Web ベース MMI システムを実現するための JavaScript ライブラリ MMI.js への対話割り込み機能の実装

Implementation of barge-in function on MMI.js : the JavaScript library realizing Web based Multimodal Interaction system

田中　遼 [1]　桂田　浩一 [1]　入部　百合絵 [2]　新田　恒雄 [1,3]

Ryo Tanaka[1], Kouichi Katsurada[1], Yurie Iribe[2], and Tsuneo Nitta[1,3]

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

[2] 愛知県立大学
[2] Aichi Prefectural University

[3] 早稲田大学
[3] Waseda University

Abstract: This paper provides barge-in function that is incorporated into a JavaScript library called "MMI.js" which enables 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. By added the barge-in function, developers can construct multimodal interaction systems that can deal with unintended inputs from user.

1. はじめに

近年, スマートフォンやタブレットの普及により, Siri[1]やしゃべってコンシェル[2]を始めとする対話エージェントの利用が活発になってきている．こうした対話エージェントを web アプリケーション開発者が容易に開発できるようにするため，筆者が所属する研究室では，これまでに Web ブラウザ上で動作するマルチモーダル対話システムを実現するための JavaScript 用ライブラリ MMI.js[3]を提案してきた．しかし MMI.js ではエージェントとの対話中に生じる割り込みに対応する機能を備えていなかった．そのためユーザの咄嗟の入力に対応する機能を対話エージェントに組み込むことができなかった．そこで本論文では，MMI.js を改良し，対話割り込み機能を追加することによって，ユーザが不便さを感じない対話エージェントの開発を可能にする．

こうした割り込み機能はすでに音声対話を記述する言語 VoiceXML[4]やマルチモーダル対話を記述する言語 XISL[5]において，複数の対話タスク（本タスクと割り込み対話のタスク）を同時に受け付ける方法により実現されている．本研究ではこれらの技術を参考に，プログラム実行中の任意の場所で割り込みを発生させる事ができる機能を組み込むと共に，有効範囲の設定や，割り込み終了後の動作の指定につても記述できるよう MMI.js を拡張する．

以下では, まず先行研究である MMI.js の概要と特徴を説明した後に MMI.js の改良内容について述べる．

2. 先行研究

まず本論文で改良するライブラリのベースとなる MMI.js について説明する．MMI.js は Web ブラウザ上で動作するマルチモーダル対話システムを記述するための JavaScript 用ライブラリである．Web ブラウザがあれば動作するので，開発者はプラットフォームを気にする必要がなく，特別なソフトウェアのインストールなしに気軽に利用することができる．

MMI.js には主要な機能として逐次的，同時的，択一的なマルチモーダル入出力機能，マルチメディアコンテンツの同期制御機能，エージェントの制御機能が備えられている．本節ではこれらの機能について説明する．

2.1. マルチモーダル入出力機能

MMI.js ではマルチモーダル入出力を実現するために，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 が用意されている．また，逐次的，同時的，択一的な入出力を行う関数 mmi.Seq，mmi.Par，mmi.Alt が用意されている．マルチモーダル入力を

受け付ける際には，type（入力手段），event（入力に利用するイベント），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．入力結果は mmi.execute 関数に指定した関数オブジェクトの引数に渡され，入力結果を用いた対話において利用される．

表 1 に入出力関数の一覧を，図 1 に MMI.js を用いた対話の記述例を示す．図 1 には択一的入力と同時的入力の二つの例が記述されている．2 行目から始まる mmi.altInput 関数はエージェントをクリックまたは音声入力のどちらかを受け付ける同時的入力である．また，15 行目から始まる mmi.parOutout 関数はエージェントが指さし動作を行いながら「商品を選択してください」と発話する同時的出力である．それぞれ，入出力の内容と共に次節で説明する同期制御のための引数が指定されている．

2.2. マルチメディアコンテンツ同期制御

MMI.js では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 SMIL[6]で提供されている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 種類の引数が用意されている．入出力の開始，終了を制御する際には引数“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際には引数“dur”に継続時間を記述することができる．これらの機能は表 1 に示したマルチモーダル入出力関数に引数を与えることにより利用できる．

図 1 の例では，5 行目と 10 行目の“begin”に 1000，6 行目と 11 行目の“dur”に 10000 が設定されているため，プログラムが読み込まれてから 1 秒後に 10 秒間継続する入力の受付を開始することになる．また，15 行目の mmi.parOutout 関数の部分では 19 行目と 24 行目に“begin”に 500 が設定されているため，2 行目の入力の後 0.5 秒後に出力を行うことになる．

2.3. エージェントの動作

表 1　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
マルチモーダル入出力	mmi.Seq	逐次的な入出力を行う
	mmi.Par	同時的な入出力を行う
	mmi.Alt	択一的な入出力を行う

```
1   // 択一的入力
2   mmi.altInput({
3       “type”: “click”,
4       “match”: “agent”,
5       “options”:{   “begin”:1000,
6                     “dur”   :10000    }
7   },{
8       “type”: “speech”,
9       “grammar”: “かいものをしたい”,
10      “options”:{   “begin”: 1000,
11                    “dur”   : 10000    }
12  });
13
14  // 同時的出力
15  mmi.parOutput({
16      “type”: “agent”,
17      “event”: “speech”,
18      “text”: “商品を選択してください”，
19      “options”:{   “begin”: 500    }
20  },{
21      “type”: “agent”,
22      “event”: “gesture”,
23      “gesture”: “point”,
24      “options”: {   “begin”: 500    }
25  });
```

図 1　対話の記述例

筒井らが提案した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 MPML[7]や，NHK が開発したテレビ番組を記述するための言語 TVML[8]を参考に，MMI.js では“お辞儀”や“指さし”といったエージェントの 62 種類の動作が用意されている．これらの機能もメディアの同期制御機能と同様に，入出力関数に引数を与えることで利用可能になる．図 1 の例では 23 行目で“gesture”に“point”が指定されているため，エージェントは発話と同時に指さしの動作を行う．

3. 対話割り込み機能の追加

3.1. 要求される機能

対話割り込みは対話中において常に発生する可能性がある．たとえば，ユーザが電車に乗るなどユーザ環境の変化により対話を一時中断する場合や，対話を急きょ終了する場合などである．

VoiceXML や XISL ではルートドキュメント（割り込み対話）とそれに付随するリーフドキュメント（本タスクの対話）を同時に入力の受け付け可能にすることで割り込み対話に対応している．この機能は対

話中のどのような場面でも利用可能であることから，対話エージェントの開発者は有効範囲を気にせずに割り込み対話を記述することができる．MMI.js で割り込み対話をサポートする際にも，割り込み可能な範囲を限定しないことが望ましい．

一方で，現在進行している対話の内容によっては途中で割り込みが発生することが不都合な場合もある．たとえば，エージェントが注意や警告など非常に重要なことを発話している最中に割り込みが発生する場合である．そこで，割り込み対話の有効な範囲を指定できるようにする機能もサポートすることが望ましい．

また，割り込み対話終了後の動作は VoiceXML や XISL ではこれらのインタプリタによって適切な位置に復帰する．ただし goto 要素等を用いることで任意の位置へ対話を遷移させることもできる．これによって対話を適切な位置から再開することができ，割り込み対話と割り込み後の対話が自然につながる．しかし，進行中の対話が中途半端な状態のときに割り込みが発生した場合，割り込み発生直前の位置から対話を再開するとエージェントの動作や発話が不自然になる可能性がある．これを回避するために，自然な区切り位置から対話に復帰する手段が必要であると考えられる．

本研究では，以上から次の要求仕様を満たす対話割り込み機能を実現する．

【要求仕様】

1. 割り込み可能な範囲を限定しない
2. 割り込み対話の有効範囲を設定できる
3. 割り込み対話終了後は任意の位置への遷移だけでなく割り込み前の対話に復帰可能にする

3.2. 対話の記述方法

要求仕様の 1 を満たすために割り込み対話を記述する mmi.bargein 関数を用意した．この関数に割り込み発生条件と割り込み対話のシナリオを指定することで，割り込みの受付を開始する．この関数を用いて記述された割り込み対話は有効範囲を設定していない限り，プログラム実行中の任意の場所で割り込むことが可能である．関数の使用例を図 2 に示す．13 行目～21 行目の mmi.bargein 関数では，15 行目の “button_skip”という ID を持つボタンをクリックするか，音声で「スキップ」と発話することによって割り込むことができる択一的な入力を受け付けている．この割り込みは，次節で説明する有効範囲から外れる 40 行目の出力部分以外ならば発生させることができる．この割り込みが発生した場合，エージェントが「スキップしますね」と発話し，一つニュースをスキップする．mmi.bargein 関数の第一引数には他にも“seq”と“par”を指定することができ，逐次的入力

```
1   // 割り込み対話 1
2   mmi.bargein( “alt”,
3     {{ “type” : “speech”, “grammar” : ”はじめから” },
4       { “type” : “click”, “match” : “#button_start” }},
5     function() {
6       mmi.output({ “type” : “agent”, “event” : “speech”,
7         ”text” : ”もう一度始めから説明しますね” });
8
9       mmi.goto({ “type” : ”bookmark”, “name” : “start” });
10  });
11
12  // 割り込み対話 2
13  mmi.bargein( “alt”,
14    {{ “type” : “speech”, “grammar” : “スキップ” },
15      { “type” : “click”, “match” : “#button_skip” }},
16    function() {
17      mmi.output({ “type” : “agent”, “event” : “speech”,
18        “text” : “スキップしますね”});
19
20      mmi.goto({ “type” : “succeeding” });
21  }, “skip_bargein” );
22
23  // 割り込み対話 3
24  mmi.bargein( “alt”,
25    {{ “type” : “speech”, “grammar” : “もういちど” },
26      { “type” : “click”, “match” : “#button_skip” }},
27    function() {
28      mmi.output({ “type” : “agent”, “event” : “speech”,
29        “text” : “今の記事を読み直しますね” });
30
31      mmi.goto({ “type” : “previous” });
32  });
33
34  // 主の対話
35  mmi.scenario( function() {
36    mmi.bookmark( “start” );
37
38    mmi.bargeinState( “skip_bargein”, false );
39
40    mmi.output({ “type” : “agent”, “event” : “speech”,
41      “text”:”今日の重要なニュースは○○です”});
42
43    mmi.bargeinState( “skip_bargein”, true );
44
45    mmi.output({ “type” : “agent”, “event” : “speech”,
46      “text” : “<BOOKMARK mark=’news1’/>”
47        + “一つ目のニュースは××です”
48        + “ <BOOKMARK mark=’news2’/>”
49        + “二つ目のニュースは△△です” });
50  });
```

図 2　記述例

と同時的入力を割り込み条件に指定することができる．

3.3. 割り込みの有効範囲の指定

要求仕様の 2 を満たすために mmi.bargeinState 関数を用意した．この関数の第 1 引数に割り込み対話の ID（mmi.bargein 関数の第 4 引数）を指定することで特定の割り込み対話を有効／無効にできる．有効にする際には第 2 引数に true を指定し，無効にする際には false を指定する．図 2 では 38 行目の mmi.bargeinState 関数が呼び出されるまで割り込み対話“skip_bargein”が有効であり，その後無効となり，43 行目の mmi.bargeinState 関数が呼び出されると再び割り込み対話が有効になる．

3.4. 割り込み対話終了後の動作

要求仕様の 3 を満たすために mmi.bookmark 関数，および mmi.goto 関数を用意した．また，発話文中への対話の遷移ができるように，日本語テキスト音声合成用記号[10]で定義された BOOKMARK 要素を発話文中に挿入できるようにした．mmi.bookmark 関数または BOOKMARK 要素で任意の位置にブックマークを設定し，mmi.goto 関数を用いて任意のブックマークまで対話を遷移させることができる．例えば図 2 の 24 行目～32 行目の mmi.bergein 関数では 31 行目にある mmi.goto 関数の引数内の“type”に“previous”を指定している．“previous”を指定することで，割り込み発生位置の直前のブックマークから対話を再開できる．したがって図 2 の例の 47 行目の発話文「一つ目のニュースは××です」の部分でこの割り込みが発生した場合，46 行目の“news1”という名前のブックマークの直後から再開し，「一つ目のニュースは××です」の部分を再度発話することになる．一方で，“type”に“succeeding”を指定すると，割り込み発生位置の直後のブックマークから対話を再開することができる．図 2 においては 13 行目～21 行目の mmi.bargein 関数の 20 行目で“succeeding”が指定されている．このため，47 行目の発話文の部分でこの割り込みが発生した場合，48 行目の“news2”という名前のブックマークの直後から再開し，49 行目の発話文「二つ目のニュースは△△です」の部分から発話を開始することになる．

また，“type”に“bookmark”を指定し，“name”にブックマークの名前を指定すると，そのブックマークに対話を遷移することができる．図 2 では 9 行目にある mmi.goto 関数の引数に“start”という名前のブックマークを指定しているため，2 行目～10 行目の割り込みの終了後は 36 行目の“start”という名前のブックマークへ対話が遷移する．

4. まとめ

本論文では，Web ブラウザ上で動作するマルチモーダル対話システムを実現する JavaScript 用ライブラリ MMI.js への対話割り込み機能の追加について報告した．これによりプログラム実行中の任意の場所で割り込みを発生させる事や，割り込み可能な有効範囲，割り込み終了後を動作を指定可能にし，現実に発生する割り込み対話に幅広く対応できるようにした．しかし現在の仕様では，対話を階層的に記述することができないため，割り込み対話の有効範囲を設定する時にソースコードが冗長になるという問題がある．今後は対話を階層的に記述する仕組みの導入のほか，モダリティを容易に追加できるようにする仕組みや，エージェントを可変にする機能，システム発話から指示対象を同定する[10]機能等を組み込みたい．

参考文献

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

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

[3] 菊地泰己，桂田浩一，入部百合絵，新田恒雄: JavaScript ライブラリ MMI.js を用いた Web ブラウザ上でのマルチモーダル対話の実現，情報処理学会全国大会講演論文集, Vol. 2013, No. 1, pp. 97-99, (2013)

[4] http://www.w3.org/TR/voicexml21/

[5] 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)

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

[7] 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)

[8] 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)

[9] 蓑輪 利光，赤羽 誠，板橋 秀一: JEIDA 日本語テキスト音声合成用記号，日本音響学会研究発表会講演論文集, Vol. 2000, No. 2, pp. 183-184, (2000)

[10] 松山 匡子，駒谷 和範，武田 龍，尾形 哲也，奥乃 博: バージイン許容音声対話システムにおけるユーザ発話の分析と指示対象同定への応用，情報処理学会研究報告, Vol.2010, No.21,pp. 1-6, (2010)