

MMI 記述言語の標準化動向と XISL の対応について

中村 有作[†] 桂田 浩一[†] 山田 博文[‡] 新田 恒雄[†]

[†] 豊橋技術科学大学 大学院工学研究科 知識情報工学専攻

[‡] 豊橋技術科学大学 マルチメディアセンター

〒441-8580 愛知県豊橋市天伯町雲雀ヶ丘 1-1

E-mail: [†]{nakamura,katurada,nitta}@vox.tutkie.tut.ac.jp, [‡]yamada@vox.tutkie.tut.ac.jp

あらまし 本報告では、W3C におけるマルチモーダル対話(MMI)記述言語標準化作業について紹介すると共に、筆者らが進めている XISL での実現案を説明する。W3C は、MMI による Web ユーザインターフェースの向上を目的として、W3C-MMI-WG を結成した。WG は、主に MMI 記述言語の要求仕様や MMI フレームワークを提案・検討している。今後、XISL をこれらの標準化作業に対応させていく必要がある。そこで、MMI 記述言語の要求仕様と XISL を、MMI フレームワークと XISL 実行システムをそれぞれ比較し、対応方法を検討した。要求仕様では、各モダリティ間の同期・統合に関する仕様を XISL は満足していなかった。新しい XISL 仕様では、SMIL 2.0 の同期メカニズムを採用することにより対応を行った。また MMI フレームワークでは、対話管理部に送られる新しい EMMA (Extensible Multi-Modal Annotation Markup Language) 形式に合わせて、マルチモーダル入出力に関する XISL の記述を、これまでの個別モダリティに対する統合制御記述から、入出力を統合解釈した後の抽象的レベルの記述へと仕様変更した。これらの改良により、XISL の記述は端末毎のモダリティに依存せず、様々な端末で再利用できるようになる。反面、各モダリティの解釈方法と統合方法という実装に関する部分の検討が、今後必要になる。

キーワード マルチモーダル対話、対話記述言語、W3C-MMI-WG、XISL、 XHTML+Voice、SALT、SMIL2.0、EMMA

Standardization of Multi-modal Interaction Description Languages and Improvement of XISL to follow it

Yusaku NAKAMURA[†] Kouichi KATSURADA[†] Hirobumi YAMADA[‡] and Tsuneo NITTA[†]

[†] Graduate School of Technology, Toyohashi University of Technology

[‡] Multimedia Center, Toyohashi University of Technology

1-1 Hibarigaoka, Tempaku-cho, Toyohashi 441-8580, JAPAN

E-mail: [†]{nakamura,katurada,nitta}@vox.tutkie.tut.ac.jp, [‡]yamada@vox.tutkie.tut.ac.jp

Abstract In this paper, we introduce activities of W3C-MMI-WG in which the standardization of multi-modal interaction (MMI) description language is targeted, then describe on our approach to specifications of the standardization. W3C-MMI-WG is mainly discussing the requirements and framework of MMI systems for designing MMI applications used across networks. In parallel with this activity, we have developed an MMI description language XISL, an execution system, and investigated them in an MMI system. This paper compares the difference of the specifications and the MMI framework of XISL from those of W3C-MMI-WG to clarify the parts to be discussed for modifying XISL. As a result, XISL does not completely satisfy to the synchronization among modalities in the requirements and to the interpretation mechanism named EMMA (Extensible Multi-Modal Annotation markup language) in the MMI framework discussed in the WG. To conform the requirement and the framework, we introduce a control mechanism of SMIL 2.0 that can synchronize modalities both in input and output media and we also investigate a method to interpret input modalities as an EMMA format.

Keyword Multi-Modal Interaction, W3C-MMI-WG, XISL, XHTML+Voice, SALT, SMIL2.0, EMMA

1.はじめに

近年、PC、PDA、電話など様々な端末から利用できるWebアプリケーションに対する関心が高まっている。多様な端末から利用可能なアプリケーションは、利用者の端末環境を特定しないため、アクセサビリティの向上に大きく役立つ。しかし、こうしたアプリケーションの開発には、音声入力、ペン入力といった多様なモダリティを制御しなければならないが、現在のWeb技術にこうした機能は備わっていない。今後は、多様なモダリティを扱うことのできるマルチモーダル対話(MMI: Multi-Modal Interaction)技術の導入が、Webへのアクセスにとって必要不可欠になる。

W3Cは、上記の要求に答える形で、今年3月にMMIワーキンググループ(W3C-MMI-WG[1])を結成した。このWGは、複数のモダリティやデバイスを制御するための記述言語(MMI記述言語)の要求仕様や、MMIシステムのフレームワークを検討している。一方、著者らの研究グループでは、以前から独自のMMI記述言語XISL(Extensible Interaction Scenario Language)[2], [3], [4], [5]を提案・検討してきた。しかし、今後、Webアクセス向けにはWGの標準化作業の中で出てくる仕様に、XISLを対応させていく必要があると考えている。

本報告では、前半(2章、3章、4章、5章)で、W3C-MMI-WGの活動と、その中で検討されているMMI記述言語の要求仕様、MMIシステムのフレームワークについて説明する。また、MMI記述言語に関する現状のアプローチをXISLと比較しながら紹介する。また後半(6章、7章)では、要求仕様に対するXISLの問題点とそれに対する対応について述べる。さらにMMIフレームワークとXISL実行システムのアーキテクチャを比較した上で、XISLの改良案を示し、これによって得られる利点と今後の検討課題を考察する。

2. W3C-MMI-WGの活動

W3Cは、WWW(World Wide Web)で使用される様々な技術の標準を定める団体である。W3C-MMI-WGは、MMIによるWebアクセスを実現することを目的として結成された、W3Cのワーキンググループである。

W3C-MMI-WGでは、以下の2つのテーマに関して主に提案・検討が行われている。

- 1) MMI記述言語に対する要求仕様
 - 2) MMIシステムのフレームワーク
- 1)については3章で、2)については4章で詳細に紹介する。これらのテーマ以外にもWGでは、各種デバイス間でやり取りされるイベントの処理方法、ペンモダリティ(ink)による手書き文字情報(ペンの色、圧力、位置、速度等)を表現するための記述言語InkXMLなども提案・検討されている。

3. 要求仕様

本節で示す要求仕様は、各参加機関から提供されたMMIのユースケースを元に作成されている。要求仕様は大きく分けて、一般的な要求、入力モダリティに関する要求、出力モダリティに関する要求、アーキテクチャ及び統合・同期に関する要求、そして実行環境やネットワーク上の配備に関する要求から構成される。

3.1. 一般的な要求

一般的な要求は以下の通りである。

- 1-A) 各種モダリティやデバイスを扱えなければならない。
- 1-B) 複数のモダリティを組み合わせた対話を扱えなければならない。
- 1-C) 各モダリティ間の同期と、複数モダリティから構成されるマルチモーダル入出力間の同期を扱えなければならない。

1-D) 容易に記述・実装できなければならない。

この他にも、マルチリンガル、プライバシー、セキュリティなどに関する要求仕様が提示されている。

3.2. 入力モダリティに関する要求

入力モダリティに関する主要な要求は以下の通りである。

- 2-A) 各入力モダリティの処理に関する情報(使用する認識エンジン、及び文法ファイル等)を扱える機構を持たなければならない。
- 2-B) 逐次的な入力を扱えなければならない。
- 2-C) 複数の異なるモダリティからの同時入力を、それぞれ別々の入力として扱えなければならない。
- 2-D) 複数の異なるモダリティからの同時入力を、1つのマルチモーダル入力として扱えなければならない。
- 2-E) 新規モダリティの追加をサポートしなければならない。

この他にも、サポートすべきモダリティやデバイス(音声、ポインティング、キーボード等)などに関する要求仕様が提示されている。

3.3. 出力モダリティに関する要求

出力モダリティに関する主要な要求は以下の通りである。

- 3-A) 逐次的な出力を扱えなければならない。
- 3-B) 同時出力を扱えなければならない。
- 3-C) 新規モダリティの追加をサポートしなければならない。

この他にも、サポートすべきモダリティやデバイスに関する要求仕様が提示されている。

3.4. アーキテクチャ及び統合・同期に関する要求

アーキテクチャ及び統合・同期に関する主要な要求は以下の通りである。

- 4-A) SMIL2.0[6]で行われている入力・出力メディア間の同期メカニズムを採用しなければならない。
- 4-B) MMI記述言語は、モデルとビューを分離しなければならない。

その他にも、入力・出力インターフェースの分離や、利用者の端末での利用可能なモダリティの検出、その端末環境への適応についての要求仕様が提示されている。

3.5. 実行環境とネット上の配備に関する要求

MMIアプリケーションの実行環境や構成に関する要求が提示されている。例えば、MMIアプリケーションは、それぞれ独立したいくつかのモジュール(ネットワーク上に分散していることもある)から構成されなければならないらず、それらのモジュールの設定は、動的に変更できるなどの要求仕様が提示されている。

4. MMIフレームワーク

W3C-MMI-WGではMMIシステムのフレームワークが検討されており、MMIシステムを構成する主要コンポーネントや、コンポーネント間でやり取りされるデータを記述する、以下に述べるEMMAという言語など検討されている。図1にMMIフレームワークを構成する主要コンポーネントを示す。MMIフレームワークは、入力部、出力部、対話管理部、及びアプリケーション機能群から構成される。入出力部は、マルチモーダル入出力を解釈するコンポーネントである。対話管理部はユーザの入力に対するシステムの応答を決定し、対話を進行・管理するコンポーネントである。対話の管理には、SALT[7]、 XHTML+Voice[8]、XISLといった言語を用いることが検討されている。

アプリケーション機能群は、データベースへのアクセス、トランザクション処理、及びアプリケーションで必要な演算等を行うコンポーネントである。アプリケーション機能群に関する仕様は、各アプリケーションに依存するものであり、MMI フレームワークの仕様外となっている。

図 2 に、入力部を構成する各サブコンポーネントを示す。入力部は、大きく分けて認識部、解釈部、入力統合部から構成される。認識部は、各モダリティの認識エンジン（音声認識、手書き文字認識 etc.）から構成される。解釈部は、各モダリティの入力を解釈する。例えば音声解釈部は、「うん」「OK」「もちろん」などの入力を、「はい」として解釈する。入力統合部では、各モダリティの解釈部から受け取った複数の解釈情報を、1 つのマルチモーダル入力として統合する。解釈部から入力統合部、及び入力統合部から対話管理部への情報伝達には EMMA が用いられる。

4.1. EMMA (Extensible Multi-Modal Annotation Markup Language)

EMMA とは、マルチモーダル入出力データを表現するための言語である。詳細な仕様は決定されておらず、現在 W3C-MMI-WG 内で検討中である。

MMI フレームワークでは、マルチモーダル入出力データを、EMMA 形式で統一することにより、モダリティに依存しない解釈を実現しようとしている。EMMA の記述例を図 3、図 4 に示す。図 3 は、表 1 に示したポインティング認識エンジンの認識結果をもとに、ポインティング解釈部が生成した EMMA の記述例である。図 4 は、表 1 に示す音声認識エンジンの認識結果をもとに、音声解釈部が生成した EMMA の記述例である。ここでは、図 3 の EMMA をもとに、EMMA の構造を説明する。

EMMA は、モデル（図 3(a)）と複数の解釈結果（図 3(c), (g)）から構成される。モデルの記述に関しては、現在、様々な記述方法(XForms[9], XMLSchema[10], RDF[11]等)が検討されている。

この記述例では、「りんごの画像をポインティング」と「みかんの画像をポインティング」という、認識結果(座標位置)に対応する 2 つの解釈結果(図 3(c), (g))がある。解釈結果を表す<interpretation>は属性 confidence を持ち、そこに解釈結果の信頼度を格納することになっている。この例では、それぞれ“90”，“10”の信頼度が格納されている。信頼度は、例えばポインティングされた座標位置と各画像オブジェクト（りんご、みかん）の距離等によって算出される。

<interpretation>は、入力情報をそのまま記述する<input>（図 3(d), (h)）と、入力情報を解釈した結果として生成される<instance>（図 3(e), (i)）から構成される。<instance>は、<data_model>内の各要素に値を埋めた形で生成される。この記述例では、各<instance>の<goods>（図 3(f), (j)）にはそれぞれ、“りんご”，“みかん”が埋められている。

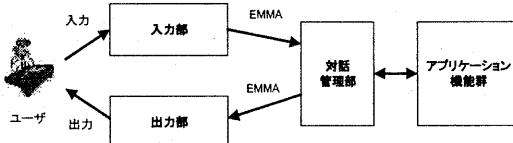


図 1 MMI フレームワークの主要コンポーネント

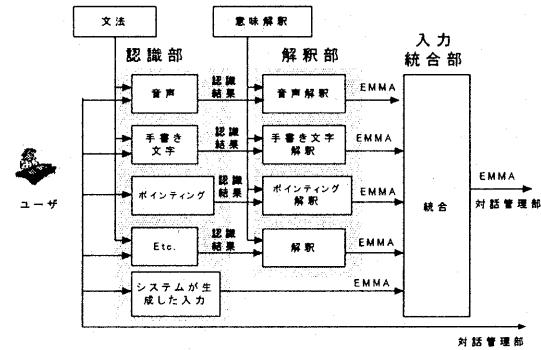


図 2 入力部を構成するサブコンポーネント

表 1 入力情報

ユーザの入力	ポインティング： りんごの画像に近い場所。少し離れた所にみかんの画像がある。
	音声： 「これを 3 個ください」
ポインティング認識エンジンの認識結果	(x, y) = (34, 58)
音声認識エンジンの認識結果	“これを 3 こください” (信頼度 90%)

```

<result>
  <data_model>----- (a)
    <order>----- (b)
      <num></num>
      <goods></goods>
    </order>
  </data_model>
  <interpretation confidence="90">----- (c)
    <input mode="pointing" timestamp="YYY">--- (d)
      <x-coordinate="34"/>
      <y-coordinate="58"/>
    </input>
    <instance>----- (e)
      <order>
        <num></num>
        <goods>りんご</goods>----- (f)
      </order>
    </instance>
  </interpretation>
  <interpretation confidence="10">----- (g)
    <input mode="pointing" timestamp="YYY">--- (h)
      <x-coordinate="34"/>
      <y-coordinate="58"/>
    </input>
    <instance>----- (i)
      <order>
        <num></num>
        <goods>みかん</goods>----- (j)
      </order>
    </instance>
  </interpretation>
</result>

```

図 3 ポインティング解釈部が生成した EMMA

```

<result>
  <data_model>
    <order>
      <num></num>
      <goods></goods>
    </order>
  </data_model>
  <interpretation confidence="90">
    <input mode="speech" timestamp="XXX">
      これを 3 こください
    </input>
    <instance>
      <order>
        <num>3</num>
        <goods>これ</goods>
      </order>
    </instance>
  </interpretation>
</result>

```

図 4 音声解釈部が生成した EMMA

5. MMI 記述言語の現状比較

MMI を記述するために、様々な言語が提案されている。以下では、著者らが提案・検討している MMI 記述言語 XISL と、これらの言語を比較しながら紹介する。

5.1. XISL(:Extensible Interaction Scenario Language)

XISL は、MMI のシナリオを記述するための XML ベースの言語であり、以下に示す 3 つの特徴がある。ここでは、それぞれの特徴と、要求仕様の関係について述べる。

1) 利用モダリティの追加が容易である。

XISL では、各モダリティによる入出力を、`<input>` 及び `<output>` を用いて指定する。`<input>` 及び `<output>` 要素の値の詳細（モダリティの種類、イベント、及び文法ファイルの指定等）は XISL の仕様から切り離し、外部の仕様としている。これにより、XISL は、利用するモダリティが増えても、その仕様を変更する必要はない。この特徴は、要求仕様の 1-A), 2-A), 2-E), 及び 3-C) をそれぞれ満たしている。

2) 複数のモダリティを、逐次的、同時的に組み合わせたマルチモーダル入出力を容易に記述できる。

XISL では上述の `<input>` 及び `<output>` を制御するためのタグに "seq"（逐次入出力）、"par"（同時に出入力）といった属性を用意している。これらの特徴は、要求仕様の 1-B), 1-C), 2-B), 2-D), 3-A), 及び 3-B) をそれぞれ満たしている。

3) 再利用性が高い。

XISL では、対話シナリオと、対話中に利用するコンテンツを切り離して記述するため、双方の再利用性が高いという特徴を持つ。この特徴は、要求仕様の 4-B) を満たしている。

以上に XISL の主な特徴と、それぞれの特徴が満たす要求仕様を示した。しかし、現仕様では SMIL2.0 の同期メカニズムを採用していないため、要求仕様 3-A) の、並列的な入力モダリティの組み合わせや、要求仕様 2-C) の、各モダリティ間の時間的な同期が記述できない。また再利用性が高いという特徴を持つ反面、XISL 製作者にとって、別ファイルに記述されたコンテンツを意識しながら、XISL を製作しなければならず、記述が複雑になるという問題を抱えている（こうした問題点に関しては、プロトタイプツールを提供しているが）。このため、要求仕様 1-D) についてはこれを充分満たしていると言えない。以上の課題への対応は 6 章で詳しく述べる。

5.2. SALT(:Speech Application Language Tags)

SALT[7] は、 XHTML、あるいは HTML 文章等に音声インターフェースを付加するためのタグセットである。SALT は、Web ページ上の音声によるフォームへの入力や、音声によるテキストの読み上げを可能にする。SALT のタグは、 XHTML 文書に埋め込む形で記述され、音声認識の文法を XHTML の `<input>` タグに対応づけることにより、音声によるフォーム入力を可能にしている。

SALT の利点は、 XHTML との親和性が高いことである。XHTML に慣れ親しんだユーザは比較的容易に SALT を扱える。これは、要求仕様の 1-D) を満たしている。またモダリティ間の同期に関しては、SMIL の同期モジュールもしくはスクリプトで記述するため、同期・統合に関する要求仕様も満たしている。

SALT の要求仕様に対する問題点は、 XHTML に SALT を埋め込むため、コンテンツとインタラクションが同一文書に混在することである。したがって要求

仕様の 4-B) は満たしていない。また SALT では、モダリティとして主に音声の追加を想定しているため、複数のモダリティに関する要求仕様 1-A), 2-A), 2-E), 及び 3-C) を満たしているとは言い難い。

5.3. XHTML+Voice

Voice は、音声対話記述言語である VoiceXML[12] を示している。VoiceXML は電話による音声対話を対象とした対話シナリオ記述言語であり、対話制御に関しては優れた記述力を持つ。しかしながら、VoiceXML は電話による対話を対象としているため、扱えるモダリティが、音声、DTMF キーなどに限定されている。そのため、MMI 記述言語としては、不十分である。これに対して、画面操作を融合させることにより、マルチモーダル対話を記述可能にするアプローチの 1 つが XHTML+Voice[8] である。

XHTML+Voice は、 XHTML 文書に、VoiceXML2.0 のタグセットを埋め込むことにより、音声によるフォームへの入力などを可能にしている。XHTML の `<input>` から、VoiceXML2.0 で記述されたタグセットの id を、イベントハンドラーとして登録する形式になる。こうした点では、SALT に近いアプローチといえる。

SALT と異なる点は、SALT が `<input>` と文法ファイルを直接対応づけるのに対し、XHTML+Voice は `<input>` と VoiceXML2.0 のタグセットを対応づけるところにある。製作者にとって、VoiceXML2.0 の記述方法を習得しなければならないため、単純な音声対話なら、SALT の方が記述しやすい。しかしながら、VoiceXML2.0 を習得すれば、 XHTML+Voice の方が、複雑な音声対話のシナリオ（ユーザ主導あるいは、混合主導等）を容易に記述できる。一方、SALT では複雑な音声対話のシナリオを記述する場合、その制御をスクリプトに頼ることになり、製作者にとって記述が困難である。

XHTML+Voice の要求仕様に対する問題点は、SALT のアプローチと極めて近いため、SALT とほぼ同じになる。異なる点は、SALT がモダリティ間の同期メカニズムを SMIL2.0 に任せているのに対し、XHTML+Voice は、VoiceXML2.0 で記述することになるため、SMIL2.0 のような同期メカニズムを持たない点である。

5.4. 各言語の現状比較

前節に紹介した各言語を、要求仕様の観点から比較したものを作成した。表 2 では、特に重要と思われる要求仕様を比較項目とした。

表 2 MMI 記述言語として各アプローチの比較

要求仕様	XISL	SALT	XHTML+Voice
複数のモダリティやデバイスが扱える	○	△	△
入出力モダリティ間の同期・統合	△	○	△
記述の容易さ	△	△	△
再利用性の高さ	○	×	×

○ … 満たしている

△ … 一部満たしている

× … 満たしていない

6. 要求仕様に対する XISL の問題点と対処方法

6.1. 要求仕様に対する現状の問題点

表 2 に示したように、XISL は要求仕様に対し、以下の 2 つの点が不十分である。

- 1) 記述の容易さ。
- 2) 入出力モダリティ間の同期・統合。

(SMIL2.0 の同期のメカニズムが扱えない。)

1)の要求仕様に対する対応としては、著者らの研究グループで、XISL ドキュメントの作成を支援するプロトタイプ「XISL ドキュメントツール」を開発している。ツールを提供することにより、XISL を容易に記述することが可能になる。ツールの詳細については文献[13]を参照されたい。次節では、2)への取り組みを説明する。

6.2. SMIL2.0 (Synchronized Multimedia Integration Language) の同期メカニズム

SMIL2.0 は、マルチメディア出力（オーディオ、動画、画像等）のストリーミングを制御する言語である。SMIL2.0 は、XISL で記述できる同時的、逐次的な出力に加えて、各種イベントを契機とした出力や、時間制御（例えば、アニメーションを出力開始して 5 秒後に音声出力を始める等）を記述することができる。

SMIL2.0 は各出力メディア間の同期に関して優れた記述力を持つおり、3.4 節でも紹介したように、この同期メカニズムは、要求仕様の 1 つになっている。

SMIL2.0 の同期メカニズムは、SMIL2.0 の 10 の機能に分類されるモジュールのうち、「タイミング及び同期モジュール」に集約されている。「タイミング及び同期モジュール」では、各メディア間の同期を可能にする様々な要素、属性、属性値が定義されている。

6.3. SMIL2.0 への対応

XISL では、統合・同期に関する要求仕様を満たすために、SMIL2.0 の「タイミング及び同期モジュール」を取り込むことにした。このモジュールを XISL に取り込むことにより、XISL で独自に行っていた入出力モダリティ間の同期に関する記述を、全て SMIL による記述に置き換えることができる。

図 5 に、SMIL に対応した XISL の記述例を示す。これは、「5 秒後 BGM を再生します」という音声合成を出力した後、5 秒後にオーディオファイルを再生する例である。`<smil:par>` (図 5(a))、及び`<output>`の属性`smil:begin` (図 5(c)) が SMIL のタグセットである。この例では、2 つの`<output>` (図 5(b), (d)) が`<smil:par>`で囲まれているため、各`<output>`は並列して実行される。1 つ目の`<output>` (図 5(b)) は、属性`smil:begin`による開始のタイミングが指定されていないため、即座に実行される。2 つ目の`<output>`は、属性`smil:begin` (図 5(c)) で、"notice.end+5s" と指定されている。`"notice"`は、1 つ目の`<output>`の id 値であり、"notice.end" は、1 つ目の`<output>`が終了したイベントを指す。つまり、"notice.end+5s" は、1 つ目の`<output>`が終了してから 5 秒後という意味を持つ。従って、2 つ目の`<output>` (図 5(d)) は、1 つ目の`<output>`が終了してから 5 秒後に実行される。

この記述例は、XISL では記述できなかった各モダリティ間の時間的な同期を記述できることを示している。この他にも、イベント（要素の開始、終了等）を契機とした同期、及びモダリティの優先度による同期など SMIL2.0 と同レベルの同期制御が記述可能になった。

```

<smil:par> ----- (a)
  <output id="notice" type="TTS" event="speech"> - (b)
    <param>
      5 秒後、BGM を再生します
    </param>
  </output>
  <output smil:begin="notice.end+5s" ----- (c)
    type="media" event="play"> ----- (d)
    <param name="uri">
      BGM.mp3
    </param>
  </output>
</smil:par>

```

図 5 XISL(SMIL 対応)の記述例

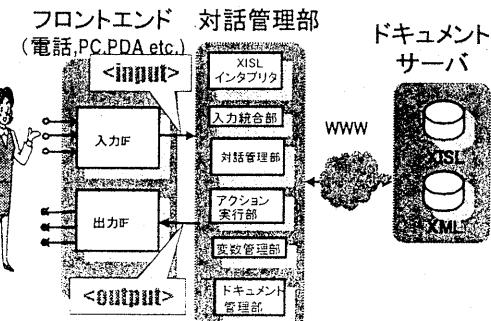


図 6 XISL 実行システム

7. MMI フレームワークへの対応検討

7.1. EMMA への対応

MMI フレームワークに対応するため、XISL 実行システム[14]と MMI フレームワークとを比較し、XISL の仕様を改良することを検討する。

XISL 実行システムは、XISL で記述された MMI シナリオを解釈・実行するシステムである(図 6)。XISL 実行システムは、端末ごとの入出力インターフェースを処理するフロントエンド、XISL を解釈・実行する対話管理部、XISL 等の XML コンテンツを管理するドキュメントサーバから構成される。XISL 実行システムの対話管理部は、フロントエンドから送られてくる、ユーザからの単独モダリティによる入力(`<input>`)を、XISL の記述に従ってマルチモーダル入力として統合し、それに対するユーザへの出力(`<output>`)をフロントエンドに送信している。

これに対し、図 1 に示した MMI フレームワークにおける対話管理部では、入力統合解釈部から EMMA(マルチモーダル入力の解釈結果)を受け、出力部に EMMA(マルチモーダル出力)を送る構成になっている。従って、XISL 実行システムを、MMI フレームワークに対応させるには、XISL のマルチモーダル入出力の記述部分を、`<input>`や`<output>`といった各入出力モダリティの指定から、EMMA 形式による指定に変更する必要がある。

図 7 に、EMMA 形式でマルチモーダル入出力の指定を行う場合の、XISL の 1 往復の対話記述案を示す。この記述案は、図 8 に示す EMMA を入力部から受け取る事を想定して、記述したものである。図 8 に示した EMMA の記述例は、図 3、図 4 で示した各解釈部が生成する EMMA をもとに、入力統合部が「りんごを 3 個注文する」という解釈で統合した結果、生成したものである。入力統合部は、モダリティ間の遅延や照応を扱う。統合方式に関しては、現在、ルールベースのも

のを検討中である。

XISLでは、1往復の対話を<exchange>（図7(a)）で記述する。1つの<exchange>内には、受け付けるEMMAを指定する<EMMA_input>（図7(b)）と、受け付けたEMMAに対するシステムの動作を記述する<action>（図7(c)）が含まれる。<EMMA_input>では、受け付けるEMMAのモデルを属性modelにより指定する。図7の記述案では、図8のモデルのルート要素名（図8(b)）である“order”を指定している。<action>内では、<EMMA_output>で出力するEMMAの指定を行う。この例では、注文の確認を行うための出力EMMA（図7(e)～(h)）を指定している。図7(f)の<num>要素の値である“order/num”や、図7(g)の<goods>要素の値である“order/goods”は、<EMMA_input>により受け付けたEMMAのインスタンス内の要素に対する参照を示している。例えば、図8のEMMAを受け付けた場合、“order/num”は図8(h)より参照される値は“3”であり、“order/goods”は図8(i)より参照される値は“りんご”である。

7.2. EMMAへ対応することの利点

現在のXISLは、マルチモーダル入出力の記述を個々のモダリティレベルで記述している。一方、EMMA形式を用いて入出力を指定するXISL（図7）は、モダリティに関する情報ではなく、“注文の受け付け”に対する“注文の確認”といった抽象的な記述になっている。これにより、端末毎のモダリティに依存しないXISLが記述でき、1つのXISLを様々な端末で再利用できるようになる。

```
<exchange>----- (a)
  <!-- 受け付ける EMMA モデルの指定 -->
  <!-- 注文の受け付け -->
  <EMMA_input
    model="order"/>----- (b)
<action>----- (c)
  <!-- 出力 EMMA の指定 -->
  <!-- 注文の確認 -->
  <EMMA_output>----- (d)
    <confirmation_of_order>----- (e)
      <num>order/num</num>----- (f)
      <goods>order/goods</goods>----- (g)
    </confirmation_of_order>----- (h)
  </EMMA_output>
</action>
</exchange>
```

図7 XISLの1往復の対話の記述案

```
<result>
  <data_model>----- (a)
    <order>----- (b)
      <num></num>
      <goods></goods>
    </order>
  </data_model>
  <interpretation confidence="90">----- (c)
    <input mode="pointing" timestamp="YYY">----- (d)
      <x-coordinate="34"/>
      <y-coordinate="58"/>
    </input>
    <input mode="speech" timestamp="XXX">----- (e)
      これを3 こください
    </input>
    <instance>----- (f)
      <order>----- (g)
        <num>3</num>----- (h)
        <goods>りんご</goods>----- (i)
      </order>
    </instance>
  </interpretation>
</result>
```

図8 入力部から受け取るEMMA

7.3. MMIフレームワークの課題

EMMAは、検討段階の仕様であるが、W3C-MMI-WG内で最も議論が活発に行われている事項の1つであり、MMIフレームワークの要となる技術である。しかしながら、EMMA仕様はマルチモーダル入出力データの表現形式を示すに留まっており、EMMAの生成や統合のメカニズムについては全く議論されていない。今後、著者らのグループでは、EMMAに対応したXISLの仕様を検討すると共に、EMMA生成・統合のアルゴリズム及び実行システムのアーキテクチャについても検討を進める予定である。

8.まとめ

本報告では、W3C-MMI-WGで提案・検討されている要求仕様、MMIフレームワークを紹介すると共に、それぞれに対するXISLの対応方法案を述べた。要求仕様に関しては、XISLはモダリティ間の同期・統合に関する要求を一部満たしていなかったため、この点をSMIL2.0の同期メカニズムを採用することで解決した。MMIフレームワークに関しては、現状のXISL実行システムのアーキテクチャとMMIフレームワークを比較した上で、XISLの仕様改良方法を検討し、マルチモーダル入出力の記述を、個別モダリティレベルの記述から、EMMAを用いた抽象的な記述へと変更した。この仕様変更により、XISLの記述が端末ごとのモダリティに依存しなくなり、多様な端末での再利用が、これまで以上に可能になった。以下に今後の課題を示す。

- (1) EMMAへの対応によるXISLの仕様改良。
- (2) EMMAの生成・統合に関するアルゴリズムの検討およびXISL実行システムのアーキテクチャ改良。
- (3) 今後予定されるW3C-MMI-WGの標準化作業に対するXISLの対応。

参考文献

- [1] <http://www.w3c.org/2002/mmi/>
- [2] <http://www.vox.tutkie.tut.ac.jp/XISL/XISL.html>
- [3] 中村有作、小林聰、桂田浩一、新田恒雄：“XISL：コンテンツ記述とインターラクション記述分離の試み” 情報処理学会第62回全国大会講演論文集（分冊4），pp.71-72(2001)。
- [4] 小林聰、中村有作、桂田浩一、山田博文、新田恒雄：“マルチモーダル対話記述言語XISLの提案”，情報処理学会研究報告 2001-SLP-37，pp.43-48(2001)。
- [5] 桂田浩一、中村有作、山田真、小林聰、山田博文、新田恒雄：“音声対話記述言語VoiceXMLとMMI記述言語XISLの比較”，情報処理学会研究報告 2001-SLP-38，pp.49-54 (2001)。
- [6] <http://www.w3.org/TR/smil20/>
- [7] <http://www.saltforum.org/>
- [8] <http://www.w3.org/TR/xhtml+voice/>
- [9] <http://www.w3.org/TR/xforms/>
- [10] <http://www.w3.org/XML/Schema/>
- [11] <http://www.w3.org/RDF/>
- [12] <http://www.w3.org/TR/voicexml20/>
- [13] 足立裕秋、桂田浩一、山田博文、新田恒雄：“MMIシステム構築のためのプロトタイピングツールの開発”，2002-SLP-43，pp.7-12(2002)。
- [14] 桂田浩一、大谷佳彦、中村有作、小林聰、山田真、新田恒雄：“多様な端末からのアクセスを可能にするMMIアーキテクチャ”，情報処理学会研究報告 2002-SLP-40，pp.51-56(2002)。