

## PRINTEPSにおける音声対話システム開発環境

## Development Environment of Spoken Dialogue System Based on PRINTEPS

西村 良太\*<sup>1</sup>      高瀬 裕\*<sup>2</sup>      中野 有紀子\*<sup>3</sup>  
 Ryota NISHIMURA    Yutaka TAKASE    Yukiko NAKANO

\*<sup>1</sup>慶應義塾大学大学院理工学研究科

Graduate School of Science and Technology, Keio University

\*<sup>2</sup>\*<sup>3</sup>成蹊大学工学部情報工学科

Department of Computer and Information Science, Faculty of Science and Technology, Seikei University

In this paper, we describe the development of a spoken dialogue system based on PRINTEPS architecture. This spoken dialogue system is composed of five modules (speech recognition, language understanding, dialogue management, response generation, speech synthesis). In PRINTEPS, when calling the spoken dialogue system, system developers specifies the small-scale dialogue goal. The system performs a dialogue with the user in order to obtain necessary information from the user. Dialogue processing rules corresponding to the dialogue goal is prepared in advance. The advantage of PRINTEPS-based system development is that system developers can build a spoken dialogue system without the knowledge of spoken dialogue system.

## 1. はじめに

現在、我々は実践知能アプリケーションを効率的に開発するプラットフォームである PRINTEPS (PRactical INTElligent aPplicationS)[山口 15, 森田 16] を研究開発している。PRINTEPS で開発されるアプリケーションは、人と機械が相互に連携して、お互いの知識を高めていく「知能共進化」を進める。このアプリケーションでは、人と機械のインタラクションが不可欠であり、このインタラクションを容易に実現するため、我々は PRINTEPS アーキテクチャに基づく音声対話システムを開発した。

本研究では、PRINTEPS のアーキテクチャに則り、各種センサ・ロボット・知識ベースを活用し、人間と対話を行う音声対話システムを対話システムについての専門的な知識を持たない人でも容易に開発できる環境の実現を目指す。実践知能アプリケーション開発者は、PRINTEPS を利用し、目的のシステムをワークフローにて記述でき、細かいプログラミングなしにアプリケーションを開発できる。この仕組みによって、音声対話システムもアプリケーションに導入できるようにする。PRINTEPS は、ROS (Robot Operating System) \*<sup>1</sup> ベースのアーキテクチャであることから、センサやロボットなどの親和性が高く、音声対話機能を持たないしゃべらないロボットも、PRINTEPS と音声対話システムによって容易に音声を用いてロボットへの指示や確認を行うことができるしゃべるロボットにすることができる。

本稿では、PRINTEPS アーキテクチャに基づき開発された、音声対話システムについて述べる。まず 2. 節で PRINTEPS の概要を説明し、3. 節で PRINTEPS に導入した音声対話システムの詳細を述べる。4. 節で PRINTEPS への音声対話システム導入の重要性を述べ、5. 節でまとめを述べる。



図 1: PRINTEPS ワークフローエディタ

## 2. PRINTEPS

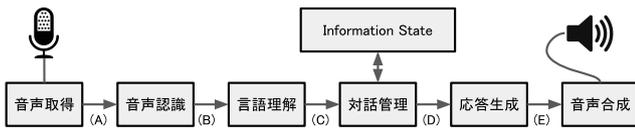
PRINTEPS は、人と協働可能な実践知能アプリケーションを容易に開発できることを目指したプラットフォームである。PRINTEPS は、専門知識を持つ開発者向けではなく、エンドユーザが設計段階から参加し (ユーザ参加型デザイン)、要素知能 (ソフトウェアモジュール) を統合して総合知能を容易に (数時間から数日で) 開発できることが特色であり、総合知能の開発・実践によって知の共進化を起こすことを目標としている。

PRINTEPS による実践知能アプリケーション開発は、ワークフローを記述することで行う。ワークフローは、プログラムやロボットアプリケーション開発の専門知識を持たない一般ユーザが記述することを想定しており、GUI のインタフェースにて記述が可能である (図 1)。PRINTEPS では、アプリケーション開発に必要な最小機能 (ロボット検索, 年齢認識, 音声認識, ロボット移動など) をモジュールとして用意し、これを組み合わせていくことで実践知能アプリケーションを構築する。

PRINTEPS の各モジュールは、1 つの ROS プログラムとして動作して (もしくは ROS のネットワークで通信ができるようにラップされて) おり、それぞれのデータのやり取りも ROS の通信の仕組みを利用している。ROS は分散システム指

連絡先: 西村 良太, 慶應義塾大学大学院理工学研究科, 〒 223-8522 神奈川県横浜市港北区日吉 3-14-1, nishimura@ae.keio.ac.jp

\*<sup>1</sup> <http://www.ros.org/>



**流れているデータ**

- A: 音声波形データ
- B: 自然言語文 「3名です。」
- C: ユーザ入力の意味表現 「Num, 3」
- D: システム応答の意味表現 「ConfToUser, 3」
- E: 自然言語文 「3名様ですね。」

図 2: 音声対話システムを構成するモジュール

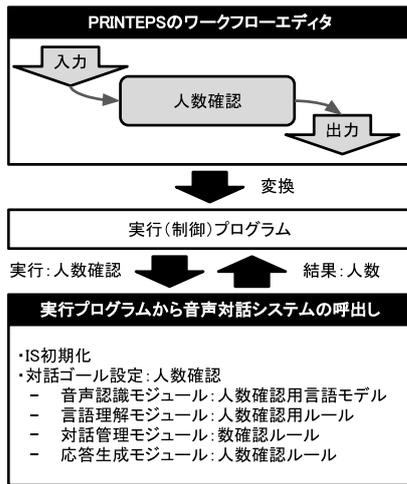


図 3: PRINTEPS と音声対話システムの関係

向であり、ロボットソフトウェア開発のデファクトスタンダードとなっている。PRINTEPS のような様々なセンサ・ロボット・ソフトウェアが介在する環境との相性は良い。

### 3. 音声対話システム

音声対話システムは、一般的に、音声認識（音声取得を含む）、言語理解、対話処理（対話管理）、応答文生成、音声合成の5つ要素から構成され、ユーザからの音声入力に対して、適切な応答を合成音声として出力する。そして、システムからの応答に対して、ユーザがまた音声入力をする。この繰り返しによって音声対話を行う。

本研究では、PRINTEPS のアーキテクチャに基づき、音声対話システムを構築するための各種モジュールを開発した(図2)。一般的な音声対話システムに必要な5つの要素に加えて、音声対話システムの内部状態の管理や、PRINTEPS から音声対話システムの制御・情報のやり取りを行うための要素である Information State を追加した。

図2全体を1つの音声対話システムと見た場合、この音声対話システムは、図1のように PRINTEPS ワークフローから呼び出すことができる。PRINTEPS ワークフローで指定された対話内容(対話から得たい情報:対話ゴール)は、実際に音声対話システムを実行する際には、対応する対話ルールを呼び出すために用いられる。図3のように、ワークフローで作成されたアプリケーションは実行プログラムに変換され、実行プログラムから音声対話システムを呼び出す際に、音声対話システム内の各モジュールが読み込む対話ルールを指定する。

```

<Word>
<From>いいえ</From>
<From>いいえ違います</From>
<From>違います</From>
<To>NO</To>
</Word>
<Word>
<From>$num 人です</From>
<From>$num 名です</From>
<From>いいえ$num 人です</From>
<From>いいえ$num です</From>
<To>Num, $num</To>
</Word>

```

図 4: 言語理解ルール

以下の節で、各モジュールの詳細を述べる。

#### 3.1 音声取得モジュール

一般的な音声対話システムでは、ユーザからの入力音声は、音声対話システム（音声認識）が動作しているコンピュータやロボットに搭載されているマイクを用いる。PRINTEPS の場合には、複数センサ、ロボットが存在しているため、その中からどのセンサを用いて音声を取得するかを選択する必要がある。

現在、音声取得モジュールでは、マイクが接続された特定のマシンか、ロボットからの音声取得が選択できる。PRINTEPS では、接客用ロボットとして Aldebaran 社にて開発 (SoftBank 社にて販売) されている Pepper\*2 を用いている。Pepper には4つのマイクからなるマイクロホンアレーが搭載されており、ここから音声を取得する。

取得された音声は、wav 形式で音声認識モジュールに送信される。

#### 3.2 音声認識モジュール

PRINTEPS では接客用ロボットとして Pepper を用いており、Pepper には予め音声認識機能が搭載されているが、Pepper の音声認識機能は言語モデルの変更などに柔軟に対応させることが難しいため、別途音声認識ソフトウェアに音声を入力して音声認識を行う。音声認識ソフトウェアには、オープンソース・ソフトウェアである Julius[河原 05] を用いている。

#### 3.3 言語理解モジュール

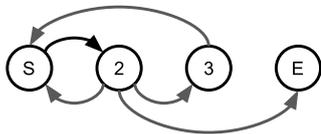
音声認識モジュールによって得られた入力発話文(テキスト)は、言語理解モジュールによって、システムで理解するための内部表現に置き換えられる。言語理解モジュールは、図4に示すようなルールに基づき入力発話文から内部表現への変換を行っており、<Word> タグ内で <From> タグで指定された文が入力された場合には、<To> タグで指定された内部表現に変換する(例えば、「いいえ違います」が入力された場合には、「NO」に変換される)。

入力発話文と <From> タグの値とのマッチングは、完全一致で行うのではなく、形態素に分割して主要な単語(名詞、接頭詞、動詞、感動詞、形容詞)のみを残して、コサイン類似度を計算し、最も類似度が高いものを採用する。これにより、ある程度の音声認識誤りや、ユーザの発話上の表現の揺れを吸収し、頑健に入力を受け付けることができる。

\*2 <http://www.softbank.jp/robot/>

システム：「いらっしゃいませ。3名様ですか？」  
 ユーザ：「いいえ、4名です。」  
 システム：「4名様ですね？」  
 ユーザ：「はい。」  
 システム：「ご案内いたします。少々お待ちください。」

図 5: 対話例：数確認



(a) 状態遷移ネットワーク

対話内容 : AskNum(数を確認)			
対話開始条件		: Num != NULL	
得られる効果		: KnowNum(確認された数字)	
状態	ユーザ入力	システム出力	遷移先
S	NULL	ConfToUser	2
2	YES	Answer	END
2	NO	AskToUser	3
2	Num	NULL	S
3	Num	NULL	S

(b) 各状態での処理内容

図 6: 対話処理ネットワーク (数確認対話)

### 3.4 対話処理モジュール

対話処理モジュールは、言語理解モジュールの出力を受けて、対話処理ルールに従って、システムの行動を決定するモジュールである。

対話システムは、文脈を考慮しない一問一答型のものと、文脈を考慮するものの2つに大別される。本対話処理モジュールでは、対話中のシステムの内部状態を InformationState に保持しており、この情報と、次のユーザ発話の理解結果とを合わせて、状態遷移ルールに基づき対話処理を行うことで、文脈を考慮した応答を生成することが可能である。

音声対話システムを構築する際、対話の内容(ドメイン)を決定する必要がある。本モジュールでは、図5の対話例に示すような小さな対話のやり取りを一つの単位として、このような対話ができるルールを対話の目的ごとに用意する。そして、PRINTEPS から音声対話システムを使用する際に、どの対話を行うかを選択することで、希望の内容の対話を実行する。

図6に、数を確認するルールの例を示す。ルールはネットワーク(有限状態オートマトン)の形で記述されており、状態Sから始まり、各状態でユーザ入力とのマッチングが行われ、マッチした場合には、対応するシステム応答を出力した後、遷移先の状態に遷移する。最終的に状態ENDに遷移すると対話は終了する。

1つの対話(例えば数確認)に対応する対話処理用ルールは、複数の状態(State)を持っており、対話の進行に応じて状態を遷移していく。各状態は複数の応答パターン(カテゴリ(Category))を持っており、ユーザの入力・システムの内部状態に応じて出力を行ったあと、状態を遷移させる。図7に対話処理ルールの一部を示す。例では、State: YesNo 内の一つのカテゴリが示されており、<Pattern> タグ内で示される入力条件(例の場合、ユーザ意図がYES)に入力・内部状態がマッ

```

<State ID="YesNo" Language="Japanese">
  <Category>
    <Pattern>
      <UttrUttd>
        <Intention>YES</Intention>
      </UttrUttd>
    </Pattern>
    <Generation>
      <Intention>Answer</Intention>
      <Function Type="bind"><Argument
        Value="$Num_conf"/><Argument Value="1"/>
      </Function>
    </Generation>
    <Transition State="END"/>
  </Category>
  
```

図 7: 対話処理用ルール

ユーザ：「4名です。」  
 ↓  
 言語理解結果：Num, 4  
 ↓  
 対話処理結果(出力)：ConfToUser, 4  
 ↓  
 応答生成結果：4名様ですか?  
 ↓  
 システム：「4名様ですか？」

図 8: 「数確認」ルールを用いて「人数確認」を行う例

チした場合に、<Generation> タグで示された出力(システム出力: Answer, 内部変数\$Num\_confに1を代入)が実行され、<Transition> タグで指定された状態(例の場合、END)に遷移する。

PRINTEPSにてアプリケーションを開発する開発者としては、対話の専門知識がない一般ユーザ(もしくは、一般エンジニア)を想定しており、上述の例のような対話処理用ルールは、独自に作成することは困難である。そのため、このような対話処理用ルールは予め PRINTEPS 内に複数用意しておき、それを必要に応じて選択して使用できるようにしている。用意する対話のルールは、より一般的に記述しておくことで汎用性を高める。例えば、図5の対話例は喫茶店への入店人数を確認する対話であるが、これを PRINTEPS で実現する場合には、対話処理ルールには「数確認」を行うルールを用い、言語理解モジュール・応答生成モジュールにおいて、意味表現と自然言語文との変換を、人数確認対話に対応したものにする(図8)。図中の内部表現 Num は「数」を表し、ConfToUser は「ユーザへの確認」を表す。

なお、内部で保持する変数の形式、変数の処理(比較、代入など)、それを実行するための対話ルールの記述方式などは、黄[Huang 08]による音声対話システム GECA を参考にした。

### 3.5 Information State モジュール

Information State (IS) では、音声対話システム全体(主に対話管理)の情報の管理(保存)を行う。システムの全体図(図2)では、ISとの情報のやり取りを表す矢印は、明示的には対話管理としか接続していない。しかし、ISとの接続

```

<Behavior>
  <Generation>
    <Intention>ConfToUser</Intention>
    <Arg num="1">$num</Arg>
  </Generation>
  <OutputText>$num 名様ですか?</OutputText>
</Behavior>
<Behavior>
  <Generation>
    <Intention>AskToUser</Intention>
  </Generation>
  <OutputText>何名様ですか?</OutputText>
</Behavior>

```

図 9: 応答生成用ルール

は ROS ネットワークをベースとして、すべてのモジュール間で可能であり、音声対話システム起動時には IS が初期化され、音声認識から音声合成までの各モジュールでどのルールを用いるかの指定や、音声対話システムの終了判定時にも、IS が参照される。

音声対話を通じてユーザから得られる情報は、IS に格納されるため、音声対話システム外のモジュールが IS を通じてこれらの情報を得ることも可能にしてある。また、各種センサや知的処理部など、PRINTEPS 内（音声対話システム外）で得られた情報を、IS に格納しておくことで、これらの情報を対話処理に用いることが可能になる。

### 3.6 応答生成モジュール

応答生成モジュールは、対話処理モジュールから出力されたシステム行動の内部表現を、自然言語文に変換するモジュールである。図 9 に、応答生成用ルールの例を示す。応答生成モジュールは、対話処理モジュールからの出力を受け取り、<Generation> タグ内で指定されている条件とのマッチングを行い、<OutputText> タグで指定された自然言語文を出力する。例えば、意図タグとして「ConfToUser」が、引数「3」と共に入力された場合には、「3名様ですか?」という文が出力される。

### 3.7 音声合成モジュール

音声合成モジュールは、応答生成モジュールから出力された自然言語文を受けて、音声合成を行う。合成された音声は、スピーカから出力される。PRINTEPS で接客用ロボットとして用いている Pepper には、音声合成機能が搭載されている為、Pepper から音声合成出力をする場合には、Pepper 内の音声合成機能を用いる。しかし、その他のロボットを用いる環境などでは、別途音声合成ソフトウェアを用意する必要がある。音声合成ソフトウェアには、オープンソース・ソフトウェアである Open JTalk<sup>\*3</sup> を用いている。

## 4. 音声対話システム導入の重要性

ここでは、本音声対話システムを PRINTEPS に導入する重要性を述べる。PRINTEPS にて構築された実践知能アプリケーションにおいて、システムがユーザと対話的に情報を授受する必要がある場面はいくつも考えられる。実践例として喫茶店におけるアプリケーションを考えても、客の入店の際の人数確認、混雑状況の説明、座席予約、案内する座席の確認・要望

受付、注文受け付けなど多数の場面が考えられる。これらの場面においてユーザと情報をやりとりするアプリケーションを構築する際に、音声対話システムがない場合には、想定される受け答えや対話の流れ、様々な入力文型への対応、音声認識を失敗する場合などをすべて考慮して手でワークフローを書き下す必要があるが、これには対話実装に関する専門的な知識が必要である。また記述できたとしても、記述量が膨大になり現実的ではない。音声対話システムを導入すれば、言語理解によって比較的自由的な文型の入力に対応でき、対話処理によって音声認識や言語理解が失敗した場合に聞き返す処理も可能になる。注文を受けるような場面では、内容の順番に関係なく入力を受け付け、注文内容に付随する事項（サイズやホット・アイスなど）の確認も行うことができる。

## 5. おわりに

本稿では、PRINTEPS アーキテクチャにて動作する、音声対話システムの開発について述べた。音声対話システムに必要な 5 つのモジュール（音声認識、言語理解、対話管理、応答生成、音声合成）を構築し、これらを組み合わせ、音声対話システムを構成している。PRINTEPS からは、対話から得たい情報（対話したい内容：対話ゴール）を指定して音声対話システムを呼び出すことで、音声対話システムはユーザと音声対話をし、必要な情報を取得することができる。各対話ゴールに対応する細かい対話処理ルールなどは、予め用意し、PRINTEPS による実践知能アプリケーション開発者は、音声対話に関する知識なしに、音声対話システムを構築・利用することが可能である。

今後は、対応できる対話（対話ゴール）を増やし、PRINTEPS から簡単により広い対話の構築を可能にしていきたい。また、対話中のロボットの動作、身振り手振りを制御する仕組みを導入したい。現在は対話ルール自体の作成は専門家が行う必要があるが、対話ルールを簡単に構築できるオーサリングツールの開発を行いたい。

## 謝辞

本研究は、科学技術振興機構 (JST) 戦略的想像研究推進事業 (CREST) 「実践知能アプリケーション構築フレームワーク PRINTEPS の開発と社会実践」の支援によって実施した。

## 参考文献

- [Huang 08] Huang, H.-H., Nishida, T., Cerekovic, A., Pandzic, I. S., and Nakano, Y.: The Design of a Generic Framework for Integrating ECA Components, *Proceeding of 7th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2008)*, pp. 128–135 (2008)
- [河原 05] 河原 達也, 李 晃伸: 連続音声認識ソフトウェア Julius, 人工知能学会誌, Vol. 20, No. 1, pp. 41–49 (2005)
- [山口 15] 山口 高平, 中野 有紀子, 斎藤 英雄, 森田 武史, 青木 義満, 萩原 将文, 斎藤 俊太: 知能共進化のための実践知能アプリケーションプラットフォーム PRINTEPS, 第 29 回人工知能学会全国大会, 114-2 (2015)
- [森田 16] 森田 武史, 西村 良太, 山口 高平: ROS に基づく総合知能アプリケーション開発プラットフォーム PRINTEPS のアーキテクチャ, 第 30 回人工知能学会全国大会, 4C4-1 (2016)

\*3 <http://open-jtalk.sourceforge.net/>