

# マルチモーダル対話システム構築のための JavaScript ライブラリ MMI.js における対話割り込み機能の実現

田中 遼<sup>†</sup> 桂田 浩一<sup>†</sup> 入部 百合絵<sup>‡</sup> 新田恒雄<sup>††</sup>  
 豊橋技術科学大学<sup>†</sup> 愛知県立大学<sup>‡</sup> 早稲田大学<sup>††</sup>

## 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、マルチモーダル対話システム構築のための 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つた。そこで本論文では、MMI.js を改良し、対話割り込み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いても記述できるように拡張する。これにより、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。

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

まず本研究のベースとなる 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。マルチモーダル入出力機能としては、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`、逐次的、同時的な出力を行う関数 `mmi.seqOutput`、`mmi.parOutput`、および逐次的、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。表 1 に入出力関数の一覧を示す。

マルチメディアコンテンツの同期制御には、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]が持つメディア同期機能を参考にした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ている。

エージェントの制御機能としては、筒井らが提案したエージェントを用いるプレゼンテーション記述言語 MPML[7]や、NHK が開発したテレビ番組の記述言語 TVML[8]を参考に、“お辞儀”や“指さし”といった `Embedding barge-in function in MMI.js: a JavaScript library for constructing multimodal interaction systems`

<sup>†</sup>Ryo Tanaka, Kouichi Katsurada, Toyohashi Univ. of Tech

<sup>‡</sup>Yurie Iribe, Aichi Prefectural University

<sup>††</sup>Tsuneo Nitta, Waseda University

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	<code>mmi.seqInput</code>	逐次的な入力を受付ける
	<code>mmi.parInput</code>	同時的な入力を受付ける
	<code>mmi.altInput</code>	択一的な入力を受付ける
マルチモーダル出力	<code>mmi.seqOutput</code>	逐次的な出力を行う
	<code>mmi.parOutput</code>	同時的な出力を行う
マルチモーダル入出力	<code>mmi.Seq</code>	逐次的な入出力を行う
	<code>mmi.Par</code>	同時的な入出力を行う
	<code>mmi.Alt</code>	択一的な入出力を行う

```
// 択一的入力
mmi.altInput({
  "type": "click",
  "match": "agent",
  "options": { "begin": 1000, "dur": 10000 }
});

// 同時的出力
mmi.parOutput({
  "type": "agent",
  "event": "speech",
  "text": "商品を選択してください",
  "options": { "begin": 500 }
});

mmi.parOutput({
  "type": "agent",
  "event": "gesture",
  "gesture": "point",
  "options": { "begin": 500 }
});
```

図 1 対話の記述例

エージェントの 62 種類の動作が用意されている。

図 1 に MMI.js の記述例を示す。

## 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。例えば、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、対話を急ぎよ終了する場合などである。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。例えば、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。そのため、対話中のどのような場面でも対話の割り込みを行えるが、割り込みを行うことができる有効な範囲も同時に指定できることが望ましい。

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や goto 要素等によって割り込み対話と割り込み後の対話が自然につながるように調整される。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. 任意の場所に割り込み対話を設定できる
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. 割り込み対話の記述方法

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対話の ID、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。

### 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。この関数に割り込み対話の ID と有効 (true) / 無効 (false) を切り替えるフラグ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。

また、対話の階層的な表現を可能にする `mmi.subdialog` 関数を用いることで簡潔に有効範囲を指定することもできる。この関数の引数に特定の割り込み対話を有効 / 無効にするオプションを追加することで、この関数内に記述された対話中ではその割り込み対話を有効 / 無効にできる。

### 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。

## 4. まとめ

本論文では、マルチモーダル対話システム構築のための 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、割り込み終了後の動作を指定可能にし、現実には発生する割り込み対話に幅広く対応できるようにした。今後はモダリティを容易に追加できるようにする仕組みのほか、エージェントを変更可にする機能、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。

```
// 割り込み対話
mmi.bargein("skip_bargein", "alt",
  {"type": "speech", "grammar": "スキップ"},
  function() {
    mmi.output({"type": "agent", "event": "speech",
      "text": "スキップしますね"});

    mmi.goto({"type": "succeeding"});
  });

// 主の対話
mmi.scenario(function() {
  mmi.bookmark("start");
  mmi.bargeinState("skip_bargein", false);

  mmi.output({"type": "agent", "event": "speech",
    "text": "今日の重要なニュースは〇〇です"});

  mmi.subdialog("news", {"skip_bargein": true},
    function() {
      mmi.output({"type": "agent", "event": "speech",
        "text": "<BOOKMARK mark='news1'/>"
          + "一つ目のニュースは××です"
          + "<BOOKMARK mark='news2'/>"
          + "二つ目のニュースは△△です"});
    });
});
```

図 2 割り込み対話の記述例

## 参考文献

- [1] <http://www.apple.com/jp/ios/siri/>
- [2] [http://www.nttdocomo.co.jp/service/information/shabette\\_concier/](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)