

音声認識を利用した録音音声ファイル連結の自動化

村山 智子 田原 義則 馬場 正浩

日本アイ・ビー・エム株式会社 大和ソフトウェア開発研究所

1. はじめに

音声対話システムでは応答メッセージとして人の声の録音音声ファイルが用いられることがある。しかし、そのメッセージに変数が含まれる場合、音声ファイルの作成作業とアプリケーションの実装においてアプリケーション開発に負担がかかっているのが現状である。

筆者らは、アプリケーションにかかっているこの負担の削減を目的として、音声サーバの機能拡張を検討した。その結果、音声サーバが音声認識機能を利用して録音音声ファイルを自動連結する手法を考案した。この手法を IBM 音声サーバに適用しその効果を検証した。本稿ではその概要を述べる。

2. 従来アプリケーションにかかっている負担

メッセージに変数を含む場合、従来の音声サーバ上で移動するアプリケーションの開発にかかっている負担を述べる。

2.1. 音声ファイル作成作業面

音声ファイルは、変数前部分、変数部、変数後部分に分割して録音されている必要がある。例えば、商品名を変数部として含む「ご注文は{お米}ですね」というメッセージの場合、「ご注文は」と「お米」と「ですね」という音声ファイルが必要である。このことが、音声ファイル作成作業において、品質、作業量に関して次のような負担を生じさせる。

- 録音者は文章として不自然に分断されたパーツを話さなくてはならないので不自然な発声になりやすい。全文をひと続きに読み上げて録音した音声声を音声ファイルエディターで分割する方法があるが音声ファイル編集者はアプリケーションの設計を正確に把握している必要がある。
- 音声ファイルをスムーズに連結するためには、音声ファイルの初めと終わりの無音部を切り取るなど連結後の音声を予想しながら編集する必要がある。文をあらかじめ変数位置で分割した状態の音声ファイルを用意するために音声ファイル数が多くなる。ファイル数に比例して編集作業量が増えるので、ファイル数の増加は問題である。

2.2. アプリケーションの実装面

アプリケーションの実装においては以下のような負担がある。

- 変数に対応する音声ファイル(以下、変数音声ファイル)を特定し、変数前部分、変数部分、変数後部分、に分かれた音声ファイルを順々に再生する、というロジックをアプリケーションが独自に開発する必要がある[1][2]。
- メッセージを変更するために、アプリケーションロジックを変更する必要がある。

3. 改善方針

変数部を含む全文をひと続きに録音した音声ファイル(以下、全文音声ファイル)を受け付け、実行時に自動的に変数音声ファイルを特定し自動連結して再生する、という機能を音声サーバが提供することにより、従来の上記負担を削減する。この方針を実現することにより、次のような効果が期待できる。

- ひと続きに全文を読み上げることで録音者の自然な発声が可能になるので、録音ファイルの品質が向上する。
- 全文音声ファイルをそのまま利用することが可能になることにより文を分割した音声ファイルが不要になり、音声ファイル数が減る。その結果編集作業量が削減される。
- 変数音声ファイルの特定、連結、再生は、音声サーバが自

動的に行うので、アプリケーションが独自のロジックを持つ必要がなくなる。

- アプリケーションロジックの変更を伴わずに、文章の変更が可能になる。

4. 改善策

4.1. アプリケーションに定めるルール

4.1.1. 用意する音声ファイル

全文音声ファイルを用意する。変数部は、ある特定の単語に置き換えて録音する。この単語は、変数部として認識するようにあらかじめ音声サーバを対応させたものである。例えば、省略時には「不明」という単語としアプリケーションによる設定も可能にする。変数音声ファイル名は、「{変数}.wav」とする。ひとつ以上の変数音声ファイルをクラス化することもできる。クラスはクラス名と同じ名前のクラス専用フォルダを持ち、クラスに属する変数音声ファイルを置く。

4.1.2. スピーチユーザインタフェースの定義

音声アプリケーションは、音声ファイルを指定する際その音声ファイルが変数部を持つかどうかを示す目印をつける。さらに、変数部を持つ音声ファイルには以下の変数情報を付加する。

- 変数部を含むテキスト全文
- 挿入する変数
- 挿入位置
- (オプション)変数用音声ファイルが属するクラス名

4.1.3. VoiceXML への適用例

上記定義を VoiceXML[3]に適用する例を示す。

運用時には人の声を録音した音声ファイルを用いる音声アプリケーションでも、開発中は音声ファイルがない状態で開発を行う必要があるので音声合成を利用する場合がある。VoiceXML で音声ファイル再生のために定義されている<audio>タグはこのような利用環境に対応するため、音声ファイル名と文字列を併記させ音声ファイルが存在すればそれを再生し、音声ファイルが再生不可能な状態であれば文字列を音声合成する、という仕様になっている。以下、この文字列部を代替テキスト部と呼ぶ。

例: <audio src="welcome.wav">ようこそ</audio>

機能的にもフォーマットのにも、<audio>タグ本来のこの仕様に則った状態で上記定義を適用する。

- 音声ファイルが変数部を持つかどうかを示すために、<audio>タグの src 属性フィールドを利用する。音声ファイル名の先頭に'#'をつけて変数挿入の必要を示す。

変数部を持つ音声ファイルに情報を付加するために、<audio>タグの代替テキスト部を利用する。

- 変数を含むテキスト全文を記述する。
- 挿入する変数を、<value>タグで指定する。
- 挿入位置に、<value>タグを挿入する。
- クラスに属する場合は、クラス名を type 属性で指定した<say-as>タグで、<value>タグを囲む。

<value>タグと<say-as>タグも既存のタグであり、この利用の仕方は、本来の仕様の範囲である。

4.2. 音声サーバの機能拡張

音声サーバは、音声認識機能を利用して、全文音声ファイルを適切な変数挿入位置で自動分割した後、変数音声ファイルの自動連結を行う[図1]。以下に本手法を適用した音声サーバの

処理の詳細を述べる。全て実行時の処理である。



図1: 音声ファイルの自動分割・連結

4.2.1. 連結必要性の有無の判定

音声ファイルが変数部を持つかどうかを音声ファイル名の目印の有無("#")で判定し従来の単純な再生処理と自動連結を伴う再生処理に岐する。以下に自動連結を伴う再生処理を述べる。

4.2.2. 音声認識グラマーの生成

変数を含むテキスト全文を、変数と変数以外の部分に分割して各パーツをルールに持つグラマーを動的に生成する。変数部は変数挿入位置を示す特定の単語に置換したルールを定義する。

4.2.3. 音声認識機能による自動分割

音声認識エンジンに対して、生成したグラマーを有効化し全文音声ファイルを入力する。認識エンジンは、入力された音声ファイルをグラマーに従い音声認識し、認識単位であるルール毎に音声ファイルとして出力する。こうして全文音声ファイルから変数部と変数部以外に分割された音声ファイルを生成する。

4.2.4. 変数音声ファイルの特定

変数用音声ファイルを、「{変数}.wav」というファイル名で検索する。見つからない場合は変数テキストを音声合成する。

4.2.5. 音声ファイルの自動連結

全文音声ファイルを分割して生成した音声ファイルと、検索した変数音声ファイル、あるいは、変数テキストの音声合成、を代替テキスト部が示す順に再生する。

4.2.6. パフォーマンスの考慮

パフォーマンス向上を考慮し、音声サーバはいったん分割した音声ファイルをアプリケーション終了時まで保管する。分割した音声ファイル断片の存在状態を記憶し、分割済み音声ファイルが存在するものはそれを利用する。音声サーバの既存のキャッシュ管理仕様に従い、適切なファイル削除を実行する。

5. 効果の検証

本手法を実装した音声サーバを利用することによりアプリケーションの開発にかかる負担がいくかに削減されるかについて効果を検証した。検証は以下の構成で行った。

- 機能拡張対象の音声サーバ：IBM WebSphere Voice Server
- 音声認識エンジン：IBM 音声認識エンジン
- スピーチユーザインタフェース：VoiceXML

5.1. アプリケーションコードの比較

従来の音声サーバ上で稼動するアプリケーションでは、文を変数位置で分割した音声ファイルをあらかじめ用意して、実行時に変数値から変数音声ファイル名を特定し変数前後と変数の音声ファイルをそれぞれ<audio>タグで指定する、という実装が1例として考えられる。

「ご注文の商品は{お米}、数量は{ひとつ}でよろしいですか。」({ } 内が変数)という文を例に実装例を図2に示す。

用意する音声ファイル
part1.wav 「ご注文の商品は」
part2.wav 「数量は」
part3.wav 「でよろしいですか?」
お米.wav 「お米」
ひとつ.wav 「ひとつ」

VoiceXML ドキュメント
<script>
<![CDATA[
function getAudioName(product) {
var audioname;

```
audioname = product + ".wav";  
return audioname;  
}  
]]>  
</script>  
<audio src="part_1.wav">  
ご注文の商品は、  
</audio>  
<audio expr="getAudioName(Product)">  
<value expr="Product"/>  
</audio>  
<audio src="part_2.wav">  
数量は、  
</audio>  
<audio expr="getAudioName(Quantity)">  
<value expr="Quantity"/>  
</audio>  
<audio src="part_3.wav">  
でよろしいですか?  
</audio>
```

図2: 従来のサーバ用アプリケーション

一方、同じ文を実装するために、本手法を適用した音声サーバを利用する場合アプリケーションは図3のようになる。

用意する音声ファイル
sample.wav 「ご注文の商品は不明、数量は不明でよろしいですか」
お米.wav 「お米」
ひとつ.wav 「ひとつ」

VoiceXML ドキュメント
<audio src="#sample.wav">
ご注文の商品は、<value expr="Product"/>、
数量は<value expr="Quantity"/>でよろしいですか?
</audio>

図3: 本手法適用済みサーバ用アプリケーション

本手法を実装した音声サーバ上では、用意する音声ファイルが減ること、音声アプリケーションはスクリプト関数を開発する必要がなくなっていること、変数部分に依存しないシンプルな記述になっていること、がわかる。音声アプリケーションでは応答文に音声合成を使い開発を開始し、その後音声ファイルを使うように変更することが多い。本手法では応答文中の分割部分毎に必要なコード変更が必要なく、一箇所の変更で済み、従来型の開発手法においても効率的であることがわかる。

5.2. 再生音声の比較

本手法を実装した音声サーバが出力した音声と、従来型のアプリケーションが手動で連結した音声の聞き比べをした。全文音声ファイルを録音する際に、変数位置を示す「不明」とこれに続く文章との間を区切って発声することで、自動分割では良い結果が得られ、連結後の文章も従来のものよりスムーズで自然な音声を得られた。

6. まとめ

音声認識機能を利用して録音音声ファイルを自動連結する、音声サーバの機能拡張を考案した。この手法を、VoiceXML をサポートする IBM 音声サーバに適用しその効果を検証した。その適用例と結果を示し、変数を含む音声ファイル再生においてアプリケーション開発の従来の負担が、この手法を実装した音声サーバ上で削減されることを確認した。本手法を実装した音声サーバによる出力は、従来型のアプリケーションによる手動連結によるものよりスムーズで自然な音声を得られた。VoiceXML への適用例を示し、アプリケーションが音声サーバのこの機能拡張を利用するために VoiceXML の既存の仕様を侵害することなく定義可能であることを示した。

7. 参考文献

- [1]Telme Studio VoiceXML 2.0 tutorials
(<http://studio.telme.com/vxml2/ow/javascript.html>)
- [2]Cisco VoiceXML Programmer's Guide
(http://www.cisco.com/univercd/cc/td/doc/product/software/ios122/rel_docs/vxmlprg/)
- [3]W3C VoiceXML Version 2.0, (<http://www.w3.org/TR/voicexml20/>)