

LF-3 XML スキーマ言語からの音声対話システムの自動生成 Automatic Generation Method of Spoken Dialog Systems from XML Scheme Language

荒木雅弘
Masahiro Araki

1. はじめに

現在、世界中の多種多様な情報が World Wide Web 上で利用可能であり、多くの人々は GUI (Graphical User Interface) ベースの Web ブラウザを用いて、これらの情報にアクセスしている。これらの Web 上の情報に対して、音声対話を用いてアクセスできるようになると、いつでもどこでも手軽に、欲しい情報に直接アクセスができるようになる。

Web コンテンツからの音声対話システムの生成という問題に対して、我々は対話ライブラリを用いた手法を提案した[1]。本稿では、その手法を洗練し、XML スキーマ言語 Relax で記述された XML スキーマを音声対話システムに変換する手法を提案する。

既存手法に対して、提案手法は以下のような利点がある。

- (1) サーバサイドプログラムを black-box 化する必要がない
- (2) コンテンツ記述に対する制限が少ない
- (3) 生成されたシステムに対する修正が容易

2. 従来手法の問題点

現在主流となっている MVC(Model View Controller)アーキテクチャに基づく Web システムにおいて音声入出力を用いる場合、サプレットからの出力を HTML から VoiceXML[2]に置き換える手法が考えられる(図 1)。

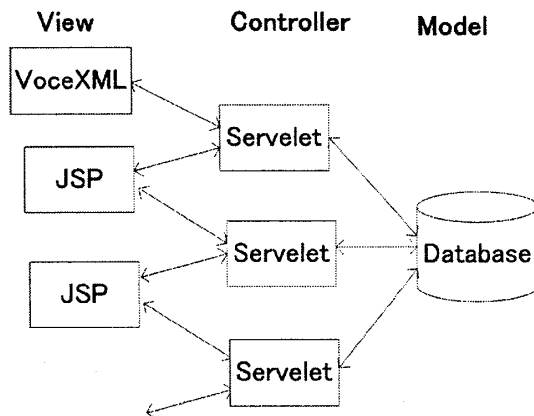


図1 MVCアーキテクチャ

我々が提案した従来手法[1]では、データベースのフィールド情報を XML で記述し、そのタグ名などの情報からユーザ入力文法を推定して初期入力対話を構成する。また、検索レコードを絞り込む対話は、データベースフィールドの重要度に関するヒューリスティクスを埋め込んだサプレットを用いて生成する。そのため、データベース構造

京都工芸繊維大学, Kyoto Institute of Technology

や XML の記述に多くの制約を必要とした。また、自動生成された対話パターンを変更する場合は、直接サプレットを書き換えなければならないため、保守性に問題があった。

3. XML スキーマ言語からの対話パターン生成

3.1 提案手法の概要

上記の問題点を解決するために、我々是对話パターンである VoiceXML を生成するのに用いる情報を XML によるデータ記述から、XML スキーマに変更した。XML スキーマとしては、記述能力が高く、Java 言語へのスキーマコンパイラ行なうソフトウェア(Relaxer[3])が公開されている Relax[4]を用いた。

Relax で記述されたスキーマを入力とすることによって、以下のような利点がある。

- (1) タグ名だけでなくデータタイプを参照することによって、より高精度なユーザ文法の推定が可能になる。
- (2) スキーマコンパイラと Tag ライブラリを用いることで、対話の流れを JSP(JavaServerPages)のみで記述できるので、修正が容易になるばかりでなく、保守性も高まる。

XML スキーマを用いて生成した対話システムの処理の流れを図2に示す。

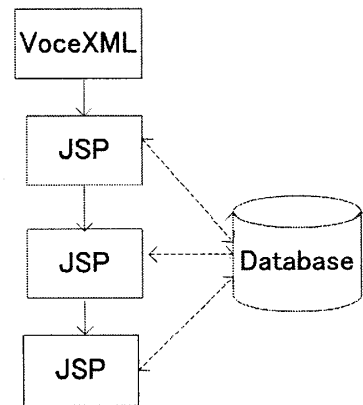


図2 提案アーキテクチャ

3.2 Relax による XML スキーマの記述

ここではデータベース検索タスクにおける対話システムの生成について説明する。データベースの各フィールドを XML の属性にマッピングしたスキーマを Relax で記述すると、図3に示すようになる。ここでは、例としてホテル情報DBを取り上げた。

```

<elementRule label="ホテル情報">
  <tag/>
  <sequence>
    <element name="ホテル名" type="string" occurs="1"/>
    <element name="地域" type="string" />
    <element name="料金" type="int" occurs="1"/>
  </sequence>
</elementRule>

```

図3 Relaxによるスキーマの記述(一部)

3.3 VoiceXMLによる初期入力対話の記述

データベース検索での初期入力対話は、一発話で複数の変数値を埋めることができる VoiceXML の混合主導型を用いる(図4)。まず、Relax で記述されたスキーマのタグ名の情報(element 要素の name 属性)や type 属性を利用してユーザ発話文法(main.gram)を推定する。VoiceXML1.0 仕様上の組み込み文法名や、処理系に組み込まれた文法名とタグ名が一致する場合は、そのまま対応する文法を用い、一致する文法がない場合は、タグ名を形態素解析し、部分一致する文法で代用する。また、複数の条件を入力する文に対応するために、条件組み合わせのための発話テンプレートを利用して文法を拡張する。

また、element 要素の occurs 属性から、データベース検索における必須条件であるかどうかを判断し、必須条件でないものは、VoiceXML で対応する field 要素作成時に、省略可能を意味する cond 属性(値は false)を付加する。図4の例では、地域名が発話に含まれていなければ、システムが主導権を取り、「地域を入力して下さい」という問いかけを行なうことになる。

```

<?xml version="1.0" encoding="Shift_JIS" ?>
<vxml version="1.0">
  <form>
    <grammar src="main.gram" />
    <block> こちらはホテル情報検索サイトです</block>
    <initial name="start">
      <prompt>検索条件を入力して下さい</prompt>
      <help> ホテル名、地域、料金を入力して下さい</help>
    </initial>
    <field name="ホテル名" cond="false"/>
    <field name="地域">
      <prompt> 地域を入力して下さい</prompt>
    </field>
    <field name="料金" cond="false"/>
  </block>
  <submit next="step1.jsp" />
</form>
</vxml>

```

図4 VoiceXMLによる初期入力対話

3.4 JSPによる対話記述

初期入力対話によって検索条件が定まると、候補レコードを絞り込む対話に遷移する。絞り込み対話は JSP で記述

され、データベース検索を行なう部分と、検索結果に従って絞り込みまたは結果報告の対話を生成する部分からなる。

これらの機能はライブラリで実現される。データベースアクセスや XSLT を用いたオブジェクトから VoiceXML への変換は Relaxer が提供する Tag ライブラリを使用し、候補レコード数に応じて絞り込み対話を生成する部分や、報告対話を生成する部分は自作の Tag ライブラリを用いる。

```

<%@ page language="java"
  contentType="text/xml; charset=Shift_JIS" %>
<%@ taglib prefix="relaxer"
  uri="http://www.relax.org/taglib"%>
<%@ taglib prefix="vox"
  uri="http://vox.dj.kit.ac.jp/taglib"%>

<?xml version="1.0" encoding="Shift_JIS"?>
  <vxml version="1.0" >
    <form>
      <データベースアクセス>
      <絞り込み or 報告の判断>
      <XSLTによるVoiceXMLの生成>
    </form>
  </vxml>

```

図5 JSPによる対話記述(擬似コード)

絞り込みを行なう場合は、決定木における分岐質問生成と同じアルゴリズムを用い、得られる情報量の期待値が最も大きいフィールドに関する質問を行なう。

また、候補レコードが閾値以下(現在は3レコードに設定)に絞り込めた場合は、報告対話を生成する。

4. まとめと今後の課題

本稿では XML スキーマ記述からデータベース検索を行なう音声対話システムを自動生成する手法を提案した。今後は、[1]で分類した他のタイプのタスク(スロットフィリング、説明)に関する本手法の適用を検討する。

謝辞

本研究は新エネルギー・産業技術総合開発機構(NEDO)平成14年度産業技術研究助成(00A18004b)を受けて行われた。

参考文献

- [1] Masahiro ARAKI, Tasuku ONO, Kiyoshi UEDA, Takuya NISHIMOTO, and Yasuhisa NIIMI: "An Automatic Dialogue System Generator from the Internet Information Contents", Proc. EUROSPEECH2001, pp1150-1156, 2001.
- [2] VoiceXML Forum: Voice eXtensible Markup Language (VoiceXML™) version 1.0, <http://www.w3.org/TR/voicexml/> 2000.
- [3] 浅海智晴: Relaxer, ピアソンエデュケーション, 2001.
- [4] Relax: (Regular Language description for XML) <http://www.xml.gr.jp/relax/> 2000.