簡単化された実行履歴に基づく 音声対話コンテンツ編集システム

山口 大介† 堤 修平† 山本 大介† 高橋 直久†

† 名古屋工業大学大学院工学研究科情報工学専攻 〒 466-8555 愛知県名古屋市昭和区御器所町 E-mail: †yamaguchi@moss.elcom.nitech.ac.jp, ††{tsutsumi.shuhei,yamamoto.daisuke,naohisa}@nitech.ac.jp

あらまし 我々の研究室では,音声対話システム構築ツールキット MMDAgent を Android 向けに移植した Android 版 MMDAgent の研究を行なっている. Android 版 MMDAgent は,FST 形式で記述された対話スクリプトを読み込ませることで対話内容を自由にカスタマイズすることができる.FST ファイルはパソコン上のテキストエディタなどによって編集することができる.しかし,Android 版 MMDAgent の FST ファイルは Android 端末をパソコンに接続しなければ,カスタマイズすることが煩わしいと感じる場合が多く,記述された対話内容の量が多いほど,後から編集したいと感じた箇所を探すことや,適切な編集をすることが困難となってくる.Android 端末上で使用するアプリの簡単なカスタマイズは,使用する端末上で完結する方が好ましいと考えられるので,本稿では MMDAgent の対話の元となっている FST ファイルを Android 端末単体でカスタマイズすることのできるシステムを提案し,その提案システムに基づいて実装したプロトタイプシステムについて述べる.さらに,プロトタイプシステムを用いた評価実験から,提案システムの有用性を評価する.

キーワード 音声対話システム,マルチメディア,編集インタフェース

1. はじめに

近年, Apple の Siri [1] や, NTT ドコモのしゃべってコン シェル [2] など,スマートフォン向けの音声対話システムが普 及しつつある.これらの音声対話システムでは,ユーザは仮想 の秘書エージェントとの音声対話を通じて,経路案内や天気予 報などの情報を得ることができると同時に,簡単な雑談なども 行なうことができる.我々の研究室では音声対話システム構築 ツールキット MMDAgent [3] を Android 端末向けに移植した Android 版 MMDAgent [4] に関する研究を行なっている.以 下の図 1 に Android 版 MMDAgent の画面を示す.



図 1 Android 版 MMDAgent

MMDAgent は、名古屋工業大学正門前にも設置されている 双方向音声案内デジタルサイネージ [5] にも用いられており、上 記の音声対話システムとは異なった音声対話を実現している. MMDAgent はネットワークを介さない対話が可能となってい るため、より人間に近い、遅延の少ない対話が実現されている. MMDAgent はネットワークを介した対話を行なわない代わり に、FST ファイルと呼ばれる対話スクリプトで記述されたファ イルを読み込むことで、そのファイル内に記述されているキー ワードに関する対話だけを実行する.この FST ファイルを編 集することで MMDAgent の対話内容をカスタマイズすること ができ、FST ファイルの編集はパソコンのテキストエディタな どを使用することで編集ができる. Android 版 MMDAgent は パソコンとのデータ転送を必要としたカスタマイズがほとんど となっており, Android 端末単体で完結するものはごく少数と なっている. また, FST ファイルでは特定のコマンドを用いた 記述をする必要がある.そのため、専門知識が無い者では FST ファイルの編集が困難であると感じる場合がある.

2. MMDAgent

2.1 MMDAgent の特徴

MMDAgent は音声対話のための高度な機能を備えたツール キットであり,音声認識,音声合成,3Dモデルの描画や物理 演算などを統合したシステムである.音声認識エンジンとして Julius [6]を,音声合成エンジンとして OpenJTalkを [7],3D モデルとして MikuMikuDance 形式 [8]を,物理演算エンジン として Bullet Physics [9]を採用している.また,MMDAgent は音声対話を記述するためのスクリプト言語に基づいて音声対 話の処理を実行する.スクリプト言語は後述する FST(Finite State Transducer)形式に基づいており,音声認識による特定 のキーワードの認識や,センサーによるユーザの検出などをト リガーとして,次の状態に遷移する有限オートマトン形式を記 述可能である.

2.2 FST 形式の特徴

MMDAgent は FST 形式で記述されたテキストファイル (以 降 FST ファイルと呼ぶ)を読み込むことで,音声対話を実現し ている.FST 形式とは,状態番号,次状態番号,条件,出力の 順に並ぶ対話スクリプトである.

以下に, MMDAgent が「こんにちは」と認識した際に, お辞 儀をしながら「こんにちは」と返答するように記述された部分 を含む FST ファイルの一部と, その「こんにちは」の処理に おける状態遷移図をそれぞれ図 2,3 に示す.

1	10	RECOG_EVENT_STOP こんにちは	<eps></eps>
1	10	RECOG_EVENT_SYOP こんにちわ	<eps></eps>
10	11	<eps></eps>	MOTION_ADD mei action Motion ¥ mei_greeting ¥ mei_greeting.vmd PART ONCE
11	12	<eps></eps>	SYNTH_START mei mei_voice_normal こんにちは
12	1	SYNTH_EVENT_STOP mei	<eps></eps>

図 2 FST ファイル



図 3 状態遷移図

図2中の番号が"1"のような状態を初期状態(又は待機状態) と呼ぶ. MMDAgent が何もしていない時は,基本的にこの初 期状態にある.そして, MMDAgent が初期状態にあるときに 音声認識によって「こんにちは」と認識すると、次の状態、例 えば図2中の番号"10"の状態へ遷移する.番号"10"の状態は、 図2中の3行目の状態となっており、出力は「お辞儀をする」 となっているので, MMDAgent がお辞儀をする. また, 図2 中の3行目の条件を見ると<eps>となっている.これは、「条件 無しで次の状態へ遷移する」という条件なので, すぐに図2中 の番号"11"の状態へと遷移する.番号"11"の状態の条件は、番 号"10"の状態と同じなので、番号"12"の状態へと遷移し、出力 は,「こんにちはと返答する」となっている. 図2中の番号"12" の状態は、条件が「音声合成が終わった時」となっている.こ れは、「こんにちはと返答する」などの音声合成の出力が終了す ると遷移するという条件である. 出力はないので, 音声合成が 終わると初期状態に遷移するという遷移である. これらの状態 遷移を状態遷移図で表したのが、図3である.

2.3 FST ファイルの編集における問題点

FST ファイルを編集する際によく起こる問題として,状態番号の意図しない重複が挙げられる.同一FST ファイル上で状態番号が重複してしまうと,FST ファイル中の,同じ状態番号の上に書かれた行だけが実行され,下に書かれた行は実行されないなどのFST ファイルを記述した者の意図しない動作が起

こることがある.他にも,誤字などの記述ミスが起こることも 挙げられる.FST ファイル内にコマンドの誤字があると,その 行は無効となってしまうため,書いたつもりの処理がされない 等の問題が生じる.

Android 版 MMDAgent の FST ファイルを編集する場合, Android 端末上で編集するには画面サイズの小ささや, エディ タがパソコンに比べて不便であるなどの理由により困難と感 じる場合がある.そのため, Android 版 MMDAgent の FST ファイルを編集する場合, パソコン上で FST ファイルを編集 してから,使用する Android 端末に保存するという方法が主で ある.

また,図2で示した通り,FSTファイルでは条件や出力をコ マンドを用いた記述を行う.そのため,FSTファイルに関して 知識の無いユーザが編集を行おうとすると,編集したい個所が 見つけられない,どのようなコマンドを記述すれば意図した出 力が得られるのか分からないといった問題が生じてしまうため, 編集が困難であると感じる.

3. 課題と解決方法

提案手法では,従来の FST 書式で発生していた編集におけ る問題点を解決する.まず,従来システムでの FST ファイル の編集手順を以下に示す.

 (1) FST ファイルをテキストエディタで開き,編集したい と感じた部分を探す

(2) 元のコマンドを消去し,変更したいコマンドを記述 する

(3) FST ファイルを保存し、再度 MMDAgent を起動して 動作を確認する

以上の手順から,従来の FST 書式での編集における問題点は 以下のものが挙げられる.

• FST ファイルに膨大なシナリオが記述されていると編集 したい個所を探すのが困難

• FST 書式では,決まったコマンドのみが有効となるため 正しくコマンドを記述しないと思い通りの出力がされない

これらの問題点を解決するために,FST ファイルを編集する ための機能を二つ MMDAgent に追加した.

まず、一つ目の問題点を解決するために、MMDAgent に実 行履歴を表示する機能を追加した.本論文で用いる実行履歴は、 MMDAgent が実際に認識したキーワードや、出力した動作な どが保存されたテキストファイルである.

そもそも,FST ファイルを編集したいと感じるときとは, MMDAgent を実際に動かしているときである.この考えを基 に,実行履歴を Android 画面上に表示する機能を追加した.こ の機能により,MMDAgent が実行したコマンドが逐次 Android 画面上に表示されるため,編集したいと感じたときに直ちに編 集個所を見つけることができるようになる.

また,表示する実行履歴を FST ファイル内で用いるコマン ドではなく,コマンドを分かりやすく日本語を用いて表示する ことで,深い専門知識が無い者でも変更したい個所を見つける ことができるようにした. 二つ目の問題点を解決するために, FST ファイルの編集を記 述式ではなく選択式に変更した.

本システムでは、変更したいコマンドが Android 画面上に 一覧で表示され、その中から編集したい内容を選択するだけで FST ファイルが編集されるため、コマンドの勘違いやタイプミ ス等が起きない.よって、より快適に FST ファイルを編集で きるようになる.

また,これらの機能はすべて Android 版 MMDAgent に追加された機能であるため,確認のために再度 MMDAgent を起動する手間が省けるため動作の確認も容易になる.

4. プロトタイブシステム

提案システムは Java [10] 及び C++を用いて開発した. 開発 環境は AndroidStudio [11] を採用した.

4.1 実行履歴表示機能

実行履歴表示機能は、ユーザが表示したいときに画面上に実 行履歴を表示する機能である.実際の画面の一部を以下の図4 に示す.



図4 実行履歴

図4は、MMDAgentのコマンドである"MOTION_ADD" や"RECOG_EVENT_STOP"等を日本語化して表示している. 実行履歴の一行はそれぞれが独立しているため、変更したいと 思った一行を選択するだけで変更箇所を指定できる.従来の編 集手法では、エディタを用いて FST ファイルを開き編集した い個所を FST ファイル全体を見て探す必要があるため提案シ ステムを用いた編集手法の方が変更が容易になる.

また,実行履歴を用いて変更箇所を指定すると,変更内容を 選択する画面に遷移する.例えば,「こんにちは」と認識したと きに「こんにちは」と返答しながらお辞儀をさせたいと感じた ときは,実行履歴中の『"ANGER"の動作を実行』を選択する. 選択すると MMDAgent が実行することのできるモーションの 一覧が実行履歴と同様に一覧で表示される.以下の図5にモー ションの一覧を表示した画面を示す.

表示された一覧の中からお辞儀を選択することで変更後の内 容が決定される.

変更箇所と変更内容が決定すると後述する FST ファイル編 集機能によって実際に Android 端末内に保存されている FST ファイルに自動的に反映される.

4.2 FST ファイル編集機能

FST ファイル編集機能では,選択式で決定した編集内容を 実際に反映させるための機能である.実行履歴表示機能によっ て,FST ファイル内の編集したい行,編集後の内容が決定して いる.FST ファイルは一行が四つの要素から成り立っている



図 5 モーションの一覧

ため,それらを配列に代入し,文字列として処理することで編 集内容を反映させる.以下に,「こんにちは」と認識したときに 「こんにちは」と返答しながら「お辞儀」をするように変更す る編集の例を図6に示す.



図 6 FST ファイルの編集手順

図6で示すように、編集したい一行を四つに分割する.その後、元々記述されている内容と編集後の内容を照らし合わせ、 一致していない部分の編集内容を反映させ、再度保存するという流れになっている。一行を四つに分割した理由は、FSTファ イルを編集したいと感じた時、実際に編集する部分は、上手く 遷移しない状態番号を編集するか、入力か出力が気に入らない 時が多いと考えたためである.

4.3 実行履歴簡単化機能

FST ファイルでは、"MOTION_ADD"や"RECOG_EVEN-T_STOP"などコマンド毎に用途がはっきりと分かれている. そ のため,これらのコマンドを条件に合わせて簡単な日本語で表示 することで深い知識の無いユーザでも手軽にFST ファイルの編 集を行うことができるようにした.例えば、"MOTION_ADD" ならば「そのコマンドの後ろに記述されているモーションファ イルの動作を行う」というコマンドであり、その後ろのファイ ル名がどのような動作を行うかを決定する. なので、その後ろ のファイル名を解析し、「お辞儀のモーションをする」というよ うに実行履歴を表示する際に簡単化を行う.

以下の図7,8にFSTファイルのコマンドをそのまま表示した実行履歴の場合のMMDAgentの画面と、同じ内容の実行履歴を簡単化して表示した場合の画面を示す.



図 7 FST ファイルの編集手順



図 8 FST ファイルの編集手順

5. 評価実験

5.1 目 的

実際にプロトタイブシステムを用いて FST ファイルを編集 してもらい,編集にかかった時間を測定する.同じ編集内容を 従来の編集手順で編集してもらい,どのくらい編集にかかる時 間が短縮されるかを比較することで提案システムを用いた FST ファイルの編集がどれだけ楽になったかを評価する.また,ア ンケートを用いてユーザが実行履歴を使った編集が楽と感じる か等を評価する.

5.2 実験方法

Android 端末内に保存されている FST ファイルにおいて, MMDAgent が反応する特定のキーワードを

(1) パソコンを使った従来手法

(2) パソコンを使って実行履歴編集を模した画面を見なが ら編集してもらう中間手法

(3) 簡単化された実行履歴を用いて Android 上で編集し てもらう提案手法

の3つの編集を行ってもらう.評価実験を行う際に用いる FSTファイルは,評価実験用に用意したものを使用する.この FSTファイルは行数約380行,状態数は約200である.編集 時間を計算しやすくするために,それぞれの場合において,三 つのキーワードで編集を行ってもらう.三つのキーワード全て の編集が終わるまでの時間を測定し,平均を取って3つの編集 手法で比較する.

5.3 実験結果と考察

それぞれの編集手法でかかった時間の計測結果を以下の表1 に示す.また,計測結果およびアンケート結果をグラフ化した ものを以下の図に示す.

公Ⅰ 棚米にかかうた时间の可倒加っ	表	1 編	[集にか]	かった	時間の	計測結果
-------------------	---	-----	-------	-----	-----	------

被験者番号	従来手法 [秒]	中間手法 [秒]	提案手法 [秒]
1	189	160	93
2	202	167	108
3	214	177	98
4	221	172	89
5	197	132	92
6	171	144	79
平均	199.0	158.7	93.2



図 9 一つ当たりのキーワードの編集にかかる時間



図 10 編集にかかった総時間



図 11 アンケート結果

実験結果から従来手法によって FST ファイル内の1行を3回 編集するのにかかる時間はおよそ 199 秒であるのに対し、実行 履歴を用いてパソコンで編集するのにかかる時間はおよそ 158 秒となっており、およそ2割の編集時間の短縮が実現されてい る. この結果より、実行履歴を用いた編集では編集時間の短縮 が可能であると考えられる. 今回パソコンでの編集において用 いた実行履歴は、FST ファイル内の編集したい個所が FST 内 の何行目にあるかだけを表示していた. そのため, Android 端 末上の実行履歴のように、クリックしただけで FST ファイル の編集したい個所に直接リンク等がないにも関わらずこれだけ の編集時間の短縮が行えたため、パソコン上で実行履歴を実装 すればより編集時間の削減が可能であると考えられる.よって, 実行履歴を用いることは編集時間を短縮する上で有用であると 言える.また,従来手法と提案手法を比べてみると,提案手法 を用いた編集ではおよそ5割の編集時間が短縮されている.こ れは、実行履歴を用いた編集の方が従来手法と比べて、編集の 手間が極めて減少しているからであると言える. 実行履歴を用 いていることで、FST ファイル内の編集したい個所を探す時間 が短縮されているだけではなく, FST ファイルをキーボードに よる文字の打ち込みが殆ど無く、タップ操作だけで FST ファ イルを編集できるためであると考えられる.よって, FST ファ イルの編集を Android 端末上で行う方が,ユーザの負担が減っ ていると言える.また以下の表2に編集個所を見つけるまでに かかった時間の計測結果の平均を示す.

表 2 編集個所を探すまでにかかった時間の計測結果

従来手法 [秒]	中間手法 [秒]	提案手法 [⁵	秒]
92.7	72.8	24.7	

表2から分かる通り,従来手法に比べて提案手法では,編集 個所を探すまでにかかる時間がおよそ7割削減されている.こ の結果からも, Android 端末上で FST ファイルを編集する際 のユーザの負担が減っていると言える.

図 11 より Android 画面上の実行履歴は読みやすいと感じま したかという質問に対して高い評価が得られた。また、実行履 歴を用いた2つの編集手法では Android 端末上での編集の方 が編集しやすいと回答した者が多かった.これらの結果から, 提案手法が FST ファイルの編集をより楽にしていると言える. パソコンを用いた2つの編集手法を比べてみても,実行履歴 があった方が編集しやすいと回答した者が多かったことから, FST ファイルの編集において実行履歴は有用であることが言え る.これらのアンケート結果から比べると、パソコンでの編集 においてデータの転送が面倒と回答した者は少なかった.この ことから、ユーザは FST ファイルの転送が面倒なのではなく、 FST ファイルの編集自体が面倒であると感じることが多いと 考えられる. そのため, Android 端末上で FST が編集できな かったとしても、実行履歴を用いれば FST ファイルの編集に おけるユーザの負担を軽減することができるのではないかと考 えられる.よって、今後はパソコン上で Android 端末に実装し た実行履歴を用いた編集手法を実現し、提案手法と比較する必 要があると考えられる.

6. おわりに

本論文では、Android 端末上の FST ファイルを編集する際 に感じていた、編集のしづらさを原因毎に分析し、それらの解 決策を考えるとともに実際に新しい編集手法を提案、実装し た.また、実装したプロトタイブシステムにおいてその有用性 を評価実験により評価した.評価実験では、編集時間は半分以 下に短縮され、アンケートでは編集しやすいと回答した者が多 かった.ただし、今回の評価実験では、Android 端末上で全て の編集を行う方が編集しやすいかどうかについて疑問が残って しまった.今後の課題として、この問題を解決するとともに、 その結果次第では、実行履歴や、選択式といったもの以外に新 たな編集の簡単化を図る必要があると考えられる.

7. 関連研究

7.1 タブレット端末のための複合状態を用いた 音声対話コンテンツ編集手法

タブレット端末のための複合状態を用いた音声対話コンテン ツ編集手法[12]では、FSTファイルの編集手法として、タブ レットのタップ操作を用いて手軽に編集できるシステムを提案 している.このシステムでは、よく使う一連の処理をテンプ レートとして用意し、そのテンプレートを用いることで手軽に FSTファイルの編集が可能となっている.しかし、既に書かれ ている処理に対して編集をしようとすると、編集したいと感じ た部分を探すのに手間がかかってしまう場合がある.

7.2 タブレット端末のための複合状態を用いた

音声対話コンテンツ編集手法

音声対話コンテンツにおける状態遷移ログの可視化と簡単化 手法[13]では、状態遷移ログと内部メッセージログを対応付け て状態遷移図として可視化することで、状態遷移ログの情報を 直感的に得ることができるようになっている.さらに、状態遷移 ログから得られた情報を見やすくするために、見かけ上の状態 数を削減し簡単化している.しかし,Android 版 MMDAgent では実装されていないためパソコンを使った MMDAgent の編 集が前提となっている.

7.3 Android 端末のための実行履歴を用いた音声対話コ ンテンツ編集システム

Android 端末のための実行履歴を用いた音声対話コンテンツ 編集システム [14] では、実行履歴を用いて FST ファイルを編 集するシステムについて提案している. このシステムでは、評 価実験の結果から FST ファイルのコマンドをそのまま表示し た実行履歴が見づらいという結果が得られている. そのため、 実行履歴を色分けするなどして見やすくできるのではないかと いう提案がされている.

謝

辞

本研究は JSPS 科研費 26330136, 25700009, 科学技術振興 機構 CREST, および, 総務省 SCOPE の助成を受けたもの です.

文 献

- [1] Apple.,Siri,
- 入手先 http://www.apple.com/ios/siri/(参照 2016-01-26)
 [2] NTT ドコモ,しゃべってコンシェル,
- 入手先 http://www.nttdocomo.co.jp/service/informati-on/ shabette concier/ (参照 2016-01-26)
- [3] 李晃伸,大浦圭一郎,徳田恵一,魅力ある音声インタラクション システムを構築するためのオープンソースツールキット,電子情 報通信学会技術研究報告. NLC,言語理解とコミュニケーション Vol.111,No.364,159-164,2011.
- [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
- [5] Akinobu Lee, Keiichiro Oura, Keiichi Tokuda, MMDAgent fully open-source toolkit for voice interaction systems Proceedings of the ICASSP 2013, pp. 8382-8385, 2013.
- [6] Lee, A. and Kawahara, T. Recent Development of Open-SourceSpeech Recognition Engine Julius, APSIPA, pp. 131-137, 2009
- [7] 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 OpenJTalk, 日本音響学会春季講論集, Vol.1, No.2-7-6, pp.343-344, 2010.
- [8] MikuMikuDance, 入手先 http://www.nttdocomo.co.jp/service/information/shabette_concier/(参照 2016-01-26)
- [9] BulletPhysics, 入手先 http://www.nttdocomo.co.jp/service/information/shabette_concier/(参照 2016-01-26)
- [10] Java,
- 入手先 http://www.oracle.com/ (参照 2016-02-04) [11] AndroidStudio,
- 入手先 https://developer.android.com/studio/index.html?hl =ja(参照 2017-1-10)
- [12] 若林敬太郎,山本大介,高橋直久,タブレット端末のための複合状態を用いた音声対話コンテンツ編集手法,マルチメディア,分散協調とモバイルシンポジウム 2014 論文集 2014,pp.781-788,2014.
- [13] 田中佑太朗,山本大介,高橋直久,音声対話コンテンツにおける状態遷移ログの可視化と簡単化手法,マルチメディア,分散協調と モバイルシンポジウム 2015 論文集 2015,pp.1263-1270,2015.
- [14] 山口大介, 堤修平, 山本大介, 高橋直久, Android 端末のための実 行履歴を用いた音声対話コンテンツ編集システム, マルチメディ ア, 分散協調とモバイルシンポジウム 2016 論文集 2016, pp. 961-968, 2016.