音声対話コンテンツのパッケージ化とその配信システム

石川 博規[†1]　山本 大介[†1]　高橋 直久[†1]

概要：本稿では，ユーザ生成型コンテンツ共有の概念を音声対話コンテンツに導入する手法を提案する．提案手法により，音声対話システムの利用が活発化すると考えられる．しかし，一般に音声対話コンテンツを共有するシステムがない．そこで，スマートフォンに搭載した音声インタラクションシステム構築ツールキット MMDAgent における音声対話コンテンツを機能ごとに分割しパッケージ化する手法と，それらの音声対話コンテンツ配信の仕組みを提案する．建物案内や天気予報などの機能単位で，パッケージ化された音声対話コンテンツを複数同時並列に実行することによって，音声対話の容易かつ自由度の高い拡張を可能にする．また，それらパッケージ化された音声対話コンテンツを配信する仕組みを提案し，その実現法について述べる．

1. はじめに

近年，YouTube，AppStore 等のユーザ生成型のコンテンツ共有システムが普及している．これらのユーザ生成型コンテンツ共有システムでは，多数のユーザが作成した多数のコンテンツがコンテンツ共有 Web サービスによって共有される．ユーザ生成型コンテンツ共有システムの特徴として，ユーザの要望や需要に沿ったコンテンツ作成が行われることで，コンテンツの利用が促進されシステムの利用も活発になることが挙げられる．

一方で Siri のように音声対話システムが普及している．また，音声インタラクションシステム構築ツールキット MMDAgent[1] が提案されている．MMDAgent は対話スクリプト（FST スクリプト）を編集することで自在に音声対話をさせることができ，また対話スクリプトはユーザも作成することが可能である．しかし，現在の MMDAgent の主な利用形態は，個人で対話スクリプトを作成し自分の端末でのみ実行するというものであった．そこで，ユーザの作成した対話スクリプトを多くのユーザが自由に共有することができれば，MMDAgent の音声対話コンテンツの作成と利用が活発となり，音声対話コンテンツの利用と活用の幅が広がると考えた．

現状における問題点として，一般に音声対話コンテンツに適した共有システムが提案されていないことが挙げられる．そのため，ユーザは他のユーザが作成した音声対話コンテンツを容易に利用することができない．そこで，本研究ではユーザ生成型コンテンツ共有の概念を音声対話コンテンツに導入することによって，音声対話コンテンツの作成と利用を促進させることを目的とする．

実現にあたり以下の要求を満たす必要がある．

- 音声対話コンテンツを内容の把握と編集が容易になる形式でパッケージ化する
- ユーザが簡単に音声対話コンテンツを追加できるインターフェースを実装し，まるでアプリを追加するかのように音声対話の内容を拡張できるようにする
- 複数の音声対話コンテンツを同時に実行したときの競合発生を回避する

2. 従来手法とその問題点

2.1 音声インタラクションシステム構築ツールキット MMDAgent

MMDAgent は FST スクリプトの編集によって自由に音声対話のシナリオを実行することができる音声インタラクションシステム構築ツールキットである．Julius[2] と OpenJTalk[3] によるリアルタイムの音声認識・音声合成，さらに 3D グラフィック描写を組み合わせることでインタラクティブ性の高い音声対話を可能にしている．

音声対話のシナリオは FST スクリプトで記述される．FST スクリプトは状態遷移を表現しており，図 1 のように遷移元の状態番号と遷移先の状態番号，状態遷移条件，状態遷移時のコマンドからなっている．状態遷移条件を満たすイベントが発行されると状態が遷移し，コマンドを実行する．

[†1] 現在，名古屋工業大学 大学院工学研究科 情報工学専攻
Presently with Nagoya Institute of Technology Graduate School of Engineering Department of Computer Science and Engineering

また，MMDAgent は複数の FST スクリプトを同時並列に実行することが可能である．起動時に実行されるメイン FST スクリプトの他に，新たに FST スクリプトを実行することで，様々な異なるタスクに並列に対応可能である．

本研究では Android 版 MMDAgent[4] を用いる．Android 版 MMDAgent はネットワーク通信無しで高速に音声認識，音声合成，3D モデル描写が可能である．

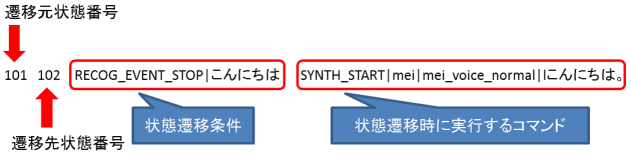

図 1　FST スクリプト．MMDAgent の音声対話を記述する．

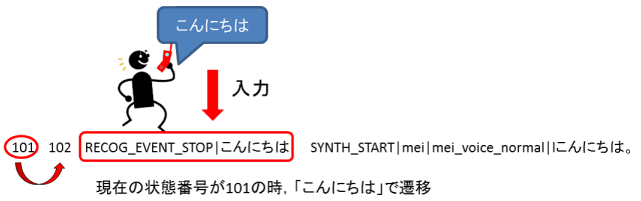

図 2　FST の状態遷移．状態番号と発行されたイベントが合致した時に状態が遷移する．

2.2　問題点

音声対話コンテンツを編集，配信する上での問題点　FST スクリプトは複雑で扱う対話の数が多くなると，状態数と認識キーワード数が多くなってしまい，また，一部のみ変更したいときでも，全体の状態遷移の関係を考慮して修正を加える必要がある．また，FST スクリプトは状態遷移図をテキストで記述したものであり，一見して対話内容を把握することが難しい．

音声対話コンテンツを追加する上での問題点　従来手法では音声対話コンテンツを追加するために，FST スクリプトを直接書き換える必要があり，FST スクリプトを編集する知識が必要であるなど，一般のユーザには困難であった．また，ユーザが追加した音声対話コンテンツを再起動なしに実行することができなかった．

競合の発生　複数の音声対話コンテンツを同時に実行すると競合が発生する可能性がある．最も競合するのが多いと思われる部分は，音声認識キーワードを状態遷移条件として状態遷移する部分である．同じ音声認識キーワードによって状態遷移する記述が，実行中の異なる FST スクリプト間に存在した場合，競合が発生する可能性がある．図 3 に例を示す．同じ「やめて」という発話に反応して二つの異なる FST スクリプトの状態が遷移する．図 3 の場合，MMDAgent が「ごめんなさい」と「はい。案内を終了します。」を同時に言おうとするため，音声が完全に再生されない，モーションが不自然になる等の問題が発生する．

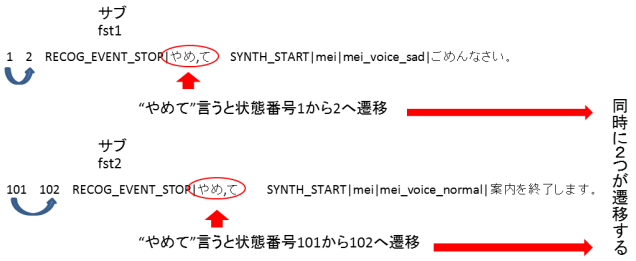

図 3　認識キーワードの競合発生の例

3.　提案手法

3.1　提案システムの概要

提案システムの利用法を図 4 に提案システムの構成を図 5 に示す．提案システムのユーザは音声対話コンテンツの利用者であり，同時に作成者にもなりえる．ユーザが生成する音声対話コンテンツを他のユーザと簡単に共有することで，MMDAgent の利用活発化がはかられると考えられる．提案システムはコンテンツ管理サーバと Android 端末からなり，ユーザの作成した音声対話コンテンツをサーバにアップロードし，ユーザは端末の操作により必要な音声対話コンテンツをダウンロードできる．ダウンロードされた音声対話コンテンツは端末の管理機能により任意のタイミングで実行できる．

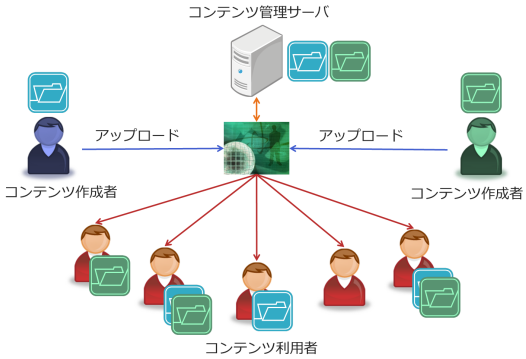

図 4　提案システムの利用法

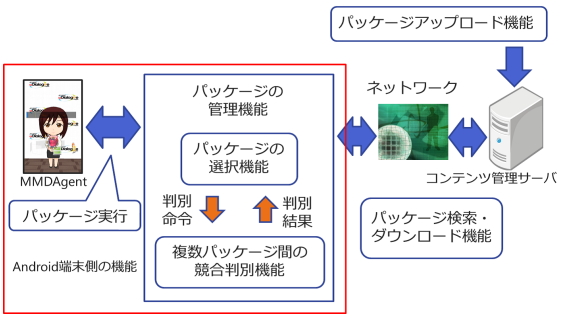

図 5　提案システムの構成図

3.2 音声対話コンテンツのパッケージ化

音声対話コンテンツを適切に 1 つのパッケージにまとめることで，配信と管理を容易にする．図 6 のようにパッケージ化されている．本稿ではパッケージ化された音声対話コンテンツを単にパッケージと呼ぶ．パッケージの内容は以下の通り．

FST スクリプト　MMDAgent の音声対話を記述したスクリプトファイル

アイコン画像　音声対話コンテンツの内容を表す画像

XML ファイル　音声対話コンテンツのメタデータを保存

リソース　追加のモデルデータ・モーションデータ

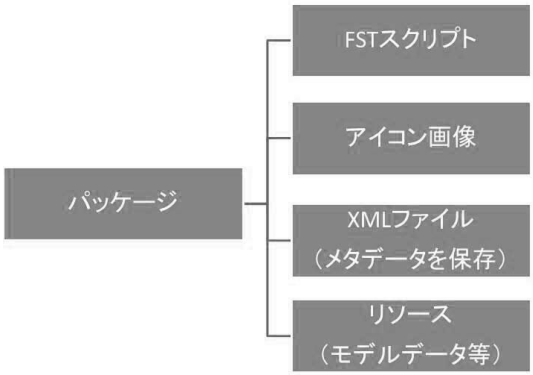

図 6　パッケージのデータ構造

音声対話コンテンツのタイトル，タグ，説明等を記述した XML ファイルを用意することで内容の把握を容易にし，また検索も容易にできる．追加のリソースも一緒にパッケージ化することで，パッケージをダウンロードするだけで実行環境を整えることが可能になる．

個々の FST スクリプトには単一の機能のみを記述する．これにより FST スクリプトの状態数と認識キーワード数が少なくなり，管理が容易になると同時に，その音声対話コンテンツの内容が明確になることで，共有しやすくなる．これら単一の機能の音声対話コンテンツを複数組み合わせ並列に実行することで多彩な状況に対応することができる．使用例として以下のようなものが考えられる．

(1)「雑談」の音声対話コンテンツを実行し，音声対話による雑談を楽しむことができる．

(2)「施設案内」,「天気予報」の 2 つを実行し，施設を案内されながら天気予報を聞くことができる．

(3)「スケジュール」,「ニューストピック」,「天気予報」の 3 つを実行し，通勤前に今日の予定を立てることができる．

3.3 パッケージ検索・ダウンロード機能

ユーザはコンテンツ管理サーバからパッケージ化された音声対話コンテンツを Android 端末にダウンロードすることができる．パッケージ検索・ダウンロード機能は Android アプリケーションによって実装されている．

ユーザは図 7 のように，Android 端末から上記アプリケーションを用いてサーバにアクセスし，パッケージの検索をすることができる．

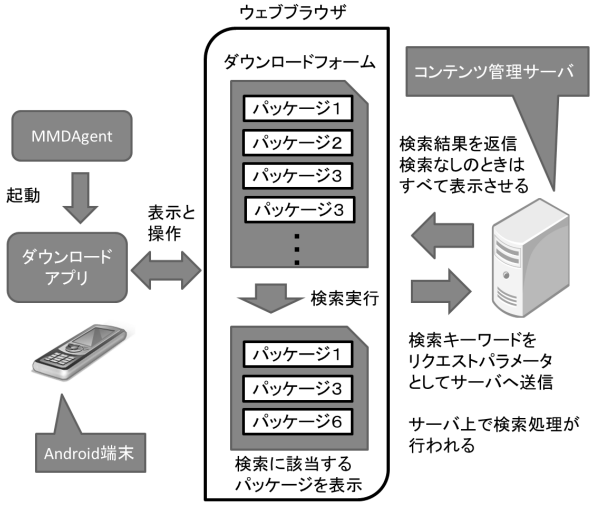

図 7　ダウンロードフォームの表示とパッケージの検索．パッケージを専用のフォームを操作して，検索・ダウンロードできる．

ユーザが必要としている音声対話コンテンツを素早く見つけ出すためには，検索機能が必須である．パッケージ化された音声対話コンテンツにはメタデータを記述した XML ファイルが含ませており，その中のタグ情報を利用することで検索機能を実現する．タグには例として「施設案内」,「雑談」などの音声対話コンテンツの属性が登録される．入力フォームに検索タグを入力するか，タグのリンクを選択することでコンテンツ管理サーバに検索のリクエストを行う．検索タグはのリクエストパラメータとしてサーバに送信され，検索処理は全てサーバ上で行われる．HTML 形式で結果が返され，端末のウェブブラウザで表示する．パッケージの一覧を表示する WEB ページとパッケージの詳細を表示する WEB ページがあり，リンクを選択することで遷移する．パッケージの詳細ページではパッケージ中のＸＭＬファイルの記述からタイトル，タグ，コメントによる説明等のメタデータを表示することで，音声対話コンテンツの内容を把握しやすいようにしている．また，app://download で始まる特別なリンクを用意し，HTTP リクエストパラメータからダウンロードするパッケージの URL を取得することで，目的のパッケージをダウンロードする．

3.4 パッケージ管理機能

Android 端末にダウンロードされたパッケージは端末の外部ストレージに展開され保存されている．これらのパッケージ群を管理する機能について述べる．

ユーザが自分の思い通りに MMDAgent の音声対話機能

を拡張していくためには，ダウンロードした音声対話コンテンツを任意のタイミングで切り替えられながら利用できる必要がある．そこで，Android の GUI を利用して分かりやすく操作しやすいパッケージ管理のインターフェースを実装した．パッケージ群はリスト状に上下に並んで表示され，スイッチ操作で実行不実行を切り替えることができる．

また MMDAgent を再起動した際に，前回のパッケージ実行状況を再現することも可能する．端末の外部ストレージにどのパッケージを実行しているかを保存する XML ファイルを用意している．XML の例を図 8 に示す．パッケージを新たにダウンロードした場合は XML にデータが追加され，実行状況が変化した場合はデータが書き換えられる．MMDAgent 起動時にこの XML が読み込まれ，それに従いパッケージ実行の初期状態を求める．

```
<?xml version="1.0" encoding="WINDOWS-31J"?>
- <MMDAgentSettings>
    <contents name="">ON</contents>
    <contents name="">OFF</contents>
    <contents name="">ON</contents>
    <contents name="">OFF</contents>
    <contents name="">ON</contents>
</MMDAgentSettings>
```

図 8　パッケージの実行状況を保存する XML

また，複数のパッケージを並列に実行すると，複数の FST スクリプトが並列に実行されるため，競合が発生する可能性がある．そこで，新たにパッケージを実行しようとするときに競合判別を行う．競合が最も起こりやすいと考えられる認識キーワードの競合について判別する．認識キーワード競合判別のアルゴリズムを以下に示す．

ステップ 1　FST スクリプトを 1 行ずつ読みだす
ステップ 2　1 行をスペースで分解する．
ステップ 3　遷移条件の部分を取り出す．
ステップ 4　パイプ (|) で分解する．
ステップ 5　遷移条件となっている入力単語を取り出す．
ステップ 6　取り出した単語群を保持する．

上記ステップ 1 から 6 をすでに実行されているパッケージの FST スクリプトとこれから実行しようとしているパッケージの FST スクリプト，及びメイン FST スクリプトに対して実行し，遷移条件となっている入力単語を全て取り出す．

これから実行しようとしているパッケージが複数ある場合，ステップ 7 からステップ 9 の操作を実行する．そうでない場合はステップ 7 からステップ 9 を飛ばす．

ステップ 7　実行しようとしているパッケージを 2 つ組の組み合わせで分ける．組の数はパッケージ数を n とすると，
${}_nC_2 = \frac{n*(n-1)}{2}$ となる．

ステップ 8　2 つ組になったパッケージの内片方の FST スクリプトから取り出された単語を 1 つずつ，もう片方のパッケージの FST スクリプトから取り出された単語群に含まれているか判定する．
ステップ 9　1 つでも含まれていたとき，その時点で競合と判別する．

競合と判別されなかった場合，次の操作を実行する．

ステップ 10　これから実行しようとしているパッケージの FST スクリプトから取り出された単語を 1 つずつ，すでに実行されているパッケージの FST スクリプトとメイン FST スクリプトから取り出された単語群に含まれているか判定する．
ステップ 11　1 つでも含まれていたとき，その時点で競合と判別する．

競合と判別されたときは，そのことをユーザに通知することで競合を回避する．

4. プロトタイプシステム

プロトタイプシステムはコンテンツ管理サーバと Android 携帯端末からなっている．コンテンツ管理サーバには Apache と Tomcat を組み合わせたものを使用している．サーバ側で実装したプログラムには Java[5] を用いた．Android 端末側で実装したプログラムも Java を用いた．

4.1 コンテンツ管理サーバの実装

コンテンツ管理サーバの実装について述べる．コンテンツ管理サーバで実装された機能はパッケージアップロード機能，パッケージ検索・ダウンロード機能の一部である．

サーバ側の機能は Web サービス化されており，Servlet を用いて実装されている．JSP も利用している．JSP は HTML に埋め込むように記述できる Servlet であるため，画面描写部分に利用している．アップロードフォームの画面表示を図 9 に示す．入力フォーム上の各欄に音声対話コンテンツのメタデータを入力する．アップロードするパッケージは「ファイルを選択」の部分から選択できる．

ダウンロードフォームの画面表示を図 10，図 11 に示す．図 10 はパッケージの一覧を表示しているページである．リスト状となっており，上段にパッケージのタイトルが，下段にパッケージに含まれるアイコン画像が表示される．各パッケージには HTML のリンクが貼られており，選択すると図 11 に示すような各パッケージの詳細ページへと遷移する．

プロトタイプシステムでは or 検索を実装した．パッケージの検索のアルゴリズムを以下に示す．

ステップ 1　検索タグを配列に格納．
ステップ 2　パッケージごとに，メタデータを保存した XML ファイルに記述されたタグをすべて取り出す．
ステップ 3　取り出されたパッケージごとのタグ群に 1 つ

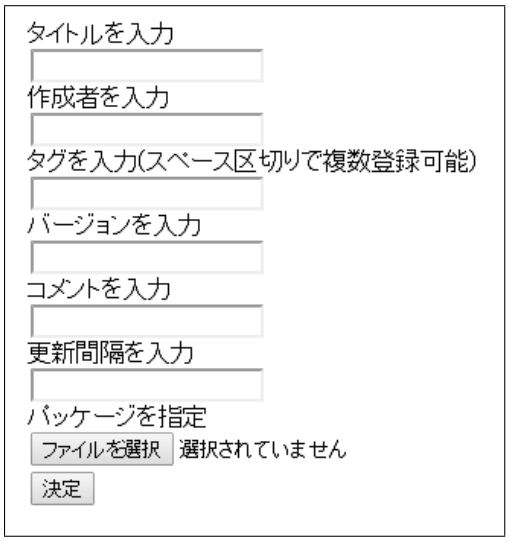

図 9　アップロードフォームの入力画面．音声対話コンテンツのメタデータを入力し，アップロードするパッケージ化された音声対話コンテンツを選択する．

図 10　音声対話コンテンツ配信 Web サービスの画面

図 11　パッケージの詳細を表示する音声対話コンテンツ配信 Web サービスの画面

でも検索タグを含まれていれば，そのパッケージは検索にヒットしたと判定する．

ステップ 2 と 3 をサーバにアップロードされた全てのパッケージに対して行う．

4.2　Android 端末側のインタフェースの実装

Android 端末側のインタフェースの実装について述べる．Android 端末に実装された機能はパッケージ検索・ダウンロード機能の一部，パッケージの選択機能，複数パッケージ間の認識キーワード競合判別機能である．メニューボタンを選択すると図 12 の画面が表示され，そこから実装した各機能を利用できる．機能は以下の通り

Download Apps ダウンロードアプリが起動し，パッケージ検索とダウンロードができる．

Select Apps パッケージを選択し実行できる．

Delete Apps パッケージを削除できる

Settings 複数パッケージ間のキーワード認識競合判別機能の ON/OFF ができる．

Select Apps を選択した画面を図 13 に示す．

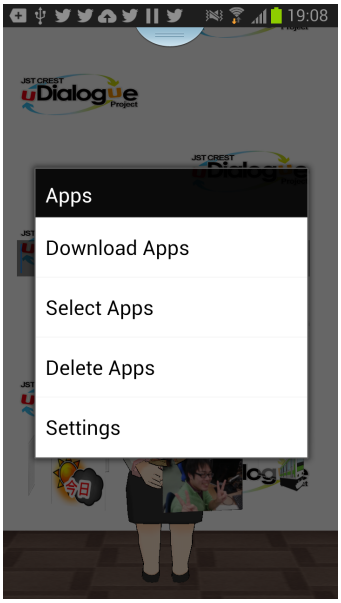

図 12　端末の各機能を利用するメニュー画面

画面はリスト状となっており，画面左側に各パッケージのアイコン画像が，中央上に各音声対話コンテンツのタイトル，中央下に各音声対話コンテンツの説明が表示される．そして，画面右側にはスイッチがあり，パッケージ実行の切り替えを簡単に行うことができる．また複数パッケージを実行しようとした時にキーワード競合が発生する可能性があると判別されたときは図 14 に示す画面が表示され，ユーザに競合の可能性を知らせる

パッケージの実行状態によるパッケージ選択画面の違いを図 15，図 16 に示し，3D モデル描写の違いを図 17，図 18 に示すパッケージに含まれるアイコン画像が 3D モデルのテクスチャ画像として適応され，3D モデルの描写が変化していることがわかる．

図 13　実行するパッケージを選択する画面

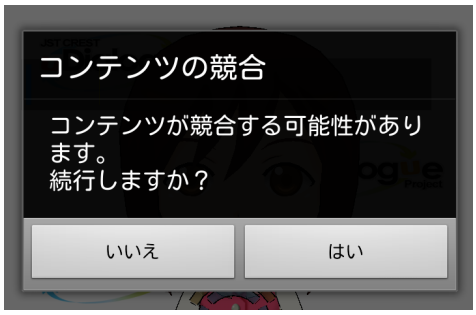

図 14　競合をユーザに知らせる画面

図 15　全くパッケージを
実行していない状態の
パッケージ選択画面

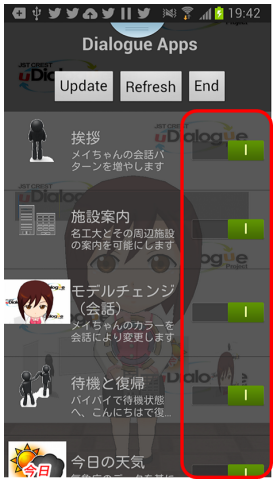

図 16　５つのパッケージを
実行した状態の
パッケージ選択画面

5.　実験と考察

複数パッケージを並列に実行した時に，キーワード競合判別機能がどの程度競合と判別するのか調べる．またその時本当に競合がするのか実際に動作させて調べる．これにより，実際どの程度のキーワード競合が発生するのか，またシステムが正しく競合判別を行えるのかを確かめる．実験で用いた FST スクリプトは，実際にデジタルサイネージシステムとして利用されている正門メイちゃん [6] 上で実行されている FST スクリプト相当のものである．この FST スクリプトは状態数が 2729，認識キーワード数が 1009 という規模が大きいものであり，「挨拶」，「施設案内」，「占い」といった機能のまとまりごとに 10 個に分割し，それぞれ提案手法でパッケージ化した．作成した 10 個のパッケージを 2 個ずつの組み合わせで同時実行し，キーワード競合判別機能により，どの組み合わせが競合と判別されるか実験した．実験の試行回数は ${}_{10}C_2 = 45$ 回．競合と判別された組み合わせについて実際に MMDAgent 上で同時実行することで，競合が発生するのか，どのような競合が発生するのか確かめた．

図 17　全くパッケージを
実行していない状態の
3D モデル描写

図 18　５つのパッケージを
実行した状態の
3D モデル描写

5.1　実験結果と考察

キーワード認識競合判別機能によって競合と判別された組み合わせは 4 通りであり，今実験におけるキーワード競合の発生確率は

$$\frac{4}{45} * 100 \simeq 8.9\%$$

であった．キーワード認識競合が発生すると判別されたパッケージの組み合わせ全てにおいて，実際に競合が発生し，音声の再生が不完全になる等が確認できた．これにより，キーワード競合判別機能は正しく動作していると確認できた．競合が発生した理由としては，分割する前の元の FST スクリプトで複数個所に同じ認識キーワードがあったことが挙げられる．MMDAgent は FST スクリプトの先に記述された部分のみを実行し状態遷移が一意に決まるため，認識キーワードの重複による競合は発生しないことを確認した．分割し同時に実行するときのみ，重複した認識キー

ワードによって同時に複数の状態遷移が発生してしまうため，認識キーワードによる競合発生は複数の FST スクリプトを同時実行する際に特有の問題であると考えられる．

6. 関連研究

データのパッケージ化のための新しい基準　多くのアプリケーションでは，コンテンツに様々な追加リソースが結合されている．文献 [7] では，アプリケーションデータを関連リソースと共に保存するための構造化された方法を，標準の ZIP ファイルを使用して定義している．この規格は Open Packaging Conventions (OPC) と呼ばれるオープン標準であり，様々なコンテンツとリソースを一つの ZIP ファイルにまとめるという点で本研究における音声対話コンテンツのパッケージ化と共通している．さらに OPC においてはパッケージ内やパッケージ間で関連付けが行われ，パッケージの構造に関する知識がなくても，コンテンツとリソースの関連付けを判断することができる．

コンテンツ作成の循環系を軸とした音声技術基盤の構築を目指して　文献 [8] では，ユーザによる音声対話コンテンツ生成という概念を導入し，それが実際に機能するための仕組みや条件を実証的に探究する試みについて紹介している．ユーザ生成型の音声対話コンテンツを普及させる仕組みを考案しているという点で本研究と共通している．文献 [8] ではユーザによる音声対話コンテンツ生成環境の構築にも触れている．

7. おわりに

本稿では，スマートフォンに搭載した音声インタラクションシステム構築ツールキット MMDAgent のためのパッケージ化による音声対話コンテンツ配信の仕組みを提案した．提案システムは機能単位でパッケージ化された音声対話コンテンツを配信することができ，ダウンロードしたパッケージを任意に複数並列に利用することができる．また，パッケージに音声対話コンテンツのメタデータを持たせることで，目的の音声対話コンテンツを素早く検索して見つけることができる．さらに，複数のパッケージ化された音声対話コンテンツを並列に実行したときに発生する競合を判別する機能も持つ．また，提案した実現法をもとに，Android 端末とコンテンツ管理サーバからなるプロトタイプシステムを実装した．さらに，プロトタイプシステムを用いて評価実験を行った．複数パッケージ並列実行時の競合判別機能が正しく機能しているかという実験を行い，システムが競合発生と判別した全ての場合で実際に競合が発生することが確認できた．

今後の課題としては，パッケージへのさらなる機能追加による実用性と利便性の向上が挙げられる．また，実際に多数のユーザ生成型音声対話コンテンツを広く配信できるようにシステムを改良し，評価実験を行っていくことも挙げられる．

謝辞 本研究は JSPS 科研費 25700009，及び，科学技術振興機構 CREST の助成を受けたものです．

参考文献

[1] Akinobu Lee, Keiichiro Oura, Keiichi Tokuda, MMDAgent - A fully open-source toolkit for voice interaction systems, Proceedings of the ICASSP 2013, pp. 8382-8385, 2013.5.

[2] Lee Akinobu, and Tatsuya Kawahara, Recent Development of Open-Source Speech Recognition Engine Julius, Proceedings of the APSIPA ASC, pp. 131137, 2009.

[3] 大浦 圭一郎, 酒向 慎司, 徳田 恵一，日本語テキスト音声合成システム Open JTalk, 日本音響学会春季講論集, Vol. 1, No.2-7-6, pp. 343344, 2010.

[4] 山本 大介，大浦 圭一郎，西村 良太，打矢 隆弘，内匠 逸，李 晃伸，徳田 恵一, スマートフォン単体で動作する音声対話 3D エージェント「スマートメイ ちゃん」の開発, インタラクション 2013, IPSJ Symposium Series Vol. 2013, No. 1, pp. 675-680, 東京，2013.

[5] Java, http://docs.oracle.com/javase/jp/7 (2014.2.2 参照)

[6] 大浦 圭一郎，山本 大介，内匠 逸，李 晃伸，徳田 恵一, キャンパスの公共空間におけるユーザ参加型双方向音声案内デジタルサイネージシステム，人工知能学会誌，Vol.28, No.1, pp.60-67, 2013.1.1.

[7] データのパッケージ化のための新しい標準，http://msdn.microsoft.com/ja-jp/magazine/cc163372.asp (2014.2.2 参照)

[8] 徳田　恵一，コンテンツ生成の循環系を軸とした音声技術基盤の構築を目指して，電子情報通信学会技術研究報告. SP, 音声 111(365)，pp. 153-157，2011