

Javaを用いた VoiceXML のフレームワーク

4R-9

石塚 隆平[†] 山崎 祥行[†] 大野 澄雄[†] 奥 正廣[†] 千種 康民[†] 小池 隆[‡]

[†]東京工科大学 [‡]富士ソフト ABC

1 はじめに

ボイスポータルサイトやカーナビゲーションなど、近年音声対話アプリケーションが注目されている。しかし音声対話専用のフレームワークは存在しない。そこで本研究では VoiceXML (以下 VXML) のプログラムに MVC モデルを用いた VXML 専用のフレームワークを開発した。

2 VXML フレームワーク

VXML フレームワークは、Web アプリケーションフレームワーク Struts 等で採用されている Model-View-Controller モデルに基づき、Java Web アプリケーションサーバに実装されている。VXMLサーバからネットワーク経由で図1の順にアクセスする。VXMLサーバからみて VXML フレームワークを実装した Web アプリケーションサーバはドキュメントサーバとして動作する。VXML フレームワークは VXML2.0 において未実装の「履歴機能」「終了機能」「トップ機能」を実装している。

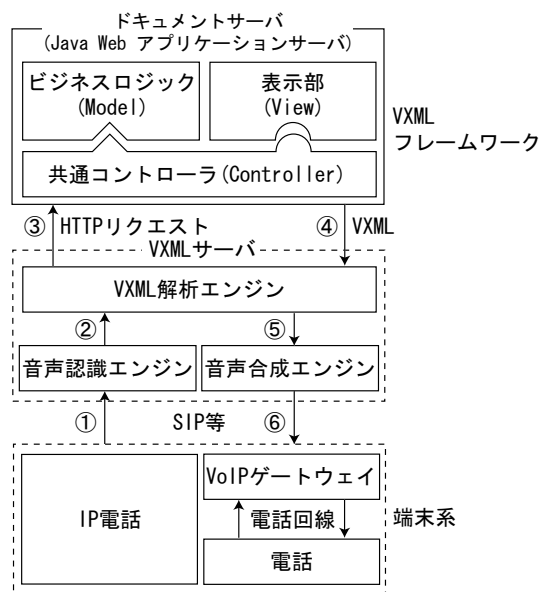


図 1: VXML フレームワークへのアクセス手順

Voice XML Framework for Java
 Ryuhei ISHIZUKA[†], Yoshiyuki YAMAZAKI[†], Sumio OHNO[†],
 Masahiro OKU[†], Yasutami CHIGUSA[†], Takashi KOIKE[†]
[†]Tokyo University of Technology, [‡]FUJISOFT ABC, Inc.
 E-Mail chigusa@cc.teu.ac.jp
 http://www.teu.ac.jp/chiit/

2.1 ページ遷移の処理

ページ遷移はすべてフレームワーク内の Controller である共通コントローラにより制御される。外部からのアクセスはすべて同一のフレームワークの共通コントローラによって処理され、ビジネスロジックとの関係を記述した XML 設定ファイルにしたがって転送される (図2)。

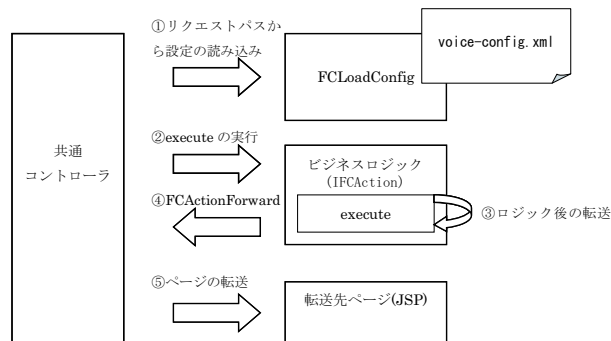


図 2: ページ遷移の流れ

2.2 リクエストデータの処理

VXML フレームワークのコントローラはリクエストを DynaBean を用いて、作られたアクションフォームに BeanUtils を用いて動的に格納する (図3)。

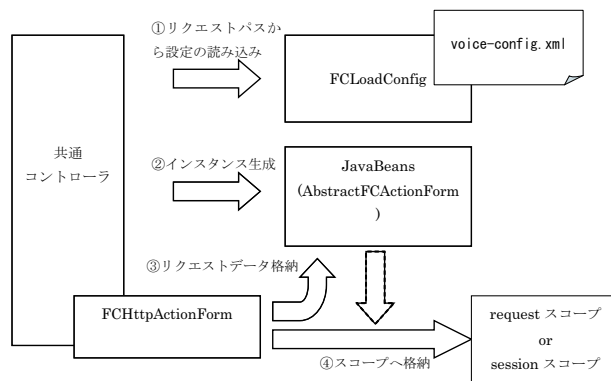


図 3: リクエストデータの流れ

2.3 履歴機能の実装

VXML フレームワークでは履歴データ (Stack) をドキュメントサーバの Session スcopeに格納する。格納する際は VXML フレームワークにアクセス時の URI を履歴データに push し、取得する際は同様に履歴データから pop して取得をおこなう。Session は電話がかかってきた時に開

始され、終了する際にメモリから削除する(図4)。履歴機能呼び出すためには「/FCBack.do」のURIにアクセスする。

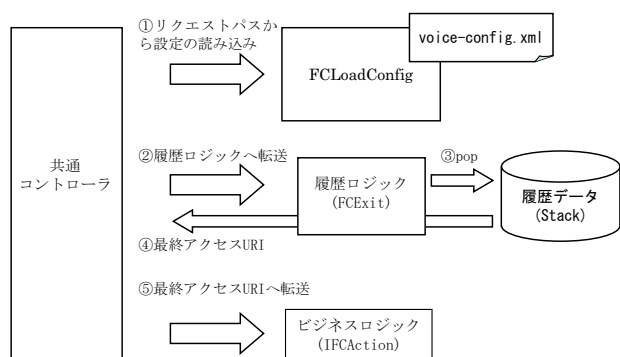


図 4: 履歴機能の流れ

2.4 終了機能の実装

表示部で <disconnect> タグを直接使うと履歴データがドキュメントサーバに残る。終了機能はアクセス中の履歴データを削除し、VXMLサーバと電話回線を切断するVXML(<disconnect>タグ)をVXMLサーバに送信するロジックである。終了機能呼び出すためには「/FCExit.do」のURIにアクセスする。

2.5 トップ機能の実装

トップ機能はユーザがVoiceサーバに最初にアクセスしたページへジャンプするロジックである。Sessionスコープに存在するStackのURIから先頭のもの取得し、取得したURIへジャンプをおこなう。トップ機能呼び出すためには「/FCTop.do」のURIにアクセスする。

3 VXMLフレームワークの機能拡張

開発者は履歴機能を実装するために「FCBack.do」へのリンクを、終了機能を実装するために「FCExit.do」へのリンクも実装しなければならない。FCBack, FCEdit, FCTopはVXMLフレームワーク内で使われるものである。

また、表示部を実装する際、実装する人はVXMLを記述しなければならない。VXMLはWebをベースにしたものであるが、HTMLとは大きくことなり、HTMLWebアプリケーション開発より複雑なものである。

3.1 Voiceカスタムタグ機能

VXMLフレームワークは出力部分にJ2EEのJSPをもちいる。本研究ではJSPを記述するときに複雑なVXML

を半自動的に記述できるよう、VXMLを完全にカプセル化したJSPカスタムタグを実装している。VXMLで定義された基本的なタグの代わりに、フレームワークの提供するJSPカスタムタグを使用することができる。JSPカスタムタグがVXMLをカプセル化することにより、VXMLフレームワーク内部で使われるURIを開発者から隠すことが可能である。

また電話において数字を選択肢としてもちいることが多い。そこで選択肢を自動的に上からナンバリングをおこない、数字の割り振りをプログラマが実装することなくユーザが数字で選択できる機能をVXMLフレームワークに実装する(図5)

図5は「次の中から選んでください。いち、りんご、に、おれんじ。さん、ばなな。」と発話されるJSPのサンプルである。

```

<%@ page language="java" contentType=
  "application/voicexml+xml; charset=Shift_JIS"
  pageEncoding="UTF-8" %>
<%@ taglib uri="TagCore.tld" prefix="fc" %>
<fc:vxml>
  <fc:prompt>次の中から選んでください</fc:prompt>
  <fc:form key="fruit">
    <fc:choice value="林檎">りんご</fc:choice>
    <fc:choice value="オレンジ">おれんじ</fc:choice>
    <fc:choice value="バナナ">ばなな</fc:choice>
    <fc:submit next="/selected.do" />
  </fc:form>
</fc:vxml>
  
```

図 5: サンプルソース

図5を同じ動作をするVXMLと比較する。(表1)

表 1: 比較データ

カスタムタグ	使用	未使用	本実装の効果
文字数	398 文字	890 文字	44.7%
タグ	8 個	18 個	44.4%

4 まとめ

本研究で開発したVXMLフレームワークにVoiceポータルのサービスを実装すると、表示部に関しては最大約45%程度までソースが簡略化され、MVCモデルによるロジックの再利用、リクエストデータの自動格納により、開発効率が従来のJ2EEより格段に向上した。

今後はVXML2.0へのより忠実な実装と大規模アプリケーションへの適用を試みている。

参考文献

- [1] The Apache Struts Web Application Framework:
<http://struts.apache.org/>