

複数のプログラミングパラダイムに基づく音声対話システム記述言語および開発環境の研究

Studies on spoken dialog system description language and development environment based on multiple programming paradigms



荒木 雅弘 (Masahiro ARAKI, Ph. D.)

京都工芸繊維大学 工学科学研究科 情報工学専攻
准教授

(Associate Professor, Kyoto Institute of Technology, Graduate School of Information Science)

情報処理学会 人工知能学会 言語処理学会 日本音響学会

受賞:人工知能学会 研究奨励賞 (2000) 第42回情報処理学会全国大会 研究奨励賞 (1992)

著書:イラストで学ぶ音声認識 講談社 (2015) フリーソフトではじめる機械学習入門 森北出版 (2014) フリーソフトで学ぶセマンティック Web とインタラクション 森北出版 (2010) フリーソフトでつくる音声認識システム・パターン認識・機械学習の初歩から対話システムまで・森北出版 (2007) 音声対話システム (共著) オーム社 (2006) Multilingual and Multimodal Dialogue Systems: Development and Assessment (共著) Wiley (2005)

研究専門分野: 音声対話処理

あらまし 現在、音声対話システムの開発は対話の状態遷移に基づく記述が主流であり、システムが高度化・複雑化するにつれて、拡張性・保守性の低さが問題になりつつある。本研究は、オブジェクト指向・論理型言語・関数型言語などのプログラミングパラダイムから音声対話システムの記述に適した要素を抽出し、新たな音声対話システム記述言語をすることを目的とし、オブジェクト指向に基づくデータ記述からの状態遷移記述の自動生成・論理的記述による対話コンテンツの自動獲得を実現した。また、言語理解結果の関数による記述について実装方式を提案した。

1. 研究の目的

音声対話システムは、ユーザとの対話を通じて特定のタスクを行うことができるものとして、また、雑談を通じてユーザに娯楽や癒やしを提供するものとして研究・開発が行われている。現在の音声対話システムの開発手法は、状態遷移記述で対話管理を行い、各状態で音声認識・音声合成・バックエンドシステムを制御するコードを記述する方式が主流であるが、実用的な音声対話システムでは状態数が膨大になり、拡張性・保守性の低さが問題となっている。本研究では、オブジェクト指向・論理型言語・関数型言語などの拡張性・保守性に優れたプログラミングパラダイムに基づいた音声対話システム記述言語を設計し、その処理系を実現することを目的とした。

2. 研究の背景

近年、音声対話システムは携帯情報端末やロボットのインタフェースとして注目されており、高齢者や視覚障害者が高度な情報通信機器を使いこなすための手段としても期待が高まっている。現在の音声対話システムの開発手法は、状態遷移記述で対話管理を行い、各状態で音声認識・音声合成・バックエンドシステムを制御するコードを記述する方式が主流である。この方式では、実用的なシステムでは状態数は数千にも及ぶものになり、その保守管理に多大なコストがかかること、また、そのようにコストの高い対話管理であっても、音声対話システム開発全体工数の20%以下でしかないことが指摘され、開発方法論のパラダイムシフトが求められている。

筆者らは、これまで近年のソフトウェア工学の知見を活用した、開発効率が高く、拡張性・保守性に優れた音声対話システム開発方法論の研究を進めてきた。具体的には、セマンティック web 標準技術に基づいて収集された実データである CSRs *1 をクラスライブラリとし、開発者はクラスライブラリに基づいたデータ定義を行うだけで、近年の Web アプリケーションの標準である MVC モデル*2 のコードが自動生成される、オブジェクト指向型の音声対話システム開発方法論を提案・実装した。

本研究は、この研究成果を発展させ、オブジェクト

複数のプログラミングパラダイムに基づく音声対話システム記述言語および開発環境の研究

Studies on spoken dialog system description language and development environment based on multiple programming paradigms

指向設計に基づく MVC コードの自動生成によって音声対話システム開発全体をカバーするという基本スタンスは維持したまま、記述論理で定義された推論規則を開発言語に組み込み、対話コンテンツの自動獲得を実現した。また、言語理解結果を関数で表現し、情報状態を入力として更新を行うプロセスの研究を行った。

3. 研究の方法、研究の結果

3.1 宣言的記述からの対話状態遷移の自動生成

ここでは、筆者らの研究グループが開発したオブジェクト指向設計に基づくマルチモーダル対話システム (MMI : Multi-Modal Interaction) 記述言語 MrailsScript [1] について説明する。

MrailsScript は、2章で述べた既存の MMI システム開発手法の問題点から、MVC の明確な分離、GUI ベースのアプリケーション開発方法論との共通性の確保、モダリティの追加・変更等に対する保守性・機能拡張性の維持の3点を重視した設計を行っている。

MrailsScript は、オブジェクト指向設計と MVC モデルに基づいた Web アプリケーションプラットフォームである Grails フレームワーク^{*3}を基礎とすることで、開発効率や保守性・機能拡張性を保っている。Grails の MVC モデルは、アプリケーションが扱うコンテンツを操作するドメインモデル (Model)、ユーザとのやり取りとコンテンツの表示方法を規定するビュー (View)、タスクを遂行するために Model 部・View 部とデータの入出力を行うコントローラ (Controller) の3つにコードを分離する。

MrailsScript は、Grails のドメインモデル定義を拡張した記述を MMI システム開発の中心的記述と位置づけた。ドメインモデル定義にセマンティック web の標準オントロジーのクラス定義を継承させることで、ドメインモデルが扱うデータのオントロジーを決定し、これにアノテーションによってモダリティ非依存の対話タスクと対話主導権を付加することで、対話システムの設計を行う。

図1は、Schema.org^{*4}の Book クラスを継承し、データベース検索タスクをシステム主導で行うためのアノテーションが付加された MrailsScript の例である。Book クラスでプロパティ名と型が定義されているた

め、定義済みのプロパティをデータモデルに採用する際は、その型を明示する必要はなく、constraints の要素として列挙するだけでよい。

MrailsScript では、タスク・主導権の分類 [2] に基づき対話を分類し、ドメインごとにアノテーションによって指定する方法を取った。対話の主導権は、達成すべき目標や想定されるユーザによって、システム主導・ユーザ主導・混合主導の3つに分類される。また、対話タスクは、システムとユーザの間を流れる情報の方向という観点から、情報検索を行う検索タスク・ユーザに情報の入力を求めるスロットフィリングタスク・ユーザに情報を提供する説明タスクの3つに分類される。MrailsScript はこれらのタスク・主導権の組合せ毎に異なる MVC コードを生成することで、それぞれの対話を実現するモダリティ非依存な MMI アプリケーションの雛形を自動生成する。

```
@initiative.System
@task.DBSearch
class MyBook extends Book {
    static hasMany = [comment:MyUserComments]
    static constraints = {
        name( onsearch:"ilike" )
        description( onsearch:"ilike" )
        author.name()
    }
}
```

図1 Schema.org の Book クラスを継承した MrailsScript

3.2 開発支援環境の構築と論理記述による対話コンテンツの自動獲得

MrailsScript は、状態遷移記述に基づく既存対話記述言語と比較して、保守管理性および拡張性に優れている。データモデル定義は、セマンティック web のオントロジーに基づいて行うので、比較的単純な操作であり、Linked Open Data^{*5}のメタデータをそのまま用いることもできる。しかし、MrailsScript の利点を活かしたデータ定義作業を行うためには、MMI システム開発者がセマンティック web 技術に精通する必要があった。この問題を解決するため、筆者らは、オントロジーの知識に基づいた開発支援を行う MMI シ

複数のプログラミングパラダイムに基づく音声対話システム記述言語および開発環境の研究

Studies on spoken dialog system description language and development environment based on multiple programming paradigms

システム開発環境 MrailsBuilder を構築した。

MrailsBuilder は、図 2 に示すように、オントロジーをリアルタイムで探索し、意味情報を可視化することで、データ定義作業を支援する。また図 3 に示すように、定義されたデータモデルから Linked Open Data を自動的に取得する SPARQL を自動的に作成し、さらに、別途定義された推論規則と組み合わせることによって、対話システムで提供するコンテンツを幅広い情報源から自動取得する機能を持つ。この開発環境を用いて、特定タスクの MMI システムを作成し、開発プロセスにおけるこれらの機能の有効性を確認した [3]。

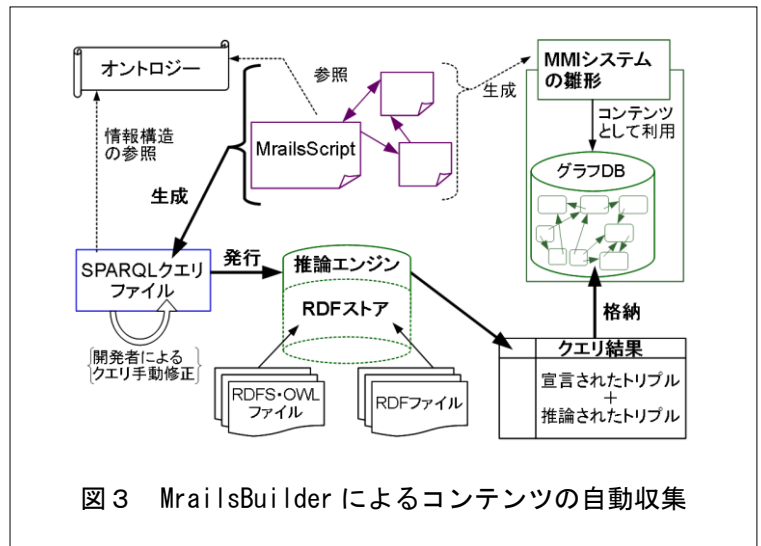


図 3 MrailsBuilder によるコンテンツの自動収集

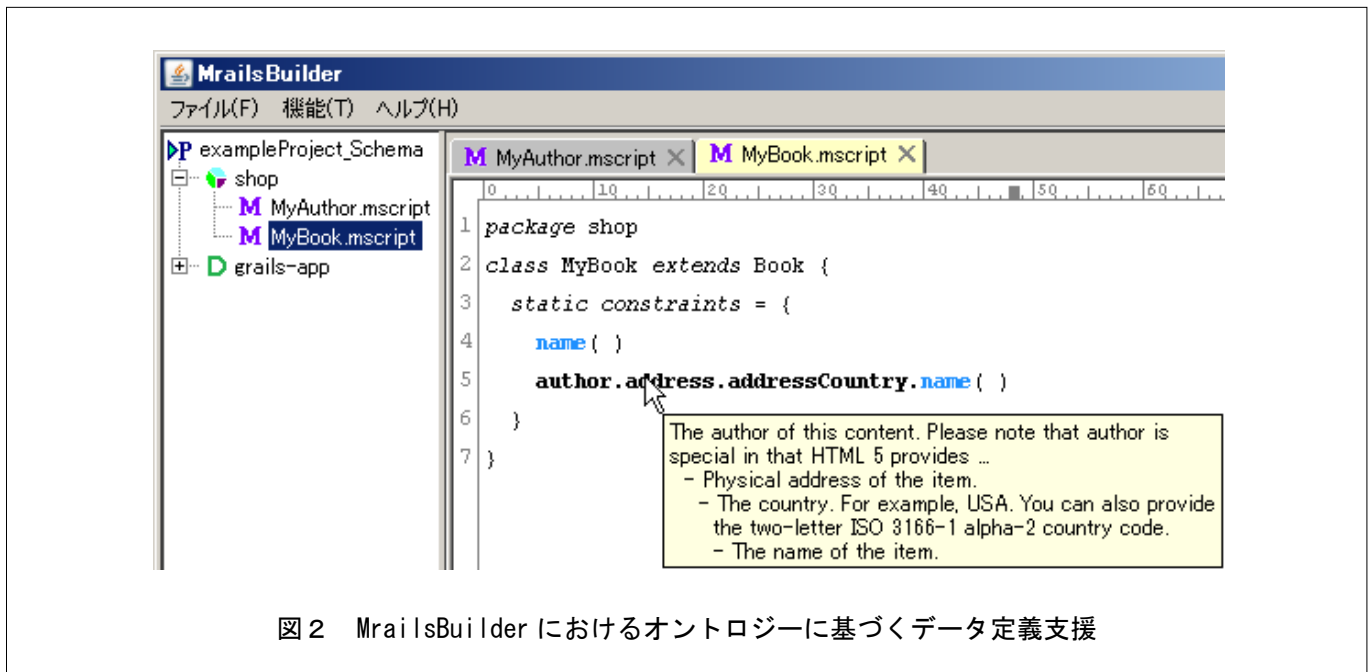


図 2 MrailsBuilder におけるオントロジーに基づくデータ定義支援

3.3 言語理解結果の関数による表現

音声対話システムやマルチモーダル対話システムにおいては、断片的な入力の連続や、複数のモダリティを組み合わせた入力を解釈する必要があるが、これらを個々の入力として扱って状態遷移記述を行うと、膨大な組合せに対する記述を行う必要がある。そこで、入力理解結果を関数として表現し、対話状態の遷移記述を関数の連続適用へと置き換え、対話制御の記述をなくすことでこの組合せ爆発の問題を解決する手法を提案した [4]。対話システムに出現する様々な発話行

為タイプ毎に関数の雛形を用意し、入力解釈によって得られた結果を用いてその雛形を書き換えることで、発話理解結果としての関数を得る。音声入力の場合は、誤認識による不確実性に対処するために複数認識結果に対する理解結果をリストとして扱う。対話状態を管理する情報状態更新モデルにこの関数を適用することで、対話を進めることができる。

上記の手法を関数型言語 Scala のドメイン依存言語として実装し、列車チケット購入を行うスロットフィリング型対話システムを実現した。スロットフィリン

複数のプログラミングパラダイムに基づく音声対話システム記述言語および開発環境の研究

Studies on spoken dialog system description language and development environment based on multiple programming paradigms

グ型タスクでは、情報状態更新モデルに持たせる情報を以下の4つに限定して実装することができた。

- ◇PLAN：長期的な対話の目標となる行動のリスト
- ◇TMP：対話参加者間で明示的に対話によって確立された情報を反映したフィールド
- ◇BEL：システムが対話によって確認した命題の集合
- ◇QUD：議論中の質問のスタック

4. 将来の展望

4.1 音声対話システム開発者に対する波及効果

MrailsScriptは、今後データの充実が見込まれるセマンティック web リソースを背景知識としているので、最新・大量のデータを対象とした音声対話システムを用いたサービスをすぐに始めることができる。また、システムの上位レベル設計には、近年の web アプリケーション開発フレームワークである Rails の考えを用いているので、web アプリケーション開発者がこれまで敷居の高かった音声対話システム開発に簡単に取り組むことができる。

4.2 情報通信端末ユーザに対する波及効果

現在のスマートフォン等の携帯情報端末でも音声操作機能が搭載されることが多くなっているが、それらは1つの音声コマンドで特定できるアプリ操作や web 検索の音声入力インタフェースとして一問一答型の利用を前提とされており、複数ターンに渡る複雑な対話は想定されていない。研究レベルの音声対話システムにおいても、タスクはバスの乗換・時刻表案内や観光案内等、比較的単純なものに限定されている。本研究の成果を発展させて開発の初期コストが低くなると、知的な推論を伴うような様々な音声対話サービスがユーザに提供されることが期待される。

4.3 今後の展開

セマンティック web リソースは、多言語で提供されているものであり、本開発手法そのものもデータモデル定義を中心とするものなので、個別の言語には依存していない。この特性を利用して、観光案内等の多言語サービスを提供するシステムへの発展が考えられる。

用語解説

- *1 CSRs (Collaboratively Constructed Semantic Resources) : 一般ユーザによって作成されたセマンティック web 標準に基づく意味構造が付与されたデータ
- *2 MVC モデル (Model-View-Controller model) : web アプリケーション開発の枠組みで、データ処理ロジックである Model と、ユーザインタフェースである View と、制御を行う Controller を明確に分離する方法
- *3 Grails フレームワーク : スクリプト言語 Groovy を使用した MVC フレームワーク
- *4 Schema.org : 意味マークアップのための共通オントロジーを提供している web サイト
- *5 Linked Open Data : 再利用を許すオープンライセンスに基づき、Web 上に公開されるデータ

参考文献

- [1] 竹腰大輔, 荒木雅弘: データモデル定義に基づくマルチモーダル対話記述言語の設計と実装, 人工知能学会論文誌, Vol.28, No.3, pp.261-266, doi:10.1527/tjsai.28.261, 2013.
- [2] M. Araki, K. Komatani, T. Hirata and S. Doshita: A Dialogue Library for Task-oriented Spoken Dialogue Systems, Proc. IJCAI Workshop on Knowledge and Reasoning in Practical Dialogue Systems, pp.1-7, 1999.
- [3] 竹腰 大輔, 荒木 雅弘: コンテンツの情報構造を利用したマルチモーダル対話システム開発支援環境, 人工知能学会論文誌, Vol.30, No.1, pp.61-70, doi: 10.1527/tjsai.30.61, 2015.
- [4] 井手麻友美, 荒木雅弘: マルチモーダル対話記述における関数による解釈表現の検討, 2013 年度人工知能学会全国大会論文集, 1K3-OS-17a-3, 2013.

この研究は、平成24年度SCAT研究助成の対象として採用され、平成25～27年度に実施されたものです。