音声言語情報処理 34－33
(2000. 12. 22)

XML-VoiceXML変換ツールの開発

荒木雅弘, 小野 佑, 植田喜代志, 西本卓也, 新美康永
京都工芸繊維大学 工芸学部 電子情報工学科

〒606-8585 京都市左京区松ヶ崎御所海道町

TEL: 075-724-7473
e-mail: {araki,nishi,niimi}@dj.kit.ac.jp, {ono,ueda}@vox.dj.kit.ac.jp

あらまし　我々はXMLで記述された情報コンテンツを対話パターン記述言語VoiceXMLに変換するツールを開発した。このツールによって、インターネット上の情報コンテンツから半自動的に音声対話システムを構築することが可能になる。本ツールではスロットフィリング・データベース検索・説明の3つの典型的な音声対話システムのタスクに関して、それぞれに適応した対話ライブラリを用いることによって変換を行なっている。本稿ではツール開発の概要とその動作例について報告する。

キーワード　XML、VoiceXML、音声対話システム、対話ライブラリ

Development of XML-VoiceXML Converter

Masahiro Araki, Tasuku Ono, Kiyoshi Ueda, Takuya Nishimoto, Yasuhisa Niimi
Department of Electronics and Information Science,
Kyoto Institute of Technology

Goshokaidou-chou Matsugasaki Sakyo-ku, Kyoto 606-8585, JAPAN

TEL: 075-724-7473
e-mail: {araki,nishi,niimi}@dj.kit.ac.jp, {ono,ueda}@vox.dj.kit.ac.jp

Abstract　We propose a semi-automatic dialogue system generator from the Internet Information Contents. This generator translates XML documents to VoiceXML, which controls a conversation between user and computer system. We have developed three types of translation pattern: slot-filling, database search, and explanation task. Also, we took a dialogue library approach on the XML to VoiceXML translation. In this paper, we explain an outline of our project and report preliminary results in restricted task domain.

key words　XML, VoiceXML, spoken dialogue system, dialogue library

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 上の情報に対して音声によってアクセスする方法に関する研究としては、マウスによるリンクのクリックを音声に置き換えるものがある。この手法は、ユーザが何を話せばよいかが明確であるが、情報への直接アクセスが出来ない、Web の画面表示が手元に必要であり PDA などへの応用が難しいなどの問題がある。一方、情報検索の際に Web ページ上で入力する情報をキーボードから音声入力に置き換えたものとして、[2], [3] があるが、特定の形式の Web ページにしか対応していないという問題がある。

本研究では、後者の方向を発展させ、Web ページ上の情報内容に応じて対話パターンを自動生成するという立場をとる。そのために、Web ページで表現されている情報を解析する必要があるが、現在 Web ページを記述するのに主に用いられている HTML (Hyper Text Markup Language) は、閉じタグの省略や、タグの誤用 (定義リストの誤用や単純なレイアウトのためのテーブルタグの使用など) のためにページの構造が把握しにくい場合が多い [4] 。そこで、本研究では 厳密なデータ型定義 DTD (Document Type Description) に基づいた記述が義務づけられている XML (eXtensible Markup Language) を情報記述言語として位置づける。そして、その情報内容を解析し、対話パターン記述言語に変換する。対話パターン記述言語としては、CTI (Computer Telephony Integration) システムのための標準言語として期待されている VoiceXML (Voice eXtensible Markup Language) [5] を用いる。

以下、2章では本研究で対象とする XML と VoiceXML の概要を述べ、3章で タスククラスに応じた対話ライブラリについて説明する。また、4章では XML-VoiceXML 変換ツールの動作例を示し、5章で今後の課題を述べる。

2 XML と VoiceXML の概要

ここでは、変換元の XML 文書に関する制約と、ターゲットである VoiceXML の概略を述べる。

XML は情報をその内容でタグ付けするものであり、レイアウト情報は XSL (XML Stylesheet Language) などを用いて別に表現する。このことによって、コンテンツとレイアウトが分離されている。XML で記述されたページの例を図1に、またその DTD を図2示す。

```
<?xml version="1.0"?>
<ホテル>
  <名前> ABC ホテル</名前>
  <施設>
    <客室> シングル </客室>
    <客室> ツイン   </客室>
    <客室> ダブル   </客室>
  </施設>
  <電話番号>012-3456-7890</電話番号>
  <fax>012-3456-7891</fax>
  <住所>京都市 XX 区 YY 町</住所>
</ホテル>
```

図 1: XML で記述されたホテル情報

```
<!ELEMENT ホテル (名前|施設|電話番号|fax|住所) >
<!ELEMENT 名前 (#PCDATA) >
<!ELEMENT 施設 (客室 | レストラン)* >
<!ELEMENT 電話番号 (#PCDATA) >
<!ELEMENT fax (#PCDATA) >
<!ELEMENT 住所 (#PCDATA) >
<!ELEMENT 客室 (#PCDATA) >
<!ELEMENT レストラン (#PCDATA) >
```

図 2: ホテル情報の DTD

VoiceXML は、主に電話を利用した音声応答サービスにおける対話パターンを記述するための言語である。1つのルートドキュメントといくつかのアプリケーションドキュメントからなる。ルートドキュメントには大域的に参照できる変数や、ヘルプ入力の処理などを記述し、アプリケーションドキュメントは局所的な対話のパターンを記述する。各ドキュメントの主な構成要素として、ユーザから変数の値を得るための <form>と、次の VoiceXML 文書を選択するための <menu>がある。<form>の構成要素には、変数を特定する <field>、入力を促す発話を行なう <prompt>、文法を規定する <grammar>、値が定められた変数を用いて何らかの処理を行なう <block>などがある。一方<menu>は、入力を促す発話を行なう <prompt>、ユーザ発話に応じて分岐先を決める <choice>、入力がないときの処理を定める <noinput>などから構成される。

3 タスククラスに応じた対話ライブラリ

現在、音声対話システムのための対話設計ツールキットがいくつか開発されている [6]。これらの多くは対話における状態とその遷移を記述するものである。VoiceXML も本来は、これらのツールの考え方に則り、

DTMF (Dial Tone Multi Frequency) と音声認識を用いた電話による対話型サービス提供のための対話パターン記述言語として位置づけることができる。

本来の目的からすると、情報コンテンツを記述するXML と対話パターンを記述する VoiceXML は相互変換ができないように見える。しかし現在、Web は一方向的あるは双方向的に情報を流す器として機能しており、対話の目的は情報の授受であることを考えると、情報の流れという観点に着目することによって、変換の方法を考えることができる。

そこで我々は、情報の流れる方向と、捕える対話の粒度に応じた対話ライブラリを作成し、その対話ライブラリを参照することによって、特定のパターンで記述された XML 文書を VoiceXML へ変換するツールを開発した。

以下では情報の流れる方向による対話タスクの分類と、それに対応した対話ライブラリの構成を説明する。

3.1 情報の流れによるタスクの分類

現在、テレフォンショッピング・道案内・観光ガイド・各種情報検索などを行なう音声対話システムが実現されている。これらのタスクはユーザとシステム間で情報の流れる方向に着目すると、表1のように分類できる [7], [8]。

スロットフィリングクラスでは、ユーザは何をしたいかというゴールと、そのゴール達成に必要な情報を全て持っている。対話システムを使う目的は、それらの情報をシステムに伝えることによってゴール達成をシステムに依頼することである。情報の流れとしては、基本的にはユーザからシステムの一方向であり、副対話の大半は予めシステムが用意したスロットに値を埋めることを目標とするものである。

データベース検索クラスは、現在実現されている音声対話システムにおいて、最もよく用いられるタスクである。ユーザは明確なゴールを持つこともあれば、比較的漠然としたゴールを持つこともある。また、そのゴールを達成するために必要な知識は部分的にまたは制約としてしか持っていない。情報の流れは双方向で、ユーザは検索対象候補を絞るための情報を伝え、システムは検索結果を伝える。これらを繰り返して、目的とするレコードの情報を得ることがこのタスクの目標である。

説明クラスは、システムからユーザに知識を与えるタイプのタスクである。ユーザは明確なゴールを持つこともあれば、比較的抽象的なゴールを持つ場合もあり、そのゴール達成に必要な知識はほとんど持っていない場合が多い。情報の流れは基本的にシステムからユーザへの一方向であり、副対話の大半は説明と確認に費やされる。

3.2 対話ライブラリ

対話ライブラリは上記の情報の流れる方向に応じて3種類あり、それぞれ扱う対話の粒度に応じて各々3レベルで記述される。トップレベルの対話ライブラリは対話全体の構成を記述するもので、VoiceXML ファイル群の構成を規定するものである。中間レベルの対話ライブラリはトップレベル要素を詳細に記述したもので、概ね談話セグメントに対応し、個々の VoiceXML ファイルの構造を記述したものである。ボトムレベルの対話ライブラリは VoiceXML ファイル中から副対話 <subdialog>として呼び出されるもので、氏名やクレジットカード番号などの入力を行なったり、「はい／いいえ」を問い合わせるなどのパッケージである。ボトムレベルの対話ライブラリは全てのクラスのタスクで共有される。

スロットフィリングクラスにおけるトップレベルの対話ライブラリを図3に、中間レベルの対話ライブラリを図4に示す。

```
(Opening)
(Slot-filling)
{Submitting data}
if status=="OK"
        (Confirmation)
else
        (Applogize)
endif
(Closing)
```

( ): sub dialogue, { }: system function

図 3: トップレベルの対話ライブラリ (スロットフィリング)

4 XML から VoiceXML への変換

ここでは、個々のタスククラスに応じた XML の構造上の制限と、その VoiceXML 変換過程について説明する。各クラスの XML には、そのクラスを示すコメントタグを XML 宣言の直後に記述するものとする。また、トップページにはページのタイトルを示すタグがあるものとする。

4.1 スロットフィリングクラス

スロットフィリングクラスの XML の内容は、空または選択肢のリストからなるスロットの列として構成される。最初の VoiceXML ファイルはページの内容

表 1: 情報の流れる方向によるタスクの分類

情報の流れ	クラス	タスクの例
ユーザ → システム	スロットフィリング	テレフォンショッピング オンライントレード オンラインバンキング
ユーザ ↔ システム	データベース検索	文献検索 ホテル検索
ユーザ ← システム	説明	道案内 機器操作マニュアル

```
Opening{
    Message=value-of(タイトル)
    output(block, "こちらは"+Message+"サイトです")
}

Slot-filling{
    foreach Element (内容)
        Tag=tag-of(Element)
        if choice(Element)==true
             make-grammar(Element, Grammar)
             output(field, Grammar)
        else
             library-checkup(Tag, Library)
             if Library != NULL
               output(field, Library)
             else error
     endif
endif
    end
}

Confirmation{
    foreach Variable (variables)
        output(YN-question, Variable)
        output(if,call(Variable))
    end
}

Closing{
    Message=value-of(first-element)
    output(block, Message+
      "サイトをご利用いただきありがとうございました")
}
```

図 4: 中間レベルの対話ライブラリ (スロットフィリング)

を告知し、システム主導でスロットに値を埋めてゆき、値が全て埋まったらバックエンドアプリケーションに値を submit する。図 5 にホテル予約ページの XML の例を示す。

```
<?xml version="1.0"?>
<!-- slot-filling --!>
<タイトル>ABC ホテル予約</タイトル>
<内容>
  <チェックイン日> </チェックイン日>
  <チェックアウト日> </チェックアウト日>
  <客室タイプ>
    <客室>シングル</客室>
    <客室>ツイン</客室>
    <客室>ダブル</客室>
  </客室タイプ>
  <氏名>
    <姓> </姓>
    <名> </名>
  </氏名>
  <電話番号> </電話番号>
</内容>
```

図 5: ホテル予約用 XML ページ

VoiceXML ファイルはトップレベルの対話ライブラリに従って構築される。図 3 に示すように、スロットフィリングのトップレベルの最初の要素は Opening なので、図 4 の Opening の定義にしたがって、最初の VoiceXML ファイルが生成される。ここでは XML 中の <タイトル>タグの内容を、挨拶用のテンプレートの中に埋め込んで、システムへの導入発話を生成する。

続く VoiceXML ファイルはトップレベルの第 2 要素 Slot-filling の定義に従って構成される。ここでは、<内容>で囲まれた各要素に対して、値を得る副対話を生成する。<内容>の最初の要素は <チェックイン日>である。ここで、このスロットの値を得るための文法を特定する必要があるが、どの文法を用いるかは、タグ名と選択肢の有無で決めることができる。VoiceXML の組み込み文法の文法名と一致するタグ名の場合 (氏名、電話番号など) は、その組み込み文法を用いる <subdialog> 要素を用いる。また、選択肢が用意されている場合は、選択肢でラティスを構成し、それを文法とする。このいずれにも当てはまらない場合は、タグ名を形態素解

析し、ボトムレベルの対話ライブラリでカバーされている概念のうち、シソーラスで最短距離で到達できるものを探索してそれを文法とする。このようにして構成された VoiceXML ファイルを図 6 に示す (複数の VoiceXML を 1 つにまとめて記述。以下も同様)。

```
<?xml version="1.0" encording="Shift_JIS" ?>
<vxml version="1.0">
 <form>
  <block> こちらは ABC ホテル予約サイトです </block>

  <field name="チェックイン日" type="date">
   <prompt> チェックイン日は何月何日ですか </prompt>
   <help>  月と日をおっしゃって下さい </help>
  </field>

  <field name="チェックアウト日" type="date">
   <prompt>チェックアウト日は何月何日ですか</prompt>
   <help>  月と日をおっしゃって下さい </help>
  </field>

  <field name="客室タイプ">
   <prompt> 客室タイプはシングル、ツイン、ダブル
            のいずれでしょうか </prompt>
   <help> シングル または ツイン または ダブル
          とおっしゃって下さい </help>
   <grammar>
    シングル{シングル}|ツイン{ツイン}|ダブル{ダブル}
   </grammar>
  </field>

  <subdialog name="氏名" src="name.vxml">
    <filled>
      <assign name="姓" expr="氏名. 姓" />
      <assign name="名" expr="氏名. 名" />
    </filled>
  </subdialog>

  <subdialog name="電話番号" src="phone.vxml">
  </subdialog>
 ...
 </form>
</vxml>
```

図 6: ホテル予約ページから生成された VoiceXML

4.2 データベース検索クラス

データベース検索クラスの最初の検索条件を入力する XML ページはスロットフィリングクラスとほぼ同じ構成を取る。しかし、このクラスでは、データベース検索の結果を知らせるページが動的に生成され、その結果に応じて検索条件を変更するページへと制御が移されることになる。

最初の検索条件入力対話は、スロットフィリングと異なり全てのスロットを埋める必要がないので、混合主導型の構成を取る。また文法は、フィールド名を基準に、組み込み文法を使うか、またはデータベースの値から生成したものを用いるかのいずれかとなる。

各フィールドは初期値として NULL を持ち、ユーザ発話によって言及されたものだけが値を持つことになり、データベース検索の条件として用いられる。図 7 にデータベース検索クラスの VoiceXML ファイルを示す。

```
<?xml version="1.0" encording="Shift_JIS" ?>
<vxml version="1.0">
 <form>
  <grammar src="ホテル検索.gram"
           type="application/x-jsgf" />
  <block>こちらはホテル検索サイトです </block>

  <initial name="start">
   <prompt> どのような条件でホテル検索を
            行ないますか </prompt>
   <help> 地域 または ホテルタイプ または
          ホテル名 を指定して下さい </help>
  </initial>

  <field name="地域" expr="NULL">
   <prompt> 地域を指定してください </prompt>
   <grammar src="地域.gram"
            type="application/x-jsgf" />
  </field>

  <field name="ホテルタイプ" expr="NULL">
   <prompt> ホテルタイプを指定してください </prompt>
   <help>シティまたはリゾートを指定して下さい</help>
   <grammar>
      シティ {シティ} | リゾート {リゾート}
   </grammar>
  </field>

  <field name="ホテル名" expr="NULL">
   <prompt> ホテル名を指定してください </prompt>
   <grammar src="ホテル名.gram"
            type="application/x-jsgf" />
  </field>

  <block>
    <submit next="seach"
            namelist="地域 ホテルタイプ ホテル名"/>
  </block>

 </form>
</vxml>
```

図 7: ホテル検索ページから生成された VoiceXML

4.3 説明クラス

説明クラスでは、XML ファイルは木構造のドキュメントと見なすことができる。そこで、VoiceXML の最初のファイルは、サイトの説明を行なった後、知りたい要件の指定を促す発話を生成して、対応するコンテ

ンツを探すという構成になる。もし、ユーザが要件を指定すれば、木構造の探索を行なって該当する部分を段階的に説明する。また、ユーザが要件を指定できなかった場合は、システム主導で選択肢を示しながら木構造を段階的に降りてゆく対話となる。図8に説明クラスの VoiceXML ファイルを示す。

```
<?xml version="1.0" encording="Shift_JIS" ?>
<vxml version="1.0">
 <form>
  <block> こちらは ABC ホテル情報サイトです </block>
  <field name="item">
   <prompt>
     どのような情報をお知りになりたいですか
   </prompt>
   <grammar src="top.gram"/>
   <help> 客室タイプ、住所、電話番号、FAX 番号、
          交通の便 のいずれかを指定して下さい </help>
  </field>
  <if cond="item==客室タイプ">
    <goto next="roomtype.vxml">
    ...
 </form>
</vxml>
```

図 8: ホテル情報ページから生成された VoiceXML

ユーザが要求した情報が、番号付き (enumerate) で表現されている時は、その全てを順番に説明する。また、項目 (itemize) で表現されている時は、いずれかを選択させて説明する。

5 おわりに

本稿では我々が開発した XML-VoiceXML 変換ツールの概要について述べた。このツールでは、我々が定義した3種類のものにあてはまる単純なページのみが変換対象であるが、3種類のクラスを組み合わせることによって、ある程度複雑なページもカバーできると考えている。今後はボトムレベルの対話ライブラリを充実させ、複数のタスクを行なうサイトを実際に構築して本ツールの評価を行なう予定である。

謝辞

本研究は財団法人立石科学技術振興財団の援助を受けて行なわれた。関係各位に感謝します。

参考文献

[1] 西本卓也, 藤澤正樹, 新美康永: 音声ウェブブラウザ VOXplorer の評価 . 日本音響学会平成 10 年度秋季講演論文集 1-R-26, pp.169-170, 1998.

[2] S. Issar: A speech interface for forms on WWW. In *Proc. of EUROSPEECH'97*, pp. 1343-1346, 1997.

[3] 中野崇広, 甲斐充彦, 中川聖一: フォーム型情報検索サービスのための音声／ペンタッチ入力インタフェースの比較. 情報処理学会研究会資料, SLP-33-11, 2000.

[4] S. Sorderland: Learning to extract text-based information from the world wide web. In *Proc. of 3rd International Conference on Knowledge Discovery and Data Mining*, 1997.

[5] VoiceXMLForum: Voice extensible markup language VoiceXML, http://www.voicexml.org/, 2000.

[6] P. Bohlin, J. Bos, S. Larsson, I. Lewin, P. Ljunglf, and D. Milward: Survey of existing interactive systems. Deliverable D1.3. Trindi Project, http://www.ling.gu.se/research/projects/trindi/publications.html, 1999.

[7] 荒木雅弘, 平田大志, 駒谷和範, 堂下 修司: 音声対話システム構築のための対話ライブラリ. 人工知能学会研究会資料, SIG-SLUD-9901-1, 1999.

[8] M. Araki, K. Komatani, T. Hirata, and S. Doshita: A dialogue library for task-oriented spoken dialogue systems. In *Proc. IJCAI Workshop on Knowledge and Reasoning in Practical Dialogue Systems*, pages 1–7, 1999.