Web ブラウザ上でマルチモーダル対話を可能にする JavaScript ライブラリ MMI.js の提案

MMI.js: A Proposal of JavaScript library enabling multimodal interaction on Web browsers

菊地　泰己 [1*]　桂田　浩一 [1]　入部　百合絵 [1]　新田　恒雄 [1,2]

Taiki Kikuchi[1], Kouichi Katsurada[1], Yurie Iribe[1], and Tsuneo Nitta[1,2]

[1] 豊橋技術科学大学
[1]Toyohashi University of Technology

[2] 早稲田大学
[2] Waseda University

Abstract: This paper proposes a JavaScript library called “MMI.js” which enables us to use multiple modalities on Web browsers. This library supports sequential multimodal inputs/outputs, simultaneous multimodal inputs/outputs, alternative multimodal inputs/outputs, synchronization of multimodal inputs/outputs and gestures given by the dialogue agents. To show usefulness of this library, we embedded multimodal interaction into an English pronunciation training application for Japanese students. Through the development of this application, we confirmed the library makes it easy to describe combination of multiple inputs/outputs appearing in complicated interaction.

1　はじめに

近年，スマートフォンの普及により，Siri[1]やしゃべってコンシェル[2] 等の対話エージェントの利用が活発になってきている．ユーザはこれらの対話エージェントを通して様々なサービスを利用することができる．しかし一方で，こうした対話エージェントは特定の端末のみにおいて提供されているため，多様な端末上でのアプリケーションの開発という点では限界がある．一般の技術者が対話エージェントを利用したアプリケーションを開発できる環境が整えば，より多くのサービスが提供できると考えられる．

筆者の研究グループではこれまでに，アプリケーションの開発者が擬人化エージェントを利用した自由に対話をデザインできる Web ベース MMI (Multi-Modal Interaction)システム[3]を開発してきた．このシステムは一般的な Web ブラウザ上で動作可能なため，開発者はユーザの利用環境を考慮することなくアプリケーションを開発することができる．しかし，このシステムを利用したアプリケーションの開発には，XISL[4]という対話シナリオ記述言語を習得する必要があった．新たな言語の習得には労力を要するため，開発者にとって慣れ親しんだ開発環境を提供する必要があった．

そこで本論文では，Web コンテンツの開発言語として一般的に用いられている JavaScript を用いて MMI システムを開発するためのライブラリ MMI.js を提案する．JavaScript は Web コンテンツを開発する際の最も基礎となる言語であることから，Web 技術者は新たな言語の習得なしに MMI アプリケーションを開発することが可能になる．提案するライブラリでは，まず筆者らが提案してきた XISL でサポートしている，逐次的，同時的，択一的なマルチモーダル入出力機能を提供する．また，マルチメディアコンテンツの同期制御を行う言語 SMIL[5]でサポートされている，入出力のタイミング制御機能を提供する．さらに，エージェントによるプレゼンテーションを記述する言語 MPML[6]や，テレビ番組を記述する言語 TVML[7]で整備されている，対話エージェントによる様々なジェスチャーを実行するための API を提供する．以上の機能により，複雑な対話を容易に記述することが可能になった．

以下では，まず関連する技術とその特徴を説明した後に，本ライブラリに要求される機能とその仕様について述べる．その後ライブラリを用いて作成したサンプルアプリケーションを紹介する．

* 連絡先：豊橋技術科学大学　情報・知能工学専攻
〒441-8580 愛知県豊橋市天伯町雲雀ヶ丘 1-1
E-mail:kikuchi@vox.cs.tut.ac.jp

表 1　関連技術がサポートする機能の一覧

機能	XISL	SMIL	MPML	TVML	Microsoft Agent	Google Speech API	w3voice
マルチモーダル入力	○	×	△	×	○	○	○
マルチモーダル出力	○	○	○	○	○	×	×
メディア同期機能	△	○	○	○	×	×	×
エージェントの動作	○	×	○	○	○	×	△
インストールが不要	○	×	×	×	×	○	○
JavaScript での記述	×	×	×	×	○	○	○

○：対応　△：一部制限あり　×：非対応

2　関連技術

2.1　提案手法に関連する記述言語

マルチモーダル対話を記述するための言語として，これまで筆者らは XISL[4]を提案してきた．XISL は XML ベースの言語であり，マルチモーダル対話で頻繁に現れる逐次的，同時的，択一的な入出力を記述するためのタグセットを備えている．また，対話シナリオの記述に必要な条件分岐や算術演算といった基本的な対話制御タグも提供している．

SMIL[5]は W3C で策定されたマルチメディアコンテンツの同期制御を行うための言語である．動画像や音声の同期制御など出力に特化した言語であり，開始終了時間のタイミングやオフセット，複数メディアの同期的な制御など，細かいチューニングを施したメディアの再生が可能である．

石塚らが提案した MPML[6]はエージェントを用いたプレゼンテーションを記述する言語である．エージェントが行う様々なジェスチャーを用意しており，効果的なプレゼンを実現している．

NHK が開発したテレビ番組を記述するための TVML[7]は，エージェントのジェスチャーの他，照明や小道具といった細かい演出もサポートしている．

以上に述べた言語は，それぞれの目的に対する十分な記述能力を備えており非常に有用であるが，それぞれが独立した言語であるため，コンテンツの記述には各言語の習得が必要である．

2.2　JavaScript を用いた従来技術

JavaScript を用いた対話制御用の API やライブラリも既にいくつか提案されている．Microsoft Agent[8]は，Web 上でエージェントのアニメーションを行うためのフレームワークである．数種類のキャラクターが提供されており，そのキャラクターに合ったジェスチャーのアニメーションが行われる．

Google Speech API[9]は，Chrome ブラウザに音声入力機能を付加するための API である．ユーザは特殊なソフトウェアやプラグイン等を使用することなく音声インタフェースを利用することが可能である．

w3voice.js は，西村らが開発した w3voice システム[10]の一部に含まれる，Web ページを音声入力対応にするためのライブラリである．サーバで音声認識エンジン Julius を動作させることで，Web ブラウザを通した DSR(Discrete Speech Recognition)を実現している．

このように，Web ブラウザ上でエージェントのジェスチャーや音声インタフェースを提供する技術が増えてきている．しかし，Microsoft Agent はソフトウェアのインストールが必要であり，Google Speech API と w3voice.js は音声のみの入力を対象にしているため，複数モダリティの同時入出力や同期制御はサポートされていない．

3　提案するライブラリ

3.1　要求される機能

マルチモーダル対話記述言語に要求される機能として，まず，マルチモーダル対話に頻繁に現れる逐次的，同時的，択一的な入出力を制御する機能が挙げられる．この機能により，複数の入力モダリティからのユーザの好みによる選択や，複数メディアの同時出力等，様々な対話のパターンがカバーできる．次に，入出力メディアの同期制御機能が挙げられる．例えば，ある入力と別の入力が同時であるかどうかの判断や，複数の出力の開始や終了のタイミングを制御する上で必要になると考えられる．また，ユーザに対して対話の意図を的確に分かりやすく伝え，なおかつ親しみ易いものにするために，様々なジェスチャーが可能な対話エージェントを用意する必要がある．さらに，誰もが簡単にシステムを利用できるよう，特殊なソフトウェア等のインストールが必要のない利用環境が望ましい．

これらの機能が，前節で述べた関連技術によってどの程度サポートされているかを表 1 に示す．表に示すように，関連技術でこれら全ての機能を備えて

いるものはない．本研究ではこれらの機能全てを満たすライブラリを提供するために，以下の要求仕様を満たすインストール不要の JavaScript 用ライブラリを提供する．

【要求仕様】

1．マルチモーダル入出力機能を備える．
2．マルチメディアコンテンツの同期制御が可能である．
3．エージェントの様々な動作アニメーションを記述できる．

3.2 マルチモーダル入出力機能

要求仕様の１．の機能を提供するために，XISL のマルチモーダル入出力機能を参考に，逐次的，同時的，択一的な入力を受け付ける関数 mmi.seqInput, mmi.parInput，mmi.altInput および逐次的，同時的な出力を行う関数 mmi.seqOutput，mmi.parOutput を用意した．これらの関数を使用する際には，入力では type（入力手段），event（入力に利用するイベント）と，match（入力を受け付ける HTML の要素）または grammar（音声入力に用いる文法ファイルのパス），および必要に応じたその他オプションを JSON 形式で記述する．表 2 に入出力関数の一覧を，表 3 と表 4 にそれぞれ入力，出力関数の引数として記述可能な項目の一部を示す．

3.3 マルチメディアコンテンツの同期制御

要求仕様の２．の機能を提供するために，SMIL のメディア同期機能を参考に，入出力のタイミング制御を行うための 17 種類の引数を用意した．入出力の開始，終了を制御する引数“begin”, “end”に時間（ミリ秒単位）またはブラウザ上で発生するイベントを，入出力の継続時間を制御する引数“dur”に継続時間を記述することで，指定した時間長の入力受付，出力を実行することができる．これらの機能は表 2 に示したマルチモーダル入出力関数の引数を通して利用できる．

3.4 エージェントの動作

要求仕様の３．の機能を提供するために，MPML, TVML を参考にして“お辞儀”や“指さし”といったエージェントの 62 種類の動作を用意した．また，対話エージェントのジェスチャーをより効果的にするために，従来の様な上半身のみのエージェントから，全身を使った動作が可能なオリジナルのキャラクターに変更した．この機能もメディアの同期制御機能と同様に，入出力関数の引数を通して利用できる．

表２　入出力関数一覧

分類	関数名	処理内容
マルチモーダル入力	mmi.seqInput	逐次的な入力を受付ける
	mmi.parInput	同時的な入力を受付ける
	mmi.altInput	択一的な入力を受付ける
マルチモーダル出力	mmi.seqOutput	逐次的な出力を行う
	mmi.parOutput	同時的な出力を行う

表３　入力関数の引数

項目	記述内容	概要
“type”	“click” “keydown” “speech” ：	クリック入力の受付 キーボード入力の受付 音声入力の受付 ：
“match” “grammar”	HTML ID 文字コード 文法ファイル	入力を受付ける要素 入力を受付ける文字 音声入力のための文法
“options” “begin” “dur” “end” ：	 時間 or イベント 時間 時間 or イベント ：	 入力開始のタイミング 入力受付継続時間 入力終了のタイミング ：

表４　出力関数の引数

項目	記述内容	概要
“type”	“agent” “audio” ：	エージェントによる出力 音声ファイルの出力 ：
“event”	“speech” “gesture” “play”, “stop” ：	音声を合成して出力 ジェスチャーを実行 ファイルの再生，停止 ：
“text” “gesture” “src” ：	出力する文章 ジェスチャー 音声ファイル ：	合成する音声の内容 実行するジェスチャー 出力するファイルのパス ：
“optins” “begin” “dur” “end ：	 時間 or イベント 時間 時間 or イベント ：	 出力開始のタイミング 出力の継続時間 出力終了のタイミング ：

3.5 対話の記述例

本ライブラリを使用した対話シナリオの入出力部分の記述例を表 5 に示す．この記述例の入力部分は，クリック入力または音声入力のどちらかを受け付ける択一的入力となっている．3.3 節で述べたように，引数“options”で定義される“begin”は，入力受付開始のタイミングを指定するもので，この例では，プログラムが読み込まれてから 1000 ミリ秒後に入力受付を開始するよう指定されている．また，同じく定義されている“dur”は，入力受付の継続時間を表し，入力受付が開始されてから 10 秒間入力がなければ

表 5　入出力の記述例

```
//択一的入力
mmi.altInput({
      “type” : “click”,
      “match” : “agent”,
      “options” :{   “begin” :1000,
                     “dur”   :10000   }
},{
      “type” : “speech”,
      “grammar” : “./grammar.txt”,
      “options” :{   “begin” :1000,
                     “dur”   :10000   }
});
//同時的入力
mmi.parOutput({
      “type” : “agent”,
      “event” : “speech”,
      “text” : “商品を選択してください”,
      “options” :{   “begin” :500   }
},{
      “type” : “agent”,
      “event” : “gesture”,
      “gesture” : “point”,
      “options” :{   “begin” :500   }
});
```

入力受付を終了するよう規定されている．出力部分はエージェントによる発話と“指さし”ジェスチャーを同時に実行する同時的出力となっている．引数“options”に“begin”が定義されているため，プログラムが読み込まれてから 0.5 秒後に出力が開始される．

4　サンプルアプリケーション

提案したライブラリの有用性を確認するために，本研究室で開発している Flash ベースの英語の発音訓練ソフト[11]にマルチモーダル対話機能を組み込んだ. 図 1 にスクリーンショットを示す. ユーザは，音声入力またはマウス入力によりアプリケーションを操作することができる．図 1 画面中央に表示されている対話エージェントは，ユーザの入力や英単語の発音の結果に合わせて，指定されたタイミングで発話やジェスチャーを行う．提案したライブラリを用いることで，従来は難しかったアプリケーションの動作結果に応じたエージェントの振る舞いや，タイミングを指定したメディアの同期出力を容易に組み込むことが可能であることを確認した．

5　まとめ

本稿では，Web ブラウザ上でマルチモーダル対話を可能にする JavaScript ライブラリ MMI.js を提案した．また，Flash ベースの発音訓練ソフトにライブラリを組み込むことにより，アプリケーションの動作の結果に応じたエージェントの振る舞いや，タイミングを指定したメディアの同期出力を容易に組み込むことが可能であることを確認した．このライブラリは，ユーザの入力，エージェントのジェスチャーを組み合わせた複雑な同期制御を，一つの関数のみで記述することを可能にする．エージェントとの違和感のない自然な対話を実現する上で同期制御は必須であることから，本ライブラリは対話アプリケーションを構築するための基盤となり得る．今後はスマートフォンへの対応に取り組むと共に，エージェントを可変にする等の拡張を検討したい．本ライブラリは Web 上での公開を予定している．

図 1　英語の発音訓練ソフトのスクリーンショット

参考文献

[1] http://www.apple.com/jp/ios/siri/

[2] http://www.nttdocomo.co.jp/service/information/shabette_concier/

[3] 工藤正志，桂田浩一，入部百合絵，新田恒雄: MMI6 階層モデルに準拠した Web ベース MMI システムの開発, FIT2009, pp. 345-346, (2009)

[4] 桂田浩一，中村有作，山田　真，山田博文，小林　聡，新田恒雄: MMI 記述言語 XISL の提案，情報処理学会論文誌, Vol. 44, No. 11, pp. 2681-2689, (2003)

[5] http://www.w3.org/AudioVideo/

[6] 筒井貴之，石塚　満，キャラクタエージェント制御機能を有するマルチモーダル・プレゼンテーション記述言語 MPML，情報処理学会論文誌, Vol. 41, No. 4, pp. 1124-1133, (2000)

[7] Hayashi, M., Ueda, H. and Kurihara, T.: TVML (TV program Making Language) - Automatic TV Program Generation from Text-based Script -, ACM Multimedia'97 State of the Art Demos, (1997)

[8] http://www.microsoft.com/products/msagent/main.aspx

[9] http://www.google.com/intl/ja/chrome/browser/

[10] 西村竜一，三宅純平，河原英紀，入野俊夫: 音声入力・認識機能を有する Web システム w3voice の開発と運用, IPSJ SIG Technical Report, Vol. 2007-SLP-68-3, pp. 13-18, (2007)

[11] 森　拓郎，入部百合絵，桂田浩一，新田恒雄: 発音訓練のための調音特徴に基づく IPA 母音図へのリアルタイム表示，IPSJ SIG Technical Report, Vol. 2011-SLP-89, No. 15, pp. 1-6, (2011)