

# 音声対話システム構築フレームワーク EUROPA が参照する知識と その背景にあるオントロジー

## カーナビ音声対話システム MINOS への適用事例における考察

笹島 宗彦<sup>†</sup>, 屋野 武秀<sup>†</sup>, 河野 恭之<sup>‡</sup>

<sup>†</sup>(株) 東芝 研究開発センター ヒューマンインターフェースラボラトリー

〒 212-8582 神奈川県川崎市幸区小向東芝町 1

Tel:(044)549-2411, Fax:(044)520-1308

{munehiko.sasajima,takehide.yano}@toshiba.co.jp

<sup>‡</sup>奈良先端科学技術大学院大学

〒 630-0101 奈良県生駒市高山町 8916-5

Tel:(0743)72-5261, Fax:(0743)72-5269

kono@is.aist-nara.ac.jp

### 1 はじめに

近年, 様々な家電製品のインターフェースとして音声が目ざされている. 音声対話システム [竹林 94][山本 95][永井 97] や音声対話を統合したマルチモーダルシステム [河野 98][中川 98a] など関連する研究が行なわれる一方で, その課題なども明らかになりつつある [中川 98b][新美 98] [山本 98][竹澤 98]. 音声対話システム評価方式には現在様々なものがあるが [石川 98], 筆者らは次の3つの視点に基づくものを考えている.

- (1) 入力の満足度: ユーザが入力文パターンの種類について満足できるか
- (2) システムの即応性: どれだけ早く回答できるか
- (3) システム応答の正確性: どれだけユーザの発話意図に沿った回答ができるか

しかしこれら3つの視点は独立でなく, 特定の視点のみに沿った性能向上は難しい. 例えば入力の満足度を高めるために受理可能な文パターンを増やせば, 音声認識される文候補も増え, 後段の問題解決部の負荷が高まってシステムの即応性が損なわれる. つまりシステムの構築や性能向上にあたっては, 3つの視点からの総合的評価と機能の取捨選択が必要である. その際のガイドラインとして, オントロジーの分析が有効である [溝口 99][B.Chandrasekaran 99]. ユーザが発話によって参照する概念やシステムが参照する知識ベースの背景にあるオントロジーを明示化したものは, システム構築の際の指針となる.

オントロジーの構築方法には, EDR [EDR 93] や広辞苑のような汎用概念辞書をトップレベルとしたトップダウンの方法によるものと, ドメインとタスクの観察に基づくボトムアップのもの2通りが考えられる. オントロジー構築の目標は, 各知識の背景にある概念集合とその構造を, 少数の概念素の組合せとして表現することである. しかし EDR や広辞苑は概念の還元的定義を行なっておらず, そういった概念素を提示していない. 筆者らは, 音声対話システムの背景にある概念素を同定するには, トップレベルの概念階層を意識しつつボトムアップのモデル構築事例を積み重ねて, そこからトップレベルとのギャップを埋める作業を行なうのが効率的であると考えている. モデル構築を効率的に行うためには, システム構築のためのできるだけ汎用なフレームワークが必要である.

以上の背景に基づき筆者らは, 音声対話システム構築フレームワーク EUROPA (Environment for building Utterance Recognizable PACKages) を開発し, そのカーナビゲーションタスクへの適用システムである MINOS (Mobile Interactive NavigatiOn System) を試作した [Sasajima 99]. また, これらの構築を通じて, 音声対話システムが参照する知識を整理し, その背景にあるオントロジーについて考察した. EUROPA は大きく分けて文法, 発話者意図, 問題解決器が参照するドメイン固有知識, 問題解決手順, 応答生成知識の5つの知識を参照する. MINOS の試作においてはこれら5つの知識についてオントロジーを検討し, 実装に反映させた. MINOS をはじめとして適用事例を積み重ねる事によって各知識の背景にあるオントロジーを明らかにできると思われる. 本稿では EUROPA フレームワークと MINOS の試作過程で行ったオントロジーの観点からの考察について述べる.

### 2 音声対話システムとオントロジー

オントロジー的分析の手法は従来, 複雑な人工物を対象としたプラント運転員の知識モデル構築 [古田 99] やプラントの故障診断 [來村 99], スケジューリング [Miyashita 95], タスクオントロジーの構築支援 [堀 99][瀬田 98], エキスパートシステム構築支援 [和泉 99] など, 専門家自身も解決手順を簡単に説明できないようなタスクに適用されてきた. 音声対話タスクにこの分析を適用することは, 少なくとも知識の実装と即応性の向上における問題の解決に有効と思われる.

#### 2.1 知識の実装とオントロジー

問題解決システムとして見た場合, 音声対話システムは他と比較して全体の性能向上を単純に実現できない特徴をもつ. 筆者らは, 音声対話システム評価の視点を (1) 入力の満足度, (2) システムの即応性, (3) システム応答の正確性, の3つと考えている. しかしこれら3つの視点は独立ではない. 例えば, ユーザが可能な入力文パターンを

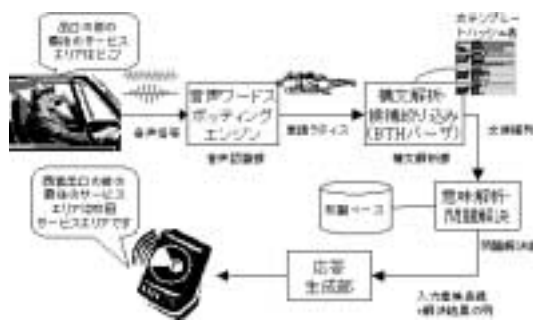


図 1: EUROPA ブロック図

単純に増やすことで多様性を高め満足度を向上させることができる。しかし、入力文パターンを増やすことは同時に認識誤りや非文、すなわち、文法的に正しくとも意味的に間違った文候補の増加につながり、問題解決器へと出力される文候補の増加を招く。結果として問題解決に必要な時間が増え、システムの即応性が失われる。また、入力文パターンを増やすと、文パターン毎に記述する問題解決知識も増え、システムの正確性を維持するためにそれらを維持するコストが高くなる。つまり、ドメインやタスクを深く分析して単純に記号化しても、音声対話システムは良くならない。3つの視点から総合的に評価して記号化すべきものとそうでないものを取捨選択する必要がある。そのためのガイドラインとして、オントロジーの分析が有効である [溝口 99][B.Chandrasekaran 99]。例えば即応性が損なわれない程度に文パターン数を調整するとき、システムのユーザにとって重要な概念を参照する発話文パターンは削除することができない。あらかじめユーザがシステムを使う場面を数え挙げ、ユーザ発話が参照する概念の体系を明示化すれば、その指針を得られる。

## 2.2 即応性の向上とオントロジー

音声対話システムのもう1つの特徴として、解決する問題の質よりも即応性が重視される点を挙げるができる。一般に問題解決システムのユーザは、解決する問題の質と計算量や応答時間など解決コストの2つの観点からシステム性能を評価することが多い。上述の故障診断やスケジューリングなど、従来オントロジーの手法が適用されてきたタスクなどは、問題の質について重く評価される傾向にある。その一方、音声対話は、誰もが日常的にこなすタスクであり専門家である為、問題の質は軽く評価されがちである。すなわち音声対話システムの場合、解決コスト、特に応答時間が重く評価される。例えば、複雑な質問に時間をかけて応答するシステムよりも、それが解決できなかった旨を短時間でユーザに通知するものの方がユーザに高く評価されることは珍しくない。音声対話のアプリケーションであるカーナビゲーション(以下、カーナビ)システムなど、1秒応答が遅れるだけでその内容が無意味になってしまうシステムはその典型例である。

即応性が重視されるため、音声対話システムはドメインやタスクに特化されたものになりやすい。音声対話システム開発時、特に初期段階では、どのような知識表現を用いれば良いかが見えていないため、知識ベースに関する処理部分と知識そのものを修正する作業が多くなる。それらは主に知識の単純化や手続きへの変換などであるが、高速化の為の無秩序な修正は、汎用な知識を手続きに埋没させてしまったり知識の前提を見失わせたりする原因となる。そのようなシステムは、他タスクやドメインへの転用だけでなく、さらなる高速化や機能拡張、大規模化などの改良も難しくなる。これを避ける為には、汎用な知識とドメインやタスクに固有のものを明示的に区別することが必要である。音声対話システムの構成モジュールが参照する各々の知識について、その背景にあるオントロジーを明らかにすることは、これら知識の明示的区別のために有効と考えられる。

## 3 EUROPA: 音声対話システム構築フレームワーク

EUROPA は対話システムが参照する知識を汎用なものとしてそうでない物に分離することによって個別システム構築の効率を高めることに貢献する。また、ドメイン固有の問題解決手続きなどシステム管理の為に可読性が重要なものについてはスクリプトによる記述を採用することによって可読性を高めている。さらに、スクリプトの解釈をより高速化する為に、スクリプトをあらかじめ中間言語にコンパイルして仮想CPUで解釈実行する方式を採用している。

図1に、筆者らが提案する音声対話システム構築フレームワーク EUROPA のブロック図を示す。各ブロックが参照する知識は大きく分けて、文法、発話者意図、問題解決器が参照するドメイン固有知識、問題解決手順、応答生成知識の5つに分けられる。

- 文法 文法は主に音声認識部と構文解析部によって参照される。音声認識部では、音声ワードスポッティングエンジンが認識語彙辞書を参照して、入力音声信号からキーワードラティスを生成する。認識語彙辞書の要素である各語彙を、キーワードと呼ぶ。各キーワードは1つ以上の品詞に所属しており、キーワードと品詞の階層関係を定義したものを品詞辞書とよぶ。
- 構文解析部は文法を参照し、単語ラティスから正しいキーワードの並びを全て抽出する。文法は文型の有限集

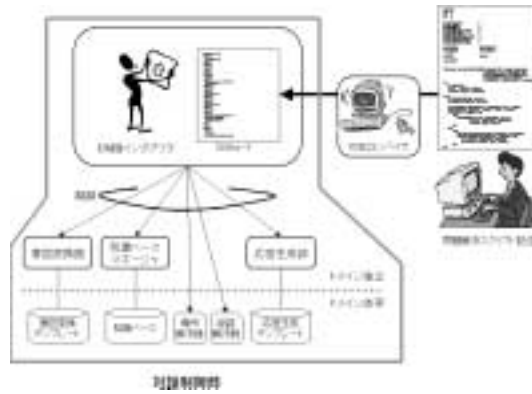


図 2: ドメイン固有の知識と汎用知識の分離

合と品詞辞書によって定義される。個々の文型は、品詞または句の並びと文型 ID からなり、句は、品詞の並びと句 ID からなる。一般に音声認識結果は曖昧であり、構文解析部の出力は、入力された発話文の候補列となる。

文法の構築は、ドメインとタスクの分析に基づいて行なう。まず、対象となるドメインにおいて、入力として受理すべき発話例を数え挙げ、そこから文型と単語を抽出し、辞書を構築する。

**発話者意図** 意味解析部によって発話文候補列は、各候補に対応する話者の問題解決意図を表現したデータ構造の列に変換される。各々のデータ構造をユーザ意図表現、あるいは意図表現と呼ぶ。各文型に所属する入力文候補をどのように変換すればよいかは意図変換テンプレートの形式であらかじめ定義されており、意味解析部の構成要素である意図変換器がこれを参照する。

ユーザ意図表現の構築には、発話表現に込められた意図を数え挙げ、それに沿った問題解決を実現するための問題解決器を設計し、動作時に参照するデータを全て意図表現に組み入れる必要がある。このため、各々の文型定義に対応する意図表現の構築はドメインモデル、問題解決器の設計と並行して行う。

**ドメイン固有知識** 受理した文に対応する問題解決手順の設計に先だって、問題解決器が参照するドメイン固有知識を設計する。具体的には、各ユーザ発話が意図する問題解決において、参照される物の概念を階層的に分類する。

**問題解決手順** 一連の音声対話処理に含まれる手続きや知識には、構文解析や知識ベースの運用など汎用なものと同様にドメイン固有知識やタスク固有の問題解決手順など固有のものが両方含まれる。問題解決手順とはこれらをどのように組み合わせて処理するかを定義するものであり、問題解決部によって参照される。解決手順の知識はタスク固有であり、再利用可能な知識とは分離する必要がある。また、解決手順の複雑さは問題の複雑さに比例するので、知識自身の可読性を高めることも重要である。EUROPA ではこれを、筆者らが開発した USHI(Unification-based Script Handling Instruction set) スクリプト言語によって記述する。

**応答生成** 問題解決結果をどのような文の形にしてユーザに伝えるかを定義するのが応答生成テンプレートであり、応答生成部によって参照される。

EUROPA フレームワークにおいて以上の知識は、ドメイン固有の知識とそうでないものに明示的に分離されている。図 2 は、本研究が対象としたカーナビタスクにおける知識の分離を概念的に示したものである。ドメイン依存のグループには、意図変換テンプレート、ドメイン固有知識の集合である知識ベース、カーナビタスク固有の問題解決器である場所解決器や経路解決器、応答生成テンプレートなどが含まれる。もう 1 つのドメイン非依存のグループには、意図変換器、知識ベースのマネージャ、構文解析器、問題解決手順を定義するスクリプト、ドメイン依存データのそれぞれに対する解釈手続きなどが含まれる。また、図 2 に示すように、スクリプトはあらかじめ高速解釈が可能な中間コードにコンパイルされ、動作時には USHI インタプリタによって解釈される。これによって、問題解決手続き設計者にとっての可読性を維持しつつ、システム全体の高速処理を実現している。

#### 4 カーナビタスクにおける知識構築

筆者らは EUROPA をカーナビ音声対話システム MINOS の構築に適用した。本章では各知識構築について、方針の概要と適用の実例を交えて述べる。MINOS は、あらかじめ設定された出発地点から目的地点までの経路について、音声対話で情報を検索できる。具体的には、経路上にある施設の場所と、時刻と共に変化する自車の現在位置を含めた任意の 2 地点間の距離について、質問に答えることができる。



運転者 (D と表記) による質問の例

- D1-1: 次のサービスエリアはどこ？
- D1-2: 豊中インターまでどれくらい？
- D1-3: 出口の前の最後のサービスエリア教えて
- D1-4: トイレはどこ？
- D1-5: ポートタワーの前にコンビニある？

運転者とナビまたは同乗者 (N と表記) との対話例 1

- D2-1: 次の交差点は右折でいいの？
- N2-1: うん、そう。
- D2-2: そこまで何分？
- カーナビそのものの操作
- D3-1: もっと詳細に表示して
- D3-2: 目的地付近を見せて

図 3: 運転者の発話例



図 4: ユーザ発話の階層的分類 (一部)

### 4.1 文法辞書

カーナビ音声対話システムを構築するにあたって、まず、運転者が運転中にナビシステムや同乗者に問い合わせを行なう場合の発話例を約 300 収集し、その分析を行なった。図 3 に収集された発話例の一部を示す。図 3 に示されるように運転者が音声を利用する行為は、D1-1、D1-2 などのように未知の情報を取得するもの、D2-1 のようにカーナビや同乗者が持つ問題解決モデルを取得するもの、D3-1、D3-2 のように情報を取得するために機器そのものを操作するものの 3 種類に分類することができる。図 4 は、これらユーザ発話を階層的に分類したものの一部である。運転者が取得を意図する未知情報は 5W2H の観点から分類され、さらにそれらが文型に細分類されている。

次に、例文とユーザ意図の分析を元に、語彙辞書と文テンプレートを並行して作成した。本タスクにおいて、語彙辞書の規模は単語数 1002 であり 302 個の品詞に分類されている。ただし、品詞によっては全く単語が所属しないもの、逆に非常に多くの単語が所属するものがある。傾向としては、問題解決可能な文型を構成する品詞には多くの単語が所属する。これは後述する問題解決部の実装が進むに連れて解決可能な文の形態も多様になるからであろう。

次に語彙辞書の品詞を利用して、文型を定義した。各々の文型は文型 ID と品詞あるいは句の並びで定義される。図 5 にその一部を示す。本タスクにおいてはユーザの場所に対する参照表現が非常に多岐に渡っていたため、句として「場所表現句」を導入している。場所表現句の数は 240、文テンプレートの数は 92、文テンプレートと場所表現句の組合せパターン数は全部で 200 万程度になる。対話システム構築者が定義したこれら文法は、ハッシュ検索が可能となるように変換され、対話システム動作時に参照される。

### 4.2 ドメイン固有知識

カーナビタスクにおいて問題解決手順が参照する概念を階層的に整理した。概念としてはサービスエリアなど道路施設、各種道路、道路を構成する区間の概念などがある。例えばサービスエリアやパーキングエリアなど道路施設

文定義

- 文型 ID: Where1
- 品詞列: < 場所表現句 > (疑問詞 Where)
- 文型 ID: Where2
- 品詞列: (疑問詞 Where) < 場所表現句 >

...

場所表現句定義

- 句 ID: P1
- 品詞列: (地名)
- ...
- 句 ID: P20
- 品詞列: (経由地)(相対位置)(絶対位置)(施設)

図 5: 文法定義例

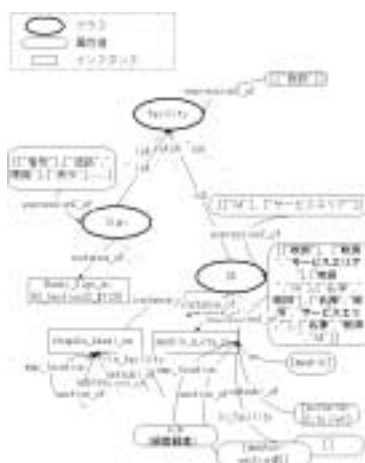


図 6: 道路施設の概念 (部分)

や、トイレ、駐車場など道路設備の概念を合わせると約 40 ある。さらに、本研究が対象とするカーナビタスクでは設定された経路上の施設などについてもそのモデルを問題解決時に参照する。施設などのモデルは上記約 40 の概念を具体化して得られ、実行時に参照されるモデルの数は、ユーザが設定した経路の長さや施設の集中度合に比例する。一例として、関西の都市部で設定された高速道路を含む約 40 キロメートルの経路の場合、交差点、道路、ガソリンスタンド、コンビニなど道路脇の主要な施設を合わせた数は約 150 程度になる。道路施設モデルの一部を図 6 に示す。ドメイン固有知識は、素性構造を用いて表現する。カーナビタスクにおいて道路や道路施設などは、大きく分けて (1) 固有名称、(2) 所属するクラス、(3) 自転車との位置関係、(4) 他の施設や地点との位置関係、(5) 付随する設備、の 5 つの属性を用いて参照される。例えば「名神吹田サービスエリアはどこ」のような固有名称を用いた参照は (1) の属性値を参照して解決し、「サービスエリアはどこ」の場合は (2) を参照する。(3) は例えば「次のサービスエリアはどこ?」、(4) は「吹田インターの次のサービスエリアはどこ?」のような参照表現の解決に必要である。またカーナビシステム動作時に与えられる外部入力は、ユーザが設定した出発地から目的地までの経路情報と、自転車の位置情報を GPS(Global Positioning System) によって検出したものであり具体的には緯度経度の値として表現される。(3) と (4) に属する参照表現を解決するために、発話によって参照される全てのものは緯度経度の属性を備えていることが必要である。(5) は例えば発話表現「トイレはどこ?」におけるトイレのことであり、これを解決する為に、施設や道路に付随する設備を表現する属性も必要である。

### 4.3 ユーザ意図表現

意図変換処理は、入力された発話文候補の文 ID と文に含まれる句の ID をキーにして意図表現辞書から素性構造であるユーザ意図表現のテンプレートを引き、発話文の単語、品詞、句の情報を利用してそれを初期化する。意図表現テンプレートには文型毎の定義と句表現毎の定義があり、これらは、(1) 辞書引きのためのテンプレート ID (2) 問題解決部が入力として参照するデータである問題解決部入力 (3) テンプレート自身を発話表現によって初期化する初期化命令列 (4) 音響スコアなど、音声認識結果についてのデータである音響情報、の 4 つからなる。図 7 にテンプレートの具体例を示す。文型 ID が Where1 である文候補が入力されると、意図表現辞書から上記のテンプレートが検索される。初期化命令属性の値は文字列のリストである。例えば文字列“(3) 全文品詞列”は、この組成構造の 3 番目のスロットに全文を品詞列で表現したものを埋める命令を表現している。属性が値として素性構造をとる場合もあり、そのようなスロットを初期化する命令の例が“(4, 1, 1) 相対位置:1”である。この例のように、命令列の先頭のスロット指定表現がリストである場合に付いて説明する。簡単のため、例えば“(1, 2)”というリスト表現は、この組成構造の 1 番目のスロット値である組成構造の 2 番目のスロットを表現する。

### 4.4 問題解決手続きの記述

ユーザの 1 発話に対する問題解決手続きでは、ドメイン依存、非依存の各データと手続きを適切に組み合わせることが必要となる。例えばカーナビゲーションタスクでユーザがある条件を満たす場所の検索を要求する発話をした場合を考える。発話から生成されたラティスの構文解析ではドメイン独立なパーザが使用されるが、構文解析結果からユーザの意図を抽出し、その意図に沿った問題解決を実行するにはドメイン固有の問題解決器が使用される。図 2 に示すように、EUROPA では、スクリプトによる問題解決手順記述方式の採用によって対話システム構築者に対しての手続きの可読性を高めており、さらに問題解決手順を他のコードから分離することによって構築された対話システムの汎用性を維持することが実現されている。またあらかじめ中間コードにコンパイルする方式によってシステム動作の高速化を実現している。図 8 にスクリプトの例を示す。USHI スクリプト言語は Pascal に似た表記法を持ち、Pascal が提供する IF 文 (10 行目から 21 行目) や For 文による制御構造の一部、四則計算と比較のための各種演算子、スクリプト内部及び C++ で記述された手続きの呼び出し (12, 17 行目) などが可能である。データ構

素性構造名:UserIntention  
 文型 ID: Where1  
 発話品詞列: nil  
 発話単語列: nil  
 検索情報: PlaceName  
 問題解決部入力:nil  
 初期化命令: [(2) 全文品詞列, (3) 全文単語列, (4, 1, 1) 相対位置:1, . . . ]  
 音響スコア: nil

図 7: 初期化前の意図表現

```

1: PSPProcedure Int CALL_HOW_MUCH(PredicateKB pKB ,
2:     Predicate Answer , . . . < 引数略 >)
3: Var
4:   Int      nFlag , . . . < 変数定義略 > , nDefaultSC;
5:   String   szCurrSec , . . . < 変数定義略 > , szTarget;
6:   Predicate predBody , . . . < 変数定義略 > , predPlaceExp;
7: Begin
8:   nFlag := GET_CURRENT_SECTION(pKB , szCurrSec);
9:   GetArgVal("GType_Any" , predBody , "Target" , szTarget);
10:  If(szTarget == SZ_AMOUNT) Then
11:    Begin
12:      nFlag := SOLVE_HOWMUCH1_1_TYPE(pKB ,
13:        IntentionExp , . . . < 引数略 >);
14:    End
15:  Else
16:    Begin
17:      nFlag := SOLVE_HOWMUCH1_0_1_TYPE(pKB ,
18:        IntentionExp , . . . < 引数略 >);
19:    End
20:  < 以下省略 >
21: End

```

図 8: 問題解決手順のスク립ト記述 (一部)

造として特徴的なものには、各種知識を表現するための素性構造とその操作の為の各種手続きが挙げられる。例えば 9 行目は、素性構造名が “ GType\_Any ” であるデータ predBody の、スロット名 “ Target ” から値を取り出して変数 szTarget を初期化する、という手続き GetArgVal の具体例である。その他、スク립ト記述によって素性構造どうしの単一化や比較、それらの知識ベースへの入出力手順などを定義することが可能である。

#### 4.5 カーナビタスクにおける問題解決

BTH パーザから文候補の集合を受理すると、対話制御モジュールは文候補の集合から 1 つずつ候補を取り出し、意図変換器によって対応するユーザ意図表現を生成する。各ユーザ意図表現はさらに問題解決器に入力され、解決される。ユーザ意図表現、地理データなどは述語の形式で表現されており、それらの探索、比較、初期化などの処理には素性構造の単一化手法が用いられている。

カーナビタスクを対象とする音声対話システムにおいてもドメインとタスクの両方に対する修正要求が発生する。前者の例としては、レストランとガソリンスタンドの場所について質問できるインターフェースをテストしたユーザからの、コンビニエンスストアの場所も質問できるようにしてほしいという要求が挙げられる。また後者としては、様々な地点の位置を質問できるシステムをテストしたユーザからの、2 地点間の距離も質問できるようにしてほしいというものが挙げられる。EUROPA フレームワークはドメインとタスクに対する独立性が高く、このような要求に基づくシステムへのフィードバックのコストは低くなっている。4.4 節で述べたように、EUROPA フレームワークに基づく音声対話システム構築では、ドメインに依存して定義される手続きやデータのグループとドメイン非依存のモジュールとデータのグループをわけて実装する。カーナビ音声対話の場合、前者には、カーナビシミュレータアプリケーション部に対する入出力手続き、カーナビドメインに特化した単語辞書と文法辞書、意図表現の変換ルール、回答文の生成ルール、場所参照表現解決器や 2 地点間経路探索器など各種の問題解決器、ドメイン依存データのそれぞれに対する解釈手続き、発話入力に対する問題解決手順のスク립ト記述などが含まれる。後者はドメイン非依存

```

素性構造名:UserIntention
  文型 ID:      Where1
  発話品詞列:  [経由地, 相対位置, 絶対位置, 施設, 疑問詞 Where]
  発話単語列:  [出口, 前, 最後, サービスエリア, どこ]
  検索情報:    PlaceName
  問題解決部入力:素性構造名:PlaceExpression
                  場所表現 ID:P20
                  修飾表現: 素性構造名:PRExp
                              相対位置: 前
                              絶対位置: 最後
                              場所クラス:サービスエリア
                              基準点表現:素性構造名:PRExp
                              場所クラス:出口
                              . . . < 以下略 > . . .

  初期化命令:  [(2) 全文品詞列, (3) 全文単語列, (4, 1, 1) 相対位置:1, . . .]
  音響スコア:  800

```

図 9: 発話入力による意図表現の初期化例

のモジュールとデータのグループで、音韻データの辞書、ラティスのパーザ、知識ベース操作の基本プリミティブ、素性構造操作のためのプリミティブなどが含まれる。MINOS の試作において、例えば上記コンビニエンスストア検索への拡張問題は、単語辞書と文法辞書、ドメイン依存データである地図データへのコンビニ情報の追加によって実現された。また 2 地点間の距離計算への拡張は、問題解決手順のスクリプト記述変更によって実現された。言い換えれば、ドメインやタスク固有の機能拡張はそれぞれ固有の知識のみの変更で実現可能であり、汎用かつ対話システムを中心とするモジュール群を再構築する必要がない。この点で、システムに対する変更要求のフィードバックは容易であると言える。

#### 4.6 対話処理の例

カーナビゲーションタスクにおける対話処理の過程について、対話制御部に入力された発話文候補が「出口の前の最後のサービスエリアはどこ」であった場合を例として説明する。発話文候補には文型 ID や品詞情報などの構文情報と、音響スコアなど音響情報が付加されている。

次に、文型 ID をキーにして、意図変換器は意図表現テンプレートと句表現のテンプレートを検索する。得られたテンプレートの初期化命令に従って、意図変換器は意図表現を図 9 のように初期化する。

初期化された意図表現を入力として意味解析と問題解決が行われる。まず、検索情報属性の値 "PlaceName" から、この意図表現はある制約を満たす地点の検索を意図するものであることが判定される。地点検索発話に対する処理は次の 3 つのステップからなるスクリプトによって制御される。

ステップ 1: ユーザ意図表現から、検索対象場所の制約条件を抽出する

ステップ 2: 場所の制約を引数として場所解決器を呼び出す

ステップ 3: 場所解決器が制約を充足する場所の検索に成功したら、ユーザが質問に用いた表現と解である場所を引数として応答生成のためのテンプレートを選択し、質問表現、解、テンプレートを引数として「応答生成部」を呼び出す。

はじめにステップ 1 で、図 9 の意図表現から問題解決部入力スロットの値を取り出す。この値は、ユーザ発話のうち「出口の前の最後のサービスエリア」に対応し、場所の制約条件を表現している。

次にステップ 2 で、検索条件を引数としてドメイン固有の問題解決器である場所解決器を呼び出す。場所解決器は場所検索条件、自車位置、地理情報、設定ルート情報を入力として、設定ルート上で現在地に対して「出口」と呼べる基準点を検索し、その基準点と現在位置の間で最も基準点側に近いサービスエリアと呼べる場所を検索する。

仮に、基準点である出口として名神高速西宮出口が、また基準点に対する残りの制約を満たす場所として名神高速吹田サービスエリアが検索されたとする。場所解決器から検索成功の通知を受けると、スクリプトの解釈はステップ 3 に進む。ステップ 3 では、場所の名称を要求する入力意図、質問に用いた表現、自車位置、検索されたサービスエリアの位置をキーにして応答文生成のためのテンプレートを検索する。さらにテンプレートをその検索に用いたキーワード、自然言語文の体裁を整えるための助詞や補助的な言葉で整える。応答文としては、例えば「西宮インターの前の最後のサービスエリアは、吹田サービスエリアです。」

が生成される。また自車位置が吹田サービスエリアを通過していた場合には、応答文の最後に「既に通過しています」が追加される。最後にテキストデータである回答文は、TTS エンジンに入力され、合成音に変換されて出力される。TTS としては、TOS Drive TTS[Kagoshima 98][Seto 98][籠島 00] の PC 版である東芝音声システムを用いている。



## 5 まとめ

音声対話システム構築の汎用フレームワークである EUROPA について，システムを支える知識の観点から述べた．タスクやドメインが変わった時にどの知識を修正すべきかが明示的になっており，またスクリプトの援用によって知識の可読性が高められているので，EUROPA はこれらの点において汎用フレームワークであると言える．しかし個々の知識構築や修正は現在のところ，不完全なオントロジーをガイドラインとして手で書き起こすしかない．EUROPA を知識の構築や修正の支援までも含めた真の汎用フレームワークにするためには，異なるタスクやドメインへの適用事例を積み重ねて，各知識を支えるオントロジーを見極めることが必要である．

## 参考文献

- [B.Chandrasekaran 99] B.Chandrasekaran, J.R.Josephson, and R.Benjamins, : Why are ontologies, and why do we need them?, *IEEE Intelligent Systems*, pp. 20–26 (1999).
- [EDR 93] EDR 電子化辞書仕様説明書, (株) 日本電子化辞書研究所 (1993).
- [Kagoshima 98] Kagoshima, T., Morita, M., Seto, S., and Akamine, M.: An F0 Contour Control Model for Totally Speaker Driven Text to Speech System, in *ICSLP'98*, pp. 1975–1978 (1998).
- [Miyashita 95] Miyashita, K. and Sycara, K.: CABINS: A framework of knowledge acquisition and iterative revision for schedule improvement and reactive repair, *Artificial Intelligence*, Vol. 76, No. 1-2, pp. 377–426 (1995).
- [Sasajima 99] Sasajima, M., Yano, T., and Kono, Y.: EUROPA: A Generic Framework for Developing Spoken Dialogue Systems, in *EuroSpeech'99*, pp. 1163–1166 (1999).
- [Seto 98] Seto, S., Morita, M., Kagoshima, T., and Akamine, M.: Automatic Rule Generation for Linguistic Features Analysis Using Inductive Learning Technique, in *ICSLP'98*, pp. 2031–2034 (1998).
- [永井 97] 永井, 石川: 対話システムのための概念素理解方式における対話音声理解, in *97-SLP-7* (1997).
- [河野 98] 河野, 屋野, 池田, 知野, 鈴木, 金澤: ATMS ベースのマルチモーダル入力統合方式を用いたインタフェースエージェントシステム, *人工知能学会誌*, Vol. 13, No. 2, pp. 212–220 (1998).
- [古田 99] 古田, 吉川: 行動発話分析によるプラント運転員の知識モデル構築, *人工知能学会誌*, Vol. 14, No. 6, pp. 989–1000 (1999).
- [溝口 99] 溝口, 池田, 來村: オントロジー工学基礎論, *人工知能学会誌*, Vol. 14, No. 6, pp. 1019–1032 (1999).
- [山本 95] 山本, 伊藤, 肥田野, 中川: 人間の理解手法を用いたロバストな音声対話システム, *情報処理学会論文誌*, Vol. 36, No. 4, pp. 471–482 (1995).
- [山本 98] 山本幹雄: 音声対話データベースの現状, *日本音響学会誌*, Vol. 54, No. 11, pp. 797–802 (1998).
- [新美 98] 新美康永: 音声対話システムの対話制御, *日本音響学会誌*, Vol. 54, No. 11, pp. 791–796 (1998).
- [瀬田 98] 瀬田, 池田, 角所, 溝口: 問題解決オントロジーの構成 - スケジュールタスクオントロジーを例にして -, *人工知能学会誌*, Vol. 13, No. 4, pp. 597–608 (1998).
- [石川 98] 石川泰: 音声対話システムの評価, *日本音響学会誌*, Vol. 54, No. 11, pp. 807–811 (1998).
- [竹林 94] 竹林: 音声自由対話システム TOSBURG-II, *信学論*, No. 8, pp. 1417–1428 (1994).
- [竹澤 98] 竹澤寿幸: 対話音声文法の構築, *日本音響学会誌*, Vol. 54, No. 11, pp. 803–806 (1998).
- [中川 98a] 中川, 傅田, 伊藤: マルチモーダル観光案内対話システム, *人工知能学会誌*, Vol. 13, No. 2, pp. 241–251 (1998).
- [中川 98b] 中川: 音声対話システム構築の課題, *日本音響学会誌*, Vol. 54, No. 11, pp. 783–790 (1998).
- [堀 99] 堀雅洋: ドメイン・オントロジーに基づく問題解決メソッド群の組織化 - 実運用環境への適用と評価 -, *人工知能学会誌*, Vol. 14, No. 6, pp. 1041–1050 (1999).
- [和泉 99] 和泉, 丸山, 鈴木, 山口: エキスパートシステム開発のための再利用可能メソッドライブラリの設計と実装, *人工知能学会誌*, Vol. 14, No. 6, pp. 1061–1071 (1999).
- [來村 99] 來村, 西原, 植田, 池田, 小堀, 角所, 溝口: 故障オントロジーの考察に基づく故障診断方式 - 網羅的故障仮説生成 -, *人工知能学会誌*, Vol. 14, No. 5, pp. 838–847 (1999).
- [籠島 00] 籠島, 赤嶺: 閉ループ学習に基づく最適な音声素片の解析的生成, *信学論 (D-II)* (2000), 掲載予定.