合は、同じ動作と見なされて、マッチスロットで結ばれる。

このようにして順次、動作が補完されていく。図 10 に動作の補完の様子を示す。



□: 新たに作られるフレーム、点線: 消滅する関係、太線: 新たに作られる関係

図 10: 動作の補完

さて、Act3 をシミュレートしようとしたとき、動作条件として受話器が上がっていることがあり、この時点では受話器が上がっていないので、このままではシミュレートできない。このような場合は、Act3 と照合している「ダイヤルする」という動作が複合動作であるので、部分動作に分解してシミュレートを行っていく。

最後に、Act4 と Act5 の周辺のシミュレートを考えてみる。既に述べたように、Act4 が「応答する」に照合する場合と、Act5 が「応答する」の部分動作「リング受話器を上げる」に照合する場合の 2 つの候補がある。後者の場合、Act4 は、Act3 のシミュレート後、Act5 のシミュレート前の中で、実行する条件が揃った時点でシミュレートされるので、「呼出す」と対応する動作が終わった時点でシミュレートされる。その結果、Act5 のシミュレート時に、「リング受話器を上げる」の動作条件に矛盾があり、この矛盾は修正できない。これに対し、前者は、Act4 のシミュレート後、発信側の受話器がまだ上がっていないのでシミュレート可能である。よって後者は棄却され、前者の候補が採用される。

このようにして、順次、補完、修正が行われる。

4 おわりに

本稿では 通信サービスのための要求理解手法について述べた。現在、本稿で述べた要求理解の手法に基づいて、システム化を行っている。現状では、パーザ、照合部については、lisp によって実現している。また、類似サービス検索部は、推論システム ART を用いて実現した。しかし、動作補完部については、より詳細な機能を検討しているところである。

今後、動作補完部の機能の整理とまとめと、知識ベースの拡充を行い、システムの完成と、それによる理解実験を行っていく予定である。

将来の課題としては、理解結果のユーザへの提示と要求サービスの修正のためのインターフェース、及び、知識獲得への応用などがある。

参考文献

- [1] 横田, 島 他: 「KANT における通信世界モデル」, 情報処理 学会第 35 回全国大会, 1987,
- [2] 横田, 島 他: 「知識ベースとの照合による情報補完法」, 情報処理学会第 38 回全国大会, 1989,
- [3] Bordia, A., Greenspan, S., Mylopoulos, J., "Knowledge Representation as the Basis for Requirements Specifications", *IEEE Computer*, pp.82-90, April 1985.
- [4] Minsky, M., "A Framework for Representing Knowledge", in *The Psychology of Computer Vision*, pp.211-277, P. Winston(ed.), McGraw-Hill, New York, 1975.
- [5] Schank, R.C., Abelson, R.P., *Scripts, Plans, Goals, and Understanding*, Erlbaum, Hillsdale, N.J., 1977.
- [6] Fillmore, C., "The Case of Case", in *Universals in linguistic theory*, pp.1-88, E. Bach and R. Harms (eds.), Holt, Rinehart, and Winston, New York, 1968.
- [7] Lenat, D., Prakash, M., Shepherd, M., "CYC: Using Common Sense Knowledge to Overcome Brittleness and Knowledge Acquisition Bottlenecks", *The AI Magazine*, vol.6, no.4, pp.65-85, 1986.