

# MMI 記述言語 XISL の W3C-MMI フレームワークへの対応について

中島 将宏<sup>†</sup> 中村 有作<sup>†</sup> 桂田 浩一<sup>†</sup> 山田 博文<sup>†</sup> 新田 恒雄<sup>†</sup>

<sup>†</sup>豊橋技術科学大学 大学院工学研究科 知識情報工学専攻

<sup>‡</sup>豊橋技術科学大学 マルチメディアセンター

## 1 はじめに

近年、Web アクセシビリティ向上のため、複数のモダリティを用いるマルチモーダル対話 (MMI) が注目されている。MMI では音声入出力やペン入力といった複数モダリティが同時、あるいは選択的に利用されるなど、複雑な対話を記述することが求められるが、現在、MMI 記述に関する標準規定はない。そのため、W3C は Web アクセスにおける MMI 利用の標準を定めるべく MMI ワーキンググループ (W3C-MMI-WG) を結成した [1]。W3C-MMI-WG ではモダリティ制御のための記述言語 (MMI 記述言語) の要求仕様 [2] や、MMI システムのフレームワーク [3] が検討されている。一方、我々もこれまで MMI 記述言語 XISL [4] と XISL 実行システム [5] を提案・検討してきた。今後これらの MMI 記述言語/システムとしての利便性を高めるには、W3C-MMI-WG の標準に対応していく必要がある。

本稿では W3C-MMI-WG の提案と我々の提案との相違点を述べ、標準化対応について報告する。

## 2 W3C-MMI-WG の標準化作業

W3C-MMI-WG では、主に以下の二項目が検討されている。

- a. MMI 記述言語への要求仕様
- b. MMI フレームワーク

a については策定が完了しており [2]、b についても、ノート [3] が公開され外部から閲覧可能になっている。以降でこれらの概略を述べる。

### 2.1 MMI 記述言語への要求仕様

W3C-MMI-WG が提示している MMI 記述言語に対する要求仕様を以下に示す。

- 1) 一般的な要求  
様々なモダリティを用いた対話や、マルチモーダル入出力の制御が可能でなければならない。
- 2) 入出力モダリティに関する要求  
逐次・同時的な入出力の取り扱いや、モダリティの拡張が可能でなければならない。
- 3) アーキテクチャ及び同期・統合に関する要求  
アプリケーションロジックとコンテンツを分離して記述できなければならない。また、SMIL2.0 [6] の入出力同期メカニズムを導入しな

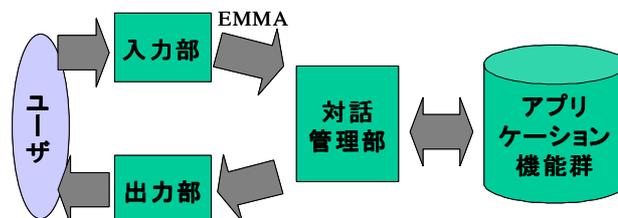


図1. MMI フレームワークのコンポーネント構成

なければならない。

この他にも、モバイルアクセスへの対応等の実行環境に関する要求も提示されている。

## 2.2 MMI フレームワーク

W3C-MMI-WG は図 1 に示すコンポーネント構成の MMI フレームワークを提案している。

入力部は音声認識等のユーザ入力の解釈を行い、その結果を対話管理部に通知する。対話管理部は入出力部の制御や対話の進行を管理する。出力部は対話管理部の指示に従いユーザへの出力を行う。アプリケーション機能群はデータベースやアプリケーションの処理に必要なコンポーネントである。

入力部 - 対話管理部間の通信には EMMA ( :Extensible Multimodal Annotation Language ) が用いられる。EMMA はマルチモーダル入力情報を表現するための XML 言語で、これによりモダリティに依存しない抽象的な入力情報の表現が可能になる。なお、EMMA の詳細は決定されておらず、現在は記述方法について検討中である。

## 3 XISL と XISL 実行システム

### 3.1 XISL と MMI-WG 要求仕様

XISL は XML をベースとした MMI 記述言語で、マルチモーダル入出力や対話制御を記述するタグセットを備えている。これは要求仕様の 1) を満たしている。

XISL は複数モダリティによる同時/逐次入出力や択一入力が記述可能である。また、入出力の記述に自由度を持たせているため、モダリティの拡張が可能である。さらに、XISL は対話シナリオのみを記述するため、アプリケーションロジックを他のコンテンツと分離できる。これらは要求仕様の 2) および同期・統合以外の 3) の仕様をそれぞれ満たしている。

このように XISL の仕様は、SMIL2.0 の同期メカニズムに関する点を除いて W3C-MMI-WG の要求をほ

Accommodation of XISL to W3C-MMI-Framework.  
Masahiro Nakajima<sup>†</sup>, Yusaku Nakamura<sup>†</sup>, Kouichi Katsurada<sup>†</sup>, Hirobumi Yamada<sup>‡</sup> and Tsuneo Nitta<sup>†</sup>  
Graduate School of Engineering, Toyohashi Univ. of Tech.<sup>†</sup>  
Faculty of Engineering, Toyohashi Univ. of Tech.<sup>‡</sup>

ば満たしていると言える。

### 3.2 XISL 実行システムと MMI-WG フレームワーク

XISL 実行システムは、対話の進行を管理する対話制御部と、入出力を制御するフロントエンドの独立した二つのモジュールからなり、別途外部の Web サーバを利用する。MMI フレームワークと比較すると、フロントエンドは入出力部、対話制御部は対話管理部、WWW サーバはアプリケーション機能群に相当する。この構成は W3C-MMI-WG の MMI フレームワークとほぼ同一であるが、モジュール間で行う通信の内容には独自の仕様を用いている。この部分は、今後、EMMA に合わせる必要がある。

## 4 標準化への対応

### 4.1 SMIL2.0 の同期メカニズムの導入

SMIL2.0 はマルチメディア出力を制御するための XML 言語である。SMIL2.0 の同期メカニズムは“タイミング及び同期モジュール”としてまとめられている。そこで、このモジュールの要素や属性を XISL の入出力記述に取り込むことにした。

図 2 に同期メカニズムを用いた出力の記述例を示す。図の<output> は一つの出力を表現する要素で、図 2 では a と b の二つの出力が記述されている。この二つは SMIL2.0 の要素である<smil:par>で括られているため、通常は同時に実行される。しかし、b は begin 属性として ‘smil:begin="navi.end+1s"’ が指定されているため、id 属性値が “navi” である a の<output>の終了から 1 秒経過するまで実行されない。

このように、SMIL2.0 の同期メカニズムを導入することで、開始/終了イベントや時間設定による細かい同期制御を行う事ができる。なお、記述を統一するため、同期制御に関わる XISL の要素は全て SMIL2.0 の要素と置き換えることにした。

### 4.2 EMMA の導入

図 3 は、ポインティングと音声のマルチモーダル入力結果の情報を表す EMMA である。EMMA は情報の構造を示すモデル (A) と情報の実体 (B) からなる。B はさらにモダリティの入力情報 (C1, C2) とインスタンス (D) で構成される。インスタンスは A を C1, C2 によって埋めたものである。

XISL 実行システムでは、EMMA はフロントエンドと対話制御部の二箇所で作成される。図 3 の例の場合、フロントエンドではポインティング入力情報のみの EMMA (A-B-C1-D からなる EMMA) と音声入力情報のみの EMMA (A-B-C2-D からなる EMMA) が生成される。この時、各 EMMA のインスタンス D は、<num>と<goods>の一方が埋められ他方が空の状態である。この二つの EMMA が対話制御部によって統合され、互いに空の情報を補い図 3 の EMMA が得られる。

上に述べた処理を行うには、C1 及び C2 の入力<num>と<goods>のどちらを埋めるか、すなわち各<input>と<instance>内の要素との関係を明確にし

```
<smil:par>
  <output id="navi" type="browser" ----- a
    <param>
      http://www.vox.tutkie.tut.ac.jp/
    </param>
  </output>
  <output smil:begin="navi.end+1s" ----- b
    type="tts" event="speech">
    <param>
      新田研究室のページへようこそ
    </param>
  </output>
</smil:par>
```

図 2. 同期出力の記述例

```
<result>
  <data_model> ----- A
    <order>
      <num></num>
      <goods></goods>
    </order>
  </data_model>
  <interpretation confidence="90"> ----- B
    <input mode="pointing" timestamp="YYY"> - C1
      <x-coordinate="34"/>
      <y-coordinate="58"/>
    </input>
    <input mode="speech" timestamp="XXX"> --- C2
      これを 3 ください
    </input>
    <instance> ----- D
      <order>
        <num>3</num>
        <goods>りんご</goods>
      </order>
    </instance>
  </interpretation>
</result>
```

図 3. マルチモーダル入力の EMMA

なければならない。しかしながら、これは対話の内容や状況に依存するため、単一のアルゴリズムを適用するのは困難である。そこで我々は EMMA の生成・統合アルゴリズムをルールで与えると共に動的に変更する仕組みを検討している。

## 5 まとめ

W3C-MMI-WG で検討中の要求仕様と MMI フレームワークを紹介し、XISL と XISL 実行システムの対応を述べた。XISL は、SMIL2.0 の同期メカニズムを導入することで、W3C-MMI-WG の要求仕様への対応を実現できる。今後は EMMA の生成・解釈メカニズムとルール記述方法について検討していく。また同時に、XISL の記述支援ツールも開発していく予定である。

### 参考文献

- [1] <http://www.w3c.org/2002/mmi/>
- [2] <http://www.w3.org/TR/mmi-reqs/>
- [3] <http://www.w3c.org/TR/mmi-framework/>
- [4] <http://www.vox.tutkie.tut.ac.jp/XISL/XISL.htm>
- [5] Katsurada, K., Ootani, Y., Nakamura, Y., Kobayashi, S., Yamada, H., Nitta, T.: “A Modality-Independent MMI System Architecture,” Proc. of ICSLP2002, pp.2549-2552 (2002).
- [6] <http://www.w3.org/TR/smil20/>