

# 対話型車両エージェント

## Voice Agent for Communicating with Automobiles

内田 渉 辻野孝輔 角野公亮

Wataru Uchida, Kosuke Tsujino, and Kosuke Kadono

NTT ドコモ サービスイノベーション部

Service Innovation Department, NTT DOCOMO, Inc.

**Abstract:** NTT DOCOMO developed a voice agent system which is intended to be used in automobiles. The system monitors vehicle's information such as brake operations, and spontaneously begins voice conversation when it detects the events such as "sudden brake". Moreover, users can begin talk with the system without tapping smartphone or pressing button on steering (Hands-free voice recognition). This paper describes the system's architecture, and process to generate system's responses.

### 1. まえがき

近年「しゃべってコンシェル」[1]などの音声エージェントシステムが普及しつつある。音声エージェントシステムはスマートフォンの基本機能や各種情報サービスへの簡便なアクセス手段として登場したが、車両や家電など様々な機器への応用やシステムとのコミュニケーション中心の使い方がされるなど、多様な領域への適用が期待されている。

本稿では、車両内での利用を想定した音声対話が可能なシステムである対話型車両エージェントについて述べる。本稿では「対話」とはユーザとシステムの音声発話を1回以上続けてやりとりする一連の動作を表すものとする。

人と話す場合と変わらないほど自然な双方向のインタラクションの実現を目指し、対話型車両エージェントを下記の特徴を備えるシステムとした。

#### ● 特徴 1: ハンズフリーの音声認識

これまでの音声エージェントシステムでは、ユーザが音声認識を開始する時に画面タップや車両のステアリングに設置したリモコンを操作するなどの予備動作が必要だった。何気ない会話を発話しやすくする効果や複数回の発話のやりとりを実施しやすくする効果を狙い、予備動作が不要でユーザが常に発話開始可能な機能を実装した。

#### ● 特徴 2: システム主導の対話

これまでの音声エージェントシステムでは、ユーザが先に発話することで、初めて対話が始まる。ユーザに発話の意志がなくても、注意を促すべき場面・提案をすべき場面など音声対話が有効な場面が存在すると考えた。そこで、運転状況などのコンテ

キストにもとづきシステムから話しかけ、対話を始める機能（システム主導の対話）を実装した。

### 2. システム概要

#### 2-1. システム構成と処理の流れ

システム構成を図1に示す。ほぼ全ての処理をネットワーク上に設置した音声認識・対話・音声合成の3サーバで実施する。Androidスマートフォン向けのクライアントアプリに各サーバとの連携処理と、車両への配置を意識したマイク・スピーカによる音声中心のユーザインタフェースを実装した。車両等のコンテキスト生成元がシステム主導の対話を開始する契機となるイベント情報を生成する。

処理の流れを説明する。クライアントアプリは画面タップやリモコン操作の代わりに車両内の音声入力状況を常に監視し、ユーザからの音声発話が行われた区間を検出する。音声発話を検出すると、その

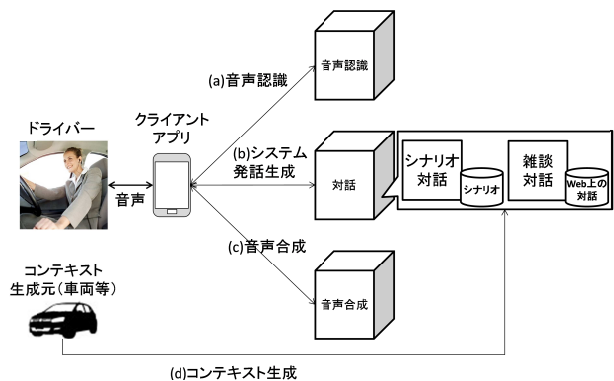


図1: システム構成

区間の音声情報をもとに発話テキストへ変換する音声認識処理を行う (図 1(a))。通常はユーザの音声発話をきっかけに、音声認識 (図 1(a))、システム発話生成 (図 1(b))、音声合成 (図 1(c)) の 3 つの処理を順に繰り返すことで対話が行われる。

コンテキスト生成元がイベント情報を伝えることで、システムの発話から対話を始めることも可能である。車両内での通信に使用される CAN (Controller Area Network)[2]等へのアクセスによって得られる車両状態等の情報を解析することで、ユーザの周囲で発生したイベントに関する情報を得る。この場合コンテキスト生成 (図 1(d))、システム発話生成 (図 1(b))、音声合成 (図 1(c)) の流れで対話を開始する。その後はユーザの発話から対話が始まる場合と同じく図 1(a)-(c)の 3 処理を繰り返し、対話を継続する。

## 2-2. システム発話生成

対話型車両エージェントでは対話の自然さ・きめ細やかさ等の品質を向上するためライター等の人手によるシナリオを組み入れる仕組みを導入した。シナリオに基づく場合、想定した対話内容に沿う限りは高品質な対話が行える一方、ユーザの広範な発話に対応することは難しい。そこで次の手段で平均的な品質向上を図った。

- 同じ意図によるユーザ入力をまとめてシナリオに記述可能な正規化処理を実装した。
- 対話内容が想定しやすいシステム主導の対話時 (急ブレーキの発生等) を中心にシナリオを作成し、それ以外は雑談対話機能[3]で発話を生成する。雑談対話機能は発話の意図や文脈を考慮しあらかじめ収集したインターネット上のテキスト等から生成した大量のデータをもとに発話を生成する。人手での記述に比べ個々の回答の品質は劣るものの幅広いユーザ発話に返答できる。システム発話生成処理の流れを図 2 に示す。ユー

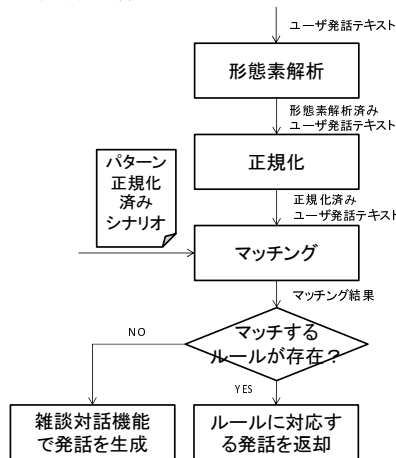


図 2: システム発話生成処理の流れ

ザ発話のテキストを入力とし、まず形態素への分割ならびに正規化を行う。例えば「今の天気教えてください」というテキストに対し「[<現在>|<時間表現:日時:今>][<天気>]」という各単語に対応する正規化候補と、文末の「教えてください」という表現に対し「疑問文である」という意味のラベルを得る。

次に予め記述されたシナリオとのマッチング処理を行い、マッチするルールが存在する場合はそれに対応した発話を出力する。シナリオ記述形式はソフトウェアエージェント記述言語 AIML (Artificial Intelligence Markup Language)[4] を参考に設計した。AIML の処理系には Program D[5]等が既に存在するが、表層文字列に対するマッチングのみが実装されているため、新たに正規化されたユーザ発話とシナリオのマッチングを含む処理系を実装した。

ここで、ユーザ発話パターンに加え前回のシステム発話やコンテキスト生成元から通知されたイベントの種別で条件づけを行うことも可能である。前回システム発話での条件付けにより、システムからの質問に対するユーザの回答へさらに返答する動作が可能になる。通知されたイベント種別による条件づけにより、イベント発生時の初回発話を指定できる。

シナリオ中にマッチするパターンが存在しない場合は雑談対話機能を利用し発話生成を試みる。

## 3. まとめ

車両の運転情報等から得られるイベントにもとづきコンテキストに応じた対話がハンズフリーで行える対話型車両エージェントについて述べた。今後はより人間に近い、自然な対話技術の実現ならびに家電機器やロボット等の様々なデバイスへの応用へ取り組み予定である。

## 参考文献

- [1] しゃべってコンシェル | サービス・機能 | NTT ドコモ, URL: [https://www.nttdocomo.co.jp/service/information/shabette\\_concier/](https://www.nttdocomo.co.jp/service/information/shabette_concier/)
- [2] ISO 11898-1:2003 - Road vehicles -- Controller area network (CAN) -- Part 1: Data link layer and physical signaling, URL: [http://www.iso.org/iso/home/store/catalogue\\_tc/catalogue\\_detail.htm?csnumber=33422](http://www.iso.org/iso/home/store/catalogue_tc/catalogue_detail.htm?csnumber=33422)
- [3] 大西 可奈子, 吉村健: コンピュータとの自然な会話を実現する雑談対話技術, NTT DOCOMO テクニカルジャーナル, Vol. 21, No. 4, pp. 17-21, (2013)
- [4] AIML - The Artificial Intelligence Markup Language, URL: <http://www.alicebot.org/aiml.html>
- [5] programd\_aiml - ainotebook, URL: [https://code.google.com/p/ainotebook/wiki/programd\\_aiml](https://code.google.com/p/ainotebook/wiki/programd_aiml)