「情報処理学会論文誌」 第44巻 第11号別刷 平成15年11月発行

MMI記述言語XISLの提案

桂田浩一　中村有作　山田真
山田博文　小林聡　新田恒雄

社団法人 情報処理学会

MMI記述言語XISLの提案

桂田浩一[†] 中村有作[†,☆] 山田真[††]
山田博文[†] 小林聡[†] 新田恒雄[†]

本論文ではマルチモーダルインタラクション（MMI）記述言語XISLを提案する．XISLの目標は，(1) MMIで必要とされるモダリティの利用方法・対話の制御を記述可能にすること，(2) モダリティの拡張性を高めることである．これらを実現するために，XISLでは，(1) VoiceXML，SMILといった従来言語を参考に，モダリティの利用方法および対話制御の諸概念や命令を導入し，(2) 入出力モダリティに関する記述に自由度を持たせている．本論文ではXISLの概略を説明するとともに，PC上に実装したXISLの実行システム，およびプロトタイプとして試作したオンラインショッピングアプリケーションについて述べる．またXISLを他の言語と比較することにより，XISLのMMI記述言語としての特徴を明らかにする．

Proposal of MMI Description Language XISL

Kouichi Katsurada,[†] Yusaku Nakamura,[†] Makoto Yamada,[††]
Hirobumi Yamada,[†] Satoshi Kobayashi[†] and Tsuneo Nitta[†]

This paper provides a multimodal interaction (MMI) description language XISL. XISL aims to be a language satisfying the following conditions: (1) it has enough power to describe MMI scenarios, (2) it has extensibility of input/output modalities. For this purpose, (1) XISL prepares a lot of commands and structures used in previous languages such as VoiceXML and SMIL, and (2) XISL has flexibility of describing input/output modalities. In this paper, we outline the specification of XISL, and show its interpreter and an application implemented on PC. Moreover, we clarify advantages of XISL by comparing it with other MMI description languages.

1. はじめに

Webアクセスにおけるマルチモーダル対話（MMI）の利用とその標準化に関する議論が，近年さかんに行われている．W3Cでは2002年2月にMMIワーキンググループ[1]が結成され，MMI記述言語およびその実行システムに関する要求仕様[2]が公開される等，活発な議論が始まっている．また，その他の機関でも独自にMMIの記述法が検討されている[3)~6)]．MMIとは，マルチモーダル，すなわち，複数の入出力モダリティを利用したユーザとシステムの間の対話を指し，その対話を記述するための言語をMMI記述言語という．MMI記述言語には最低限，モダリティの利用方法，および対話制御に関する記述力が求められるが，同時に，今後の技術進歩を考えると，新規モダリティを容易に追加できる等，拡張性に秀でた柔軟性の高い言語であることも重要と考えられる．W3Cが公開した要求仕様においても，モダリティの利用方法，対話制御に関する記述力に加え，モダリティの拡張性がMMI記述言語の満たすべき必須項目の1つとしてあげられている．しかしながら，これまでに提案されてきたMMI記述言語[3)~6)]は，利用可能なモダリティを音声，ポインティングといった特定のものに限定しているのが現状である．

そこで本論文ではモダリティの利用方法と対話制御に関する記述力に加えて，モダリティ拡張性が優れたMMI記述言語XISL（eXtensible Interaction Scenario Language）[7),8)]を提案する．XISLはXML構文に基づく言語であり，W3Cの要求仕様に示されたモダリティおよび対話記述に関する項目に準拠した言語を目指している．まず，モダリティの利用方法と対話制御の記述に関する要求を満たすために，W3Cです

† 豊橋技術科学大学
Toyohashi University of Technology
†† 茨城工業高等専門学校
Ibaraki National College of Technology
☆ 現在，松下電器産業株式会社
Presently with Matsushita Electric Industrial Co., Ltd.

でに提案されている言語（VoiceXML[9]，SMIL[10]）を参考に，対話遷移，条件分岐といった対話の制御，および同時入出力/逐次入出力/択一入力といった複数モダリティの組合せ制御を導入した．またモダリティの拡張性に関する要求を満たすために入出力モダリティの記述に自由度を持たせ，言語仕様の変更なしに新規モダリティを追加できるよう設計した．以上の特徴を持つことにより，XISL は多様な端末を利用したシームレスな Web サービスの実現を可能にする．

本論文では，まず XISL の設計指針を述べたうえで，XISL の概略を説明する．続いて PC 上に実装した XISL の実行システム，およびプロトタイプとして試作したオンラインショッピングアプリケーションを紹介した後，XISL を他の言語と比較し，MMI 記述言語としての利点を明らかにする．

2. MMI 記述言語 XISL

2.1 XISL の設計指針

W3C が公開した要求仕様[2] には，モダリティおよび対話記述に関して以下の要求項目があげられている．

(1) 広範囲のマルチモーダルアプリケーションで利用できなければならない．
(2) モダリティに関する情報（パラメータ等）を記述できなければならない．
(3) 入力モダリティの組合せ（逐次的（sequential），一斉的（simultaneous），複合的（composite），選択的（supplementary），補完的（complementary）入力）を記述できなければならない．
(4) 出力モダリティの組合せ（逐次的（sequential），一斉的（simultaneous）出力）を記述できなければならない．
(5) 多様なモダリティ☆が利用でき，また拡張も可能でなければならない．

このほかにも，多言語対応，セキュリティやアクセシビリティ，入力統合・競合解消，意味解釈の検討に加えて，MMI システムのフレームワークや実行時の環境等に関する様々な要求が示されているが，本研究では特にモダリティおよび対話制御に関する上記項目に対応することを目指した．

これらの項目を満たす言語を設計するにあたって，W3C で提案されている従来の言語を参考にした．これは従来言語との乖離を少なくすることにより，ユーザが言語学習をするための負荷を軽減できると考えたからである．これまで対話記述言語として VoiceXML[9] が，またメディア同期を扱う言語として SMIL[10] がそれぞれ提案されている．VoiceXML は対話制御に必要な条件分岐や算術演算といった基本的な命令セット，および対話の階層（セッション，アプリケーション等）や遷移といった概念を網羅しており，広範囲のアプリケーションで利用可能である．これらは上述の項目 (1) を満たしているため，同様の命令や概念を XISL でも導入することにした．しかしながら VoiceXML では，電話音声もしくは DTMF という限定されたモダリティを対象としており，入出力モダリティを組み合わせた対話を記述することができない．これに対して，マルチメディアのストリーミングを扱う SMIL は複数の出力の同期処理を記述できる．そこで SMIL を参考にして入力，出力の同期処理を記述することで，上述の項目 (3) および (4) に対応した．

さらに，VoiceXML および SMIL で対応できない項目 (2)，(5) に対処するために，モダリティの利用方法を記述できるようにするとともに，モダリティ依存の記述と非依存の記述が明確に分離されるよう設計した．そのうえで，モダリティ依存の記述を XISL の仕様外とし，新たなモダリティを導入する際，言語全体を修正しなければならないという不便を解消した．

以上の設計指針に基づき XISL を設計した．次節以降に XISL の概要を説明する．

2.2 XISL の命令セットと対話の階層

広範囲のマルチモーダルアプリケーションで XISL を利用できるように，VoiceXML で用いられている様々な命令セットや対話階層の概念をほぼそのまま導入した．以下にその概要を示す．なお，XISL のタグ，対話の階層の詳細とその記述例については文献 11) および 12) の web サイトを参照されたい．

2.2.1 命令セット

XISL は XML ベースの言語であり，図 **1** に示すようなタグの階層構造を持つ．`<xisl>`，`<body>`，`<dialog>`，`<exchange>` はそれぞれ文書全体，対話全体，1 組の対話，1 ターンの対話を表している．このように対話の各単位を階層構造にすることで，対話の内容を把握しやすくした．`<dialog>`，`<exchange>` は，それぞれ VoiceXML の `<form>`，`<field>` に相当するが，対話の記述であることを明確にするため，VoiceXML とは異なるタグ名にした．`<dialog>` 内部には，`<exchange>` のほか，初期処理を表す `<begin>`，終了処理を表す `<end>` が記述可能である．`<begin>` と `<end>` の記述は任意であり，`<exchange>` は 0 個以上，複数記述してもよい．

☆ W3C が提示しているモダリティの詳細については，文献 2) を参照されたい．

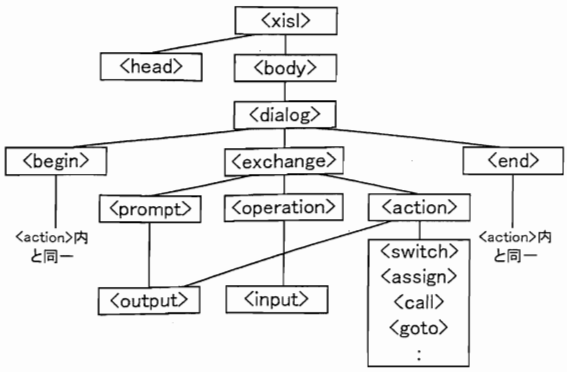

図 1 XISL のタグの階層構造
Fig. 1 Tag tree of XISL.

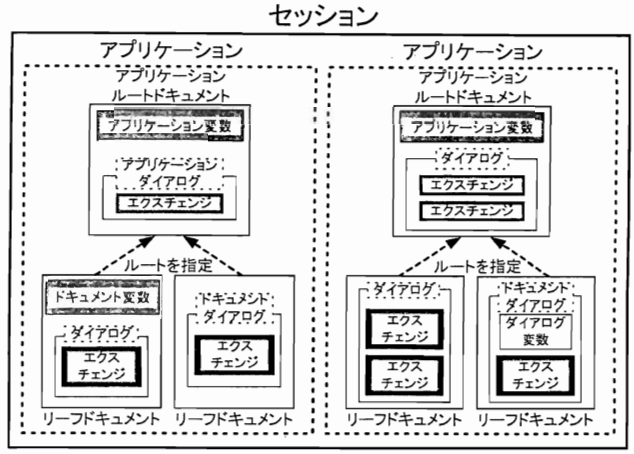

図 2 XISL ドキュメントの対話の階層
Fig. 2 Layers in XISL documents.

<exchange> は，ユーザへのプロンプトを表す <prompt>，ユーザのマルチモーダル入力を表す <operation>，その入力に対するシステムの動作を表す <action> から構成される．このうち <prompt>，<operation> は，それぞれ VoiceXML の <prompt>，<grammar> と同様の目的を持つ．相違点は，VoiceXML ではこれらタグ自体が単一の音声プロンプト，音声入力文法を表すのに対し，XISL ではマルチモーダル入出力の記述を可能にするため，<prompt> 内に複数の <output>（ユーザへの 1 出力を表すタグ）を，<operation> 内に複数の <input>（ユーザの 1 入力を表すタグ）を記述できるようにした点である．<action> は VoiceXML の <filled> に相当し，<output> をはじめとするシステムの動作に関する様々な要素の記述に用いられる．<prompt> の記述は任意とした．XISL では <action>，<begin>，および <end> の子要素として，対話の条件分岐のための <switch>，算術演算・代入のための <assign>，対話遷移のための <call> や <goto> をはじめとする多様なタグを用意している．これらのタグの多くは VoiceXML を参考に導入した．

2.2.2 対話の階層

VoiceXML では対話の階層的な管理により，各階層内で有効な変数や対話を設定可能にしている．XISL においても同様の設定を可能にするため，対話の階層を導入した．XISL で導入した対話の階層はセッション，アプリケーション，ドキュメント，ダイアログ，エクスチェンジの 5 つである．図 **2** に対話の階層構造を示す．セッションはユーザがシステムとのインタラクションを開始してから終了するまでに実行したすべての XISL 文書群から構成される．アプリケーションは 1 つのタスクを遂行するための XISL 文書群であり，1 つのアプリケーションルートドキュメントと 0 個以上のリーフドキュメント（リーフドキュメントでは <xisl> タグの属性 application にアプリケーションルートドキュメントを指定する）から構成される．ドキュメント，ダイアログ，エクスチェンジは，それぞれ 1 つの XISL 文書，1 つの <dialog>，1 つの <exchange> から構成される．

アプリケーション，ドキュメント，ダイアログ，エクスチェンジでは，各階層内で有効な変数を宣言できる．またアプリケーション，ドキュメントでは，それぞれの階層内で有効な <dialog>（それぞれアプリケーションレベルの <dialog>，ドキュメントレベルの <dialog> と呼ぶ）を設定できる．アプリケーションレベルの <dialog>，およびドキュメントレベルの <dialog> を設定するには，それぞれアプリケーションルートドキュメント内の <dialog> の属性 scope，リーフドキュメント内の <dialog> の属性 scope の値を "document" に設定すればよい．これらのレベルの <dialog> は，それぞれ当該アプリケーション，XISL 文書内の別の <dialog> を実行中にも有効となる．つまり，別 <dialog> 実行時における割込み対話として利用可能になる．<dialog> の属性 scope の値を "dialog" に設定した場合，あるいは属性 scope の値を設定しなかった場合，その <dialog> は明示的に呼ばれたときにのみ有効となる．このような <dialog> をダイアログレベルの <dialog> と呼ぶ．このように各階層内で有効な変数，<dialog> を設定可能にすることで，スコープの異なる変数や割込み対話を用いた複雑な対話を記述可能にしている．

2.3 XISL における同期制御

XISL では，SMIL で用いられている同期制御と同様の仕組みを導入することで，入出力の組合せを実現する．1 ターンのやりとりを単位とする同期制御と，個々の入出力を単位とする同期制御を実現するため，

```
<dialog id="order" scope="dialog">
      :
  <exchange>
    <operation comb="alt"> .....(a)
      <input type="touch" event="click"
          match="[item:=@id]"/> .....(b)
      <input type="speech" event="recognize"
          match="./grammar.txt#item"
          return="item"/> .....(c)
      <par_input> .....(d)
        <input type="touch" event="click"
            match="[item:=@id]"/> .....(e)
        <input type="speech" event="recognize"
            match="./grammar.txt#number"
            return="number"/> .....(f)
      </par_input>
    </operation>
    <action> .....(g)
      <output type="tts" event="speech">
        <![CDATA[ <param name="text">
            You ordered <value expr="item">.
          </param> ]]>
      </output>
          :
      <goto next="number.xisl" namelist="number">
    </action>
  </exchange>
      :
</dialog>
```

図 3 XISL 要素の同期制御
Fig. 3 Synchronization of XISL elements.

<exchange>, <input>, <output> の同期制御を可能にした．

まず <exchange> の実行を制御するために，<dialog> に属性 comb を導入した．comb の値が "par"（同時的）のときは順不同にすべて，"seq"（逐次的）のときは上から順に <dialog> 内部の <exchange> が実行される．また，値が "alt"（択一的）のときは，まず各 <exchange> 内のプロンプトがすべて出力され，その後，内部の <operation> が最初に満たされた <exchange> 内の <action> が実行される．すなわち，ユーザの入力に応じて <exchange> が択一的に実行される．<dialog> 内部の一部の <exchange> のみを制御したい場合には，対象となる <exchange> を <par_exchange>, <seq_exchange>, <alt_exchange> といったタグで括る．

同様に，<input> による入力待ち受けを制御するために，<operation> の属性 comb，および <par_input>, <seq_input>, <alt_input> の各タグを導入した．<output> に関しては，2 種類のタグ <par_output>，および <seq_output> を導入した．

図 3 に同期制御の記述例を示す．この例では，(a) の <operation> の属性 comb に alt が指定されているため，(b)，(c)，(d) の <input>，<par_input> が択一的に待ち受けられる．(b) と (c) の <input> については，それぞれ入力が受付けられた時点で (g) の <action> が実行される．一方，(d) の <par_input> については，その内部の (e)，(f) の <input> が入力された場合に (g) の <action> が実行される．なお，この例では (b) と (e) の <input> が同一である．この場合，この <input> を受付けてから一定時間待機し，その後 (f) の入力がなければ当該入力を (b) と判断する．(f) の入力があれば (e) と判断する．

ここで，要求仕様の各項目と対比する．2.1 節の (3) にあげたように，W3C では入力の組合せ処理に関して，(i) 逐次的（連続的な入力），(ii) 一斉的（別々に解釈されるべき同時入力），(iii) 複合的（統合解釈されるべき同時入力），(iv) 選択的（ユーザが利用モダリティを選択できる入力），(v) 補完的（複数のモダリティを補完的に利用する入力）な入力の 5 項目を各々記述可能にするよう求めている．XISL では，<input> の逐次的制御（seq）によって項目 (i) が，同時的制御（par）によって項目 (ii) および項目 (iii) が，択一的制御（alt）によって項目 (iv) が，択一的制御（alt）と同時的制御（par）の併用によって項目 (v) が記述できる．出力の組合せに関しても，2.1 節の項目 (4) にあげた逐次的，一斉的出力は <output> の逐次的（seq），同時的制御（par）によって記述できる．

2.4 入出力の記述

XISL では，インタラクションのみをコンテンツから分離して記述する．これは，XISL を純粋なインタラクション記述言語とすることで，対話シナリオを理解しやすくするためである．たとえば音声認識文法や web ページ等のコンテンツは，別ファイルに記述したうえで，URL を指定することによって利用（有効化，表示，切替え等）をする．こうした入出力関係のパラメータやファイルの指定は <input>, <output> 内で行う．

<input> にはユーザが用いる入力モダリティ等を指定するために type, event, match, return の属性を用意している．属性 type には入力のモダリティが，属性 event には入力イベントが，属性 match には入力対象のオブジェクト（音声認識文法や他の XML 文書内の要素等）が，属性 return には入力内容を格納するための変数が，それぞれ記述される．また <input> の子要素として <param> を記述することができ，入力デバイスに与えるパラメータを記述できる．

一方，<output> は 2 つの属性 type と event を持ち，内容として CDATA セクションを持つ．属性

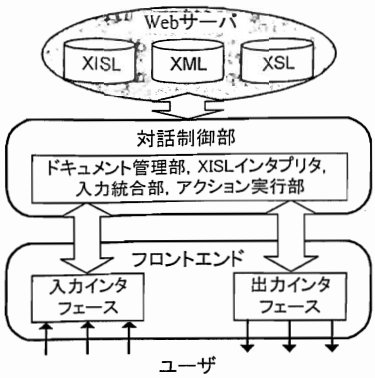

図 4 XISL 実行システム
Fig. 4 An XISL execution system.

type には出力のモダリティが，属性 event には出力イベントが，CDATA セクション内には出力の内容（web ページの URL や音声合成のテキスト等）が記述される．CDATA セクション内の要素 <value> は変数の値を埋め込むために用いる．

XISL では以上に述べた <input>，<output>，および <param> の導入によって 2.1 節の項目 (2) に対応した．さらに項目 (5) に対応するために，XISL ではこれらの要素の記述方法に自由度を持たせた．すなわち，<input>，<output>，および <param> の属性のみを規定し，属性値の記法や，内容の詳細については XISL の仕様外とした．このことにより，モダリティの拡張や修正があった場合にも，XISL は仕様変更なしに多様な端末環境に対応することができる．<input>，<output>，<param> の具体的な記述例は，次の 3.1 節で紹介する．

3. PC 上での実装

3.1 入出力の仕様

図 4 に示す各モジュールで構成した XISL 実行システムを PC 上に実装するとともに，<input> および <output> の仕様を策定した．実行環境は次のとおりである．

- 計算機：PC
- 入力デバイス：マウス，キーボード，マイク
- 出力デバイス：ディスプレイ，スピーカ
- OS，ソフトウェア☆：Windows 2000，Microsoft Agent2.0，東芝 LaLaVoice 2001，Microsoft Japanese Recognizer v5.1，Microsoft MSXML 4.0，Microsoft InternetExplorer6

図 4 に示した実行システムでは，XISL をはじめとする各文書は Web サーバに保持され，XISL は対話制御部で解釈・実行される．フロントエンドは入出力を制御するモジュールであり，音声認識や擬人化エージェント，映像の出力等を行う．このシステムでの <input> および <output> の仕様の一部を**表 1**，および**表 2** に示す．<input> には，表 1 に示すように，音声入力やポインティング入力等様々な入力モダリティに対応した記述ができる．また <output> には，表 2 に示すように，ウィンドウや合成音声，擬人化エージェントをはじめとする様々な出力モダリティを記述することができる．なお，<input> に関しては，現時点で <param> 要素を用いていない．

XISL は，上記の説明から分かるように，新規のモダリティを含む新しい端末環境に柔軟に対応できる．現在，我々は電話，および PDA を対象とした <input>，<output> の仕様を策定中であるが，このほかにも，たとえば <input> のモダリティとして人感センサ（type の属性値が "sensor"，return の変数に検出結果を格納）や GPS 情報（type の属性値が "GPS"，return の変数に位置情報を格納）といった記述を行うことで，多様なモダリティを利用したインタラクションが記述可能になる．本節で紹介した PC 用システム，および上で触れた電話，PDA 用システムの <input>，<output> 仕様については，文献 12) の web サイトを

表 1 PC 上での <input> 仕様
Table 1 Specification of <input> on a PC terminal.

type	event	match	return
"speech"	"recognize"	文法ファイル名 #ルール名	文法規則と同名の変数の集合
"touch"	"click"	XHTML 要素の ID 属性値	x 座標 y 座標
	:	:	:
"key"	"press"	0-9,A-Z の一文字	押された文字
:	:	:	:

表 2 PC 上での <output> 仕様
Table 2 Specification of <output> on a PC terminal.

type	event	CDATA セクション内の要素<param>	
		name	値
"browser"	"navigate"	"id"	ブラウザ ID
		"uri"	コンテンツの URI
		:	:
	:	:	:
"tts"	"speech"	"text"	発話テキスト
		:	:
"agent"	"speech"	"name"	エージェント名
		"text"	発話テキスト
	:	:	:
:	:	:	:

☆ Windows，Microsoft はマイクロソフト（株）の登録商標，LaLaVoice は（株）東芝の商標である．

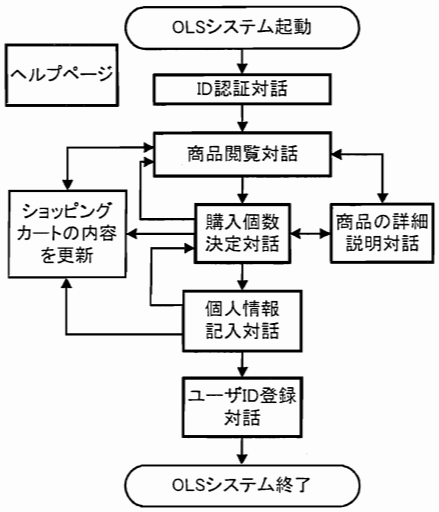

図 5 オンラインショッピングシステムの対話の流れ
Fig. 5 Dialog flow in an online shopping system.

参照されたい．

3.2 XISL によるアプリケーションの記述例

我々の研究グループでは，XISL の適用例として，オンラインショッピング（OLS）アプリケーションを開発している[13]．本節ではこのアプリケーションを取り上げ，XISL で記述した対話制御の実例を示す．

OLS のアプリケーションは，商品購入のための XISL 対話シナリオ，画面への表示内容と商品データを格納する XML コンテンツ，表示スタイルを決定する XSL スタイルシートから構成される．図 **5** にアプリケーションを構成する対話の流れ示す．また図 5 に示す対話のうち，ID 認証対話の一部を簡略化したものを図 **6** に示す．OLS アプリケーションの各対話の流れは，おおよそ図 6 の対話の流れと同様である．まず最初に図 6 (i) に示すように <begin> 内で各対話の初期処理を行うとともに，<output> によって画面を表示する（あるいは切り替える）．続いて図 6 (ii) に示すように <operation> 内部で音声，ポインティング等を用いた入力を待ち受けた後，図 6 (iii) のように <action> 内部で代入，演算，出力，条件分岐等の処理を行う．最後に図 6 (iv) のように <goto> もしくは <call> によって次の対話へ遷移する．

ヘルプ対話のように，任意の場所で呼び出し可能にすべき対話は，割込み対話として記述した．図 **7** にヘルプ対話の一部を示す．ヘルプ対話はアプリケーションレベルの対話として定義しており，アプリケーション内の任意の場所から呼び出すことが可能である．

OLS アプリケーションでは多くの対話中でマルチモーダルによる入力を可能にしている．一例として図 **8** のような画面が表示される商品閲覧対話でのマルチモーダル入力を説明する．商品閲覧対話では，図 8 中の (1) に示した購入ボタン，詳細説明ボタンをクリックすることにより，それぞれ購入個数決定対話，商品詳細説明対話へ遷移するが，同様の遷移は，商品の写真を直接クリックして「購入」あるいは「詳細説明」と発話したり，音声のみで「○○を購入」と発話することによっても可能である．2.3 節の図 3 に示した XISL はこの対話の記述例の一部で，<operation> 内部の <input> の待ち受けを “alt”，“par” で制御することにより，多様なモダリティを用いた入力を実現している．

```
<?xml version="1.0" encoding="Shift-JIS"?>
<!DOCTYPE xisl SYSTEM "xisl.dtd">
<xisl version="1.0" application="root.xisl">
  <head> ··· </head>
  <body>
    <dialog id="ID_Check" scope="dialog" comb="alt">
      <begin> .....(i)
          :
        <output type="browser" event="navigate">
          <![CDATA[
            <param name="uri">
              Cirtification.xml
            </param>
              :  ]]>
        </output>
      </begin>
      <exchange>
        <operation comb="alt"> .....(ii)
          <input type="touch" event="click"
              match="/certify_id/object/first_time"/>
          <input type="speech" event="recognize"
              match="./grammar.txt#first_time"/>
        </operation>
        <action> .....(iii)
          <assign name="user_name" expr="'NULL'"/>
          <assign name="user_id" expr="'new_user'"/>
          <goto next="./Top_Page.xisl#Top_Page"
              namelist="user_name user_id"/> ....(iv)
        </action>
      </exchange>
          :
    </dialog>
  </body>
</xisl>
```

図 6 ID 認証対話の一部の記述
Fig. 6 A part of ID-certification dialog.

4. 他言語との比較と考察

XISL の定量的な評価は現時点では困難なため，他言語との比較によってその特徴と課題を明らかにする．本章では比較対象言語として SALT，XHTML+Voice，UIMS 記述言語，およびその他の言語を取り上げる．

4.1 SALT との比較

SALT[3),4)] は XHTML 文書等に対して音声インタ

```
<?xml version="1.0" encoding="Shift-JIS"?>
<!DOCTYPE xisl SYSTEM "xisl.dtd">
<xisl version="1.0">
    <head> ··· </head>
    <body>
      <dialog id="help" scope="document">
        <exchange>
          <operation>
            <input type="speech" event="recognize"
                match="./grammar.txt#help"/>
          </operation>
          <action>
            <output type="tts" event="speech">
              <![CDATA[ <param name="text">
                What's the matter?
                </param> ]]>
            </output>
          </action>
        </exchange>
            :
      </dialog>
    </body>
</xisl>
```

図 7 アプリケーションルートでのヘルプ対話の記述
Fig. 7 Discription of help dialog in an application root.

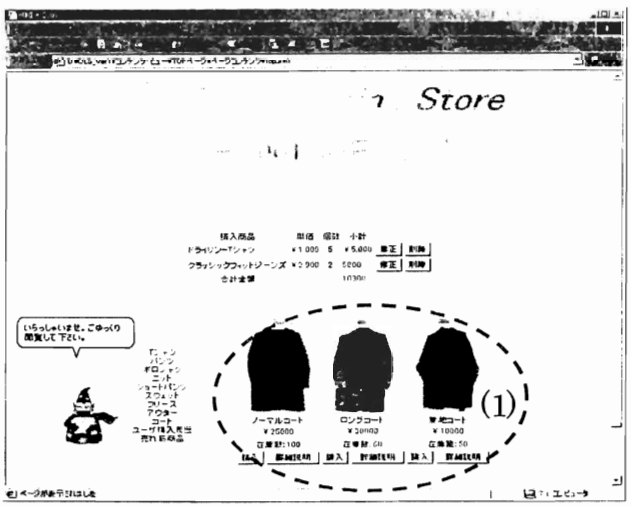

図 8 オンラインショッピングシステムの画面例
Fig. 8 A screen of an online shopping system.

フェースを付加するためのタグセットであり，web ページ上での音声によるフォームの埋め込みや，音声によるテキスト読み上げを可能にする．最大の特徴は，現状の技術，特に XHTML と親和性が高いことである．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> タグにバインドすることにより，音声によるフォーム入力を可能にしている．したがって既存の XHTML 文書が有効に利用でき，最小限の修正で音声インタフェースを付加することが可能である．

一方，SALT では対話制御の記述ができず，制御は SMIL もしくはスクリプトで記述することになる．また，XHTML に SALT を埋め込む方法では，コンテンツとインタラクションが同一文書に混在する．さらに 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．これに対して，XISL は XISL 自身で対話を制御し，インタラクションとコンテンツを分離するとともに，最初から多様なモダリティの利用を想定している点が異なる．

4.2 XHTML+Voice との比較

XHTML+Voice[5] は，音声言語のみを取り扱う VoiceXML に画面操作を融合させ，MMI を記述可能にするアプローチの 1 つである．XHTML 文書に VoiceXML 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等を可能にしている．XHTML 文書に埋め込むという点で SALT と同様のアプローチと見なせるが，VoiceXML は SALT に比べて比較的複雑な対話シナリオ（ユーザ主導，混合主導対話等）が記述できるため，対話シナリオによる入出力制御が必要な場合には SALT と比べて記述が容易である．

しかしながら，XHTML+Voice では SALT と同様，音声とポインティング等，限られたモダリティを想定しているのに対し，XISL ではモダリティの拡張性を考慮し，モダリティの記述に自由度を持たせている点が異なる．また VoiceXML では，スロットフィリング（ユーザが項目を埋めていく）の考えに基づいた対話制御を行っており，同時的，逐次的，択一的なシナリオの進行を記述するための明示的な方法がないため，対話制御に関しては XISL の方が記述しやすいと考える．

4.3 UIMS 記述言語との比較

XISL は UIMS（User Interface Management System）記述言語（ユーザと計算機のやりとりを記述する言語）の一種と見なすことができる．UIMS の研究は古くからさかんであり，その記述言語についても古くは Sassafras-UIMS[14] の ERL から，近年では UIML[15] まで，様々なものが提案されている．ここでは Sassafras の ERL，および UIML を取り上げ，我々の提案と比較する．

Sassafras は，多様なデバイスを用いたアプリケーションを開発する際の，ラピッドプロトタイピングを行うための UIMS である．実行時のシステムは図 4 に示した XISL 実行システムの構成とほぼ同じで，対話記述を保持する ERL モジュール（XISL 実行システムの対話制御部に相当），入出力を取り扱うインタラクションモジュール（フロントエンドに相当），アプリケーション固有の各種モジュール（Web サーバ内の各種文書，cgi 等に相当），およびこれら全体を

管理するクラスタコントローラから構成されている．Sassafras は web アプリケーションを前提としていない点が XISL 実行システムと異なるが，デバイスの拡張性が高い点等，我々のシステムとの共通点が多い．対話記述言語として用いられている ERL は，イベントに対するアクションを if-then 形式のルールで宣言的に記述する言語である．宣言的な記述は，保守性，再利用性に優れているという利点がある一方で，.文献 14) 内でも述べられているように，対話全体の流れが把握し難いという問題もある．

UIML は画面の構成や，ユーザ入力の受付け方法といったユーザインタフェースを記述するための言語である．端末の拡張性が高く設計されており，XISL と同様，多様な端末での利用が可能であるいう特徴を持つ．UIML は個々の入出力の記述，すなわち XISL における <input>，<output> の記述の規定に主眼をおいた言語である．これに対して XISL は入出力の流れ，すなわち対話シナリオを記述するための言語であり，UIML とは目的が異なる．

4.4 その他の言語との比較

MILES[6] は MMI システムの対話モデルを記述するために Prolog を拡張した言語である．対話は一種の if-then ルールの集合として宣言的に記述され，その内部に割込み対話，スクリプト，マルチモーダル入出力の時間制御等を記述できる．XISL では，現時点でスクリプトを記述できないため，XISL と比べて広範囲のアプリケーションでの利用が可能と考えられる．MILES は，宣言的な対話記述を行っているため，先に述べた ERL と同様，保守性，再利用性に秀でている反面，対話全体の流れが把握し難いという問題がある．また，MILES は SALT や XHTML+Voice と同様，限定されたモダリティの利用のみを想定しているため，XISL の方がこの点拡張性が高い．

MPML[16),17)] は擬人化エージェントを用いたプレゼンテーションのための言語であり，エージェントの移動，発話，複数エージェントの同期といった制御を詳細に記述できる．また音声入力によるエージェントの制御も記述可能で，効果的なプレゼンテーションに留意した設計がなされている．XISL においても，3.2 節で試作したシステムでは <output> の仕様として擬人化エージェントを取り扱うことができるが，現状では MPML に比べて記述力が十分でない．一方，XISL は擬人化エージェントに限らず様々な入出力モダリティを取り扱うことができ，割込み対話等も記述可能である．

5. おわりに

本論文では XML ベースの MMI 記述言語 XISL を提案するとともに，実装を通してその動作を検証し，また他の言語との比較により XISL の利点を明らかにした．XISL は以下の特徴を持つ．

- MMI 記述言語に必要とされる対話記述能力を持つ．
- モダリティの拡張性が高い．

本論文では，これらを実現するために行った言語仕様の策定と実装例を中心に述べた．以下にその骨子をまとめて述べる．

まず，上述の第 1 項目を満たすために，VoiceXML および SMIL を参考に，<if>，<switch> といった対話制御に関するタグ，par，seq，alt といった対話および入出力の同期処理に関するタグと属性，対話階層の導入による変数のスコープ管理，割込み対話の設定等，様々な記述を可能にした．続いて，上述の第 2 項目を満たすために，入出力を記述する <input> と <output> に関して，その属性と内容の種類のみを規定し，属性値の記法や内容の詳細については端末やモダリティごとに自由に規定できるよう設計した．これにより，XISL は表現力を持つ拡張性の高い MMI 記述言語となった．

今後の課題としては，まず XISL を用いた MMI システムの汎用アーキテクチャを設計し，今回実装した PC 以外の多様な端末で実用性を検証することがあげられる．具体的には，3.1 節で述べた電話，PDA 等を対象として検証を進めている．また XISL では SMIL の同期制御機構のうち 2.1 節にあげた項目への対応に必要な部分のみを導入しているため，時間制御等を行う際の記述力が十分でない．さらに XISL 自体の評価についても，本論文では他言語との比較という非定量的な方法で行った．今後は，SMIL の同期制御機構を導入するとともに，開発工程等に関する調査によって定量的な評価を行い，XISL を本格的 MMI 記述のための基盤言語として整備したい．

なお，本研究の一部は文部科学省 21 世紀 COE プログラム「インテリジェントヒューマンセンシング」，科学技術研究費若手研究（B）14780323，および電気通信普及財団の補助による．

参考文献

1) http://www.w3.org/2002/mmi/
2) http://www.w3.org/TR/mmi-reqs/
3) http://www.saltforum.org/

4) Wang, K.: SALT: A Spoken Language Interface for Web-based Multimodal Dialog Systems, *Proc. ICSLP'02*, pp.2241–2244 (2002).
5) http://www.w3.org/TR/xhtml+voice/
6) 秋葉友良，伊藤克亘：スクリプト言語を用いたマルチモーダル対話記述の試み，情報処理学会研究報告 98-SLP-23 98-HI-80，pp.1–6 (1998).
7) 中村有作，小林 聡，桂田浩一，新田恒雄：XISL：コンテンツ記述とインタ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).
8) Nitta, T., Katsurada, K., Yamada, H., Nakamura, Y. and Kobayashi, S.: XISL: An Attempt to Separate Multimodal Interaction from XML Contents, *Proc. EUROSPEECH2001*, pp.1197–1200 (2001).
9) http://www.w3.org/TR/voicexml20/
10) http://www.w3.org/TR/smil20/
11) 桂田浩一，中村有作，山田 真，小林 聡，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).
12) http://www.vox.tutkie.tut.ac.jp/XISL/XISL.html
13) 桂田浩一，小林剛典，中島将宏，中嶋真里子，山田博文，新田恒雄：多様な端末からのアクセスが可能な OLS システムの実装，情報処理学会研究報告 2003-SLP-45，pp.41–46 (2003).
14) Hill, R.D.: Supporting Concurrency, Communication, and Synchronization in Human-Computer Interaction — The Sassafras UIMS, *ACM Trans. Graphics*, Vol.5, No.3, pp.179–210 (1986).
15) http://www.UIML.org/
16) 筒井貴之，石塚 満：キャラクター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).
17) http://www.miv.t.u-tokyo.ac.jp/MPML/

(平成 15 年 4 月 10 日受付)
(平成 15 年 9 月 5 日採録)

桂田 浩一（正会員）

平成 7 年大阪大学基礎工学部情報工学科卒業．平成 12 年同大学院基礎工学研究科博士後期課程修了．同年豊橋技術科学大学工学研究科助手．博士（工学）．マルチモーダル対話，知識処理に関する研究に従事．AAAI，人工知能学会，日本音響学会，ヒューマンインタフェース学会各会員．

中村 有作

平成 13 年豊橋技術科学大学工学部知識情報工学課程卒業．平成 15 年同大学院工学研究科知識情報工学専攻修了．同年松下電器産業（株）入社．現在パナソニック AVC ネットワークス社に所属．パソコンの開発に従事．

山田 真

平成 12 年豊橋技術科学大学工学部知識情報工学課程卒業．平成 13 年同大学院工学研究科知識情報工学専攻中途退学．同年茨城工業高等専門学校技術職員．現在同高専技術支援センター所属．

山田 博文（正会員）

平成 5 年信州大学工学部情報工学科卒業．平成 7 年同大学院工学研究科博士前期課程情報工学専攻修了．平成 8 年同大学院博士後期課程システム開発工学専攻中途退学．同年豊橋技術科学大学工学部助手．博士（工学）．学習支援システム，パターン認識に関する研究に従事．電子情報通信学会会員．

小林 聡（正会員）

平成 3 年豊橋技術科学大学工学部情報工学課程卒業．平成 6 年同大学院工学研究科情報工学専攻修了．平成 11 年静岡大学大学院博士後期課程電子応用工学専攻単位取得退学．同年豊橋技術科学大学情報処理センター助手．博士（工学）．音声言語情報処理の研究に従事．電子情報通信学会，人工知能学会，日本音響学会各会員．

新田 恒雄（正会員）

昭和 44 年東北大学工学部電気工学科卒業．(株) 東芝総合研究所，マルチメディア技術研究所を経て，平成 10 年豊橋技術科学大学大学院工学研究科教授．工学博士．音声認識・合成・文字認識，マルチモーダル対話システム，および概念獲得の研究に従事．IEEE，電子情報通信学会，人工知能学会，日本音響学会等の会員．