

マルチモーダル対話システムのための階層的アーキテクチャの提案

新田恒雄^{*1} 桂田浩一^{*1} 荒木雅弘^{*2} 西本卓也^{*3} 甘粕哲郎^{*4} 川本真一^{*5}

^{*1}豊橋技術科学大学 ^{*2}京都工芸繊維大学 ^{*3}東京大学 ^{*4}NTT サイバースペース研究所 ^{*5}ATR

あらまし 本稿では、マルチモーダル対話システムに関する技術の標準化を目的とした情報処理学会情報規格調査会「音声入出力インタフェース委員会」における検討内容について、中間報告を行う。まず我々の提案するマルチモーダル対話システムの階層的アーキテクチャの得失を論じた後、ユースケースに基づく各階層の仕様案について説明する。提案アーキテクチャは、既存の記述言語や開発フレームワークと高い親和性を持つことで実用システムの開発を効率化するとともに、各コンポーネントの役割を明確にして独立性を高めることで研究用プラットフォームとしても機能することを目指している。

Proposal of a Hierarchical Architecture for Multimodal Interactive Systems

Tsuneo Nitta^{*1} Kouichi Katsurada^{*1} Masahiro Araki^{*2} Takuya Nishimoto^{*3} Tetsuo Amakasu^{*4}
Shinnichi Kawamoto^{*5}

^{*1}Toyohashi University of Technology ^{*2}Kyoto Institute of Technology

^{*3}The University of Tokyo ^{*4}NTT Cyber Space Labs. ^{*5}ATR

Abstract This paper reports an intermediate status of speech interface committee under ITSCJ (Information Technology Standards Commission of Japan). First, we discuss about pros and cons of hierarchical architecture of multimodal dialogue systems. Next, we explain the first draft of informative descriptions of each component based on use case investigation. The aim of the proposed architecture is to support practical system development by complying with the existing design language and development framework, and to function as a research platform by specifying the role of each component.

1. はじめに

マルチモーダル対話 (Multimodal Interaction; 以下 MMI) は、近未来のヒューマンインタフェース中核技術として携帯端末、カーナビ、情報家電からロボット応用まで実現への期待が大きい。いくつかの研究機関で先進的な MMI システム ([1],[2],[3]など) が開発されている一方で、各種モダリティの統合や分化の方法論が確立していないことや、様々なアプリケーション開発に利用できるフレームワークが存在しないことなどから、実用化への見通しが立っていないとはいえないのが現状である。

擬人化エージェントを用いた MMI システム開発のプラットフォームを目指した Galatea プロジェクト¹においても、音声認識・音声合成・顔画像合成など様々なコンポーネントが開発され、対話記述言語として XISL[4]や VoiceXML のマルチモーダル拡張版²がリリースされたが、アプリケーションの開発事例はあまり多くない。この原因として、実用システムや高度な研

究用システムの開発に適用できるアーキテクチャが提案できていないこと、そして高レベルのタスク記述から低レベルの入出力制御までの様々なニーズを満たす記述言語が策定されていないことがあげられる。

このような状況を踏まえ、情報処理学会情報規格調査会「音声入出力インタフェース委員会」では、MMI システムのユースケースの検討を出発点として、標準的なシステムアーキテクチャを提案することを目的として活動を行っている。提案アーキテクチャは、既存の記述言語や開発フレームワークとの高い親和性を持つことで実用システムの開発を効率化するとともに、各コンポーネントの役割を明確にして独立性を高めることで研究用プラットフォームとしても機能することを目指している。

以下、2章で提案する MMI アーキテクチャを概観しその得失を論じる。3章でシステムに対するユースケースを示し、そのユースケースを実現するために各階層でどのような機能が要求されるかを4章で説明する。5章で既存のアーキテクチャとの比較を行い、6章で今後の予定を述べる。

¹ <http://sourceforge.jp/projects/galatea/>

² <http://hil.t.u.tokyo.ac.jp/~nishig/galatea-doc/main.html>

2. MMI システムアーキテクチャ

一般にソフトウェア開発においては MVC の分離が推奨される。すなわち Model (アプリケーションが内部的に保持するデータモデル), View (ユーザに提示する画面表示や入力イベントなどの User Interface), Controller (Model と View を対応づけてユーザインタフェースを実現するロジック) を分離することで柔軟なシステム開発が可能になるという主張である。

我々は MMI システムにおいて、モダリティに依存する処理を集約すること、使用するモダリティの追加や変更を柔軟にすること、きめ細かいタイミングでの入出力の制御を行うことなどを、アーキテクチャと記述言語への要求として挙げた。そして MMI システムを MVC の観点から考えることでこれらを満たすことが容易になると考えた。

そこで既存の MMI 記述言語の検討などを踏まえて MMI システムの構成要素を分解した結果、MVC の観点を拡張した 6 階層の MMI システムアーキテクチャ (図 1) が提案された。各階層の内容は以下の通りである。

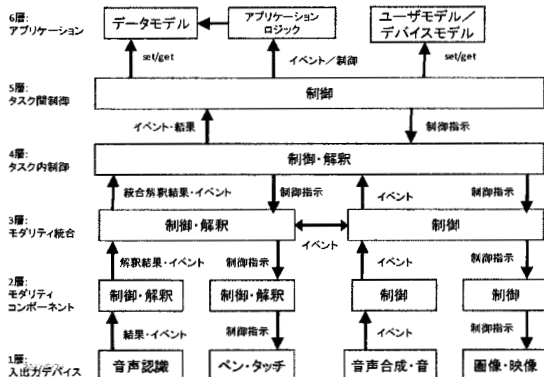


図 1 MMI システムのアーキテクチャ階層

第 1 層: 入出力デバイス層

それぞれのデバイスに依存する API やイベントなどが用いられる。

第 2 層: モダリティコンポーネント層

ここでは画面表示、音声入出力、擬人化エージェント制御など、個々のモダリティの制御を行う。入出力の取扱いは分離し、モダリティ間の連携は第 3 層を介して実現される。SVG³ の rect 要素、VoiceXML⁴ の prompt 要素と SSML⁵, SALT⁶ の listen 要素などに対応する。

なお、GUI におけるボタンなどの部品や、レベルメータや音声認識状態表示など、分解していくと複数の

モダリティから構成されるような部品については、第 2 層で定義された個別のモダリティを第 3 層で統合する方法と、第 3 層からは単一のモダリティに見えるようなモダリティ統合部品として第 2 層で定義する方法と、両方が考えられる (詳細は 4.1 で述べる)。

第 3 層: モダリティ統合層

ここでは入力の統合、出力の分化、入出力同期制御などを行う。逐次入力や同時入力の解釈、逐次出力や同時出力の同期、モダリティ拡張などが含まれる。

SMIL 2.1⁷ (出力), XISL (出力, 統合解釈) などに対応する。

第 4 層: タスク内制御層

ここでは各タスク内の対話の制御と応答内容の決定を行う。フォーム処理における充足性判定や状況判断、フォーム充足のための次応答処理のモダリティ制御、バージョンやシステム割り込みなどタスク内の対話遷移処理などが含まれる。VoiceXML の FIA アルゴリズムや HTML の form 要素などに対応する。

第 5 層: タスク間制御層

ここでは対話タスクの全般的な制御を行う。また、アプリケーション層との入出力通信を行う。SCXML⁸, VoiceXML などに対応する。

第 6 層: アプリケーション層

ここではデータモデルとアプリケーションロジックを実装する。

なお、各層におけるデータ交換形式は EMMA⁹, XForms¹⁰ などに対応する。

この階層モデルは、必ずしもすべての要素を MMI 記述言語で実装するためのモデルではない。具体的なアプリケーションとその実装の詳細を検討するにあたって、本来異なるレイヤーに属する構成要素を明確に区別することが目的である。この階層モデルにより、各要求仕様における課題がどの階層で解消されるべきかを確認すること、記述言語と動作の対応関係や原理を確認すること、ユーザ操作とシステムとのインタラクションを確認すること、要求仕様の過不足を確認することなどが容易になると考えられる。

モダリティに依存しない階層を明確に独立させて位置づけたことで、モダリティ独立な MMI 記述言語という理想をいかに実現するべきか、といった議論が容易になる。また、既存の MMI 記述言語がどの階層をどのようにカバーしているか、といった分析も可能になる。さらに AJAX など新しいウェブアプリケーションの技術を分析する上でも統一的な視点が得られる。

³ <http://www.w3.org/TR/SVG11/>

⁴ <http://www.w3.org/TR/voicexml21/>

⁵ <http://www.w3.org/TR/speechsynthesis11/>

⁶ <http://www.saltforum.org/>

⁷ <http://www.w3.org/TR/SMIL/>

⁸ <http://www.w3.org/TR/scxml/>

⁹ <http://www.w3.org/TR/emma/>

¹⁰ <http://www.w3.org/TR/xforms11/>

表 1 ユースケース一覧

	内容	入出力モダリティ	概要
a	オンラインショッピング	入力：マウス，音声 出力：ディスプレイ，音声，エージェント	音声インタフェースを備えた PC 上でのオンラインショッピング。ユーザに応じたショッピングコンテンツの選択や、多様なモダリティを組み合わせたユーザ入力が行われる。
b	音声によるディレクトリ検索	入力：マウス，音声 出力：ディスプレイ，音声	音声による飲食店の検索。主にスロットフィリング型の対話を扱っており、フィールド値に応じた対話進行や、フィールド値の動的変更等が行われる。
c	サイト検索	入力：マウス，音声，キー 出力：ディスプレイ，音声	PC あるいは携帯電話を端末とした、音声・キーによるサイト検索。
d	ロボットとの対話	入力：画像，音声，センサ 出力：ディスプレイ，音声	カメラやセンサ，ディスプレイを備えたロボットとの対話。画像によるユーザの認識や、接触によるインタラクション、web ページの表示やビデオコンテンツの再生を行う。
e	対話エージェントとの交渉	入力：音声 出力：音声，エージェント	歯科での予約受付を想定した対話エージェントとの交渉。ユーザ発話中のシステムの割り込み、ユーザ動作や状況の理解を行う。
f	音声情報案内システム	入力：マウス，音声 出力：ディスプレイ，音声	施設内の設備案内、交通情報、ニュース等の情報提供を行う情報キオスク。年少のユーザによる予想外の行動（罵声や意味の無い音声、マイクを叩いた音など）にも対応する。
g	エリアガイド	入力：ペン，音声 出力：ディスプレイ，音声	エリアガイドに関するユースケースである。ペンジェスチャによるユーザ入力を受け付ける。
h	カーナビ目的地設定	入力：タッチ，音声，PTT 出力：ディスプレイ，音声	プッシュアウト機能を備えた音声認識を行うカーナビゲーションシステム。

3. ユースケース・要求仕様・シーケンス図

検討の対象とするアプリケーションを具体化するために、まずユースケースを作成した。各ユースケースでは、対話シナリオ、利用モダリティ、システムに必要な機能等が纏められている。表 1 にユースケースの概略を示す¹¹。

これらのユースケースから、マルチモーダル対話システムの要求仕様を抽出した。要求仕様は以下に列挙する各項目からなる¹²。

- 一般的な要求
 - モダリティ利用や拡張性に関する一般的事項
- 入力モダリティに関する要求
 - 入力モダリティの種類や同期，利用法について
- 出力モダリティに関する要求
 - 出力モダリティの種類や同期，利用法について
- アーキテクチャ，統合，同期について
 - 各種標準技術への準拠，入出力の連携について
- 実行時と配置
 - 実行時環境の多様性，イベントハンドリング
- 対話制御について
 - 対話遷移の制御，履歴の利用等について
- フォーム・フィールドのハンドリング
 - フォームの充足性判定など
- アプリケーション・外部モジュールとの連携
 - サーバやデータベースとの連携について

- ユーザ情報，環境情報
 - ユーザの個人情報，端末情報等の利用
- 開発者の視点から見た機能
 - アプリケーション開発環境について
- アプリケーションとセッション
 - セッション管理について
- ECMAScript の利用

各ユースケースにおける階層の役割と階層間の情報のやり取りを明確にするために、それぞれのユースケースについてシーケンス図を作成した。シーケンス図では、各種エンジンや入力統合、タスク内遷移などの個別モジュールがオブジェクトとして表され、オブジェクト間のメッセージのやり取りが時系列順に記述されている。一例としてユースケース a のシーケンス図の一部分を図 2 に示す。



図 2 シーケンス図

¹¹ a, b, c, d, f の各ユースケースの詳細は、<http://www.astem.or.jp/istc/ISTC-SIG-MMI/index.html> において公開している。

¹² 詳細は <http://www.astem.or.jp/istc/ISTC-SIG-MMI/meeting12/MMI-Requirement3.pdf> を参照。

4. 階層のモジュール仕様

前述のシーケンス図による分析を元に、各階層の要求仕様を検討した。

最も下位に当たる第1層は信号の入出力と単独モダリティの認識・生成処理を行うモジュール(例えば Garatea toolkit における Julian や Galatea talk など)を想定している。以下では、第2層から第6層の仕様案を説明する。

4.1 第2層の仕様

第2層はモダリティコンポーネント層である。具体的には、第3層に対して統一的なインタフェースで、個々のモダリティに関する機能を提供する。第2層は多くの場合は第1層である入出力デバイス層のラッパーとして実装される。例えば音声入力や音声合成に関しては、Microsoft Speech API (SAPI) 対応のエンジンや独自仕様のアプリケーションおよびライブラリとして実装されているさまざまな第1層に対して、プラットフォーム独立のAPIを提供することが第2層の役割となる。その際、第1層のエンジンが標準入出力、ソケット通信、OS 依存の通信方法など、どのような通信方法で機能を提供しているか、ということについても抽象化がなされる。これにより、第3層ではプラットフォームに依存せずモダリティ間の協調方法のみを記述できるようになる。

また第2層では、第1層が提供している機能の概念レベルでの抽象化に基づく機能の分割や統合を行うことができる。例えば、HTML で実装された GUI を第1層の入出力デバイス層で用いる場合を考えると、これは第1層では単一のコンポーネントであるが、第2層では「GUI 入力」「GUI 出力」など複数のコンポーネントに分解され、第3層で他のモダリティコンポーネントとの協調を考慮して再統合されることが求められる。このような概念レベルでの機能分割を第2層が担うことになる。

一方で、第3層から見てブラックボックス化されるべきマルチモーダル部品が第2層の要素として実装される場合もあり得る。例えば、音声合成と顔画像出力はリップシンクを行う必要があるが、この機能は第3層から見た場合は「リップシンクを考慮した音声出力」として抽象化されていることが望ましい。また、音声認識エンジンはレベルメータ表示や認識エンジン状態の逐次出力などを伴うことがあるが、これらは時間的な粒度の小さい情報授受を伴うため、第3層で扱うには不適切な場合がある。そのような場合には、第3層から隠蔽したいモダリティ協調制御を抽象化し、仮想的なモダリティコンポーネントとして第2層で実装することが考えられる。

さらに、第3層からは仮想的な機能として要求されるが、第1層では実装の実体が存在しない場合もある。例えば「入力がなかった場合」(VoiceXML における noinput 要素)あるいは「文法にマッチしない入力が行われた場合」(VoiceXML における nomatch 要素)のような機能は、第1層の機能とタイマーなどを統合して提供される機能である。第2層では他の第2層コンポーネントを部品として用いつつこのような仮想的な機能を提供することも求められる。

このように、第2層は第1層と第3層の双方に対してAPIの抽象化を行う階層であると言える。

4.2 第3層の仕様

第3層は入力統合部と出力分化部からなる。以下、入力統合部と出力分化部の仕様、およびこれらの連携に関する仕様について順に述べる。

4.2.1 入力統合部

入力統合はマルチモーダル対話処理の要となる部分であり、これまでに多くの研究が行われている。代表的なものを挙げると、モダリティに関しては読唇+音声、顔画像認識+音声、ポインティング+音声を始めとする多様な組み合わせが検討されている[5][6]。また、統合方法に関しても、XML 言語を用いた手法やフレームの利用、型付素性構造を用いた手法など、様々な方法が提案されている[7]。

このように入力統合には多数の研究事例があるが、それぞれの研究機関で独自に行われているものが多い。したがってデファクトスタンダードと呼べるような手法がなく、入力統合の具体的な手法を仕様化することはシステムに制限を加えることになる。そこで現時点では第2層、第4層とのインタフェースを中心に入力統合部の仕様を定める。以下に、入力統合部が行う主な処理と入力統合部が備えるべきインタフェースを示す。

入力統合部における主な処理

- (1) 初期化処理
- (2) パラメータ更新
- (3) 入力受け付け開始
- (4) 通常入力の受理・統合処理
- (5) noinput, nomatch 等の受付
- (6) 終了処理

第4層とのインタフェース

- 第4層から初期化開始メッセージを受信 (1)
- 第4層からパラメータ更新メッセージを受信 (2)
- 第4層から入力受け付け開始メッセージを受信 (3)
- 第4層から終了メッセージを受信 (6)
- 第4層へデバイス情報を送信 (1)
- ユーザ入力の内容 (EMMA) を第4層に送信 (4)
- 第4層へ終了メッセージを送信 (6)

第2層とのインタフェース

- 第2層へ初期化開始メッセージを送信 (1)
 - 第2層へパラメータ更新メッセージを送信 (2)
 - 第2層へ入力受け付け開始&入力待機のメッセージを送信 (3)
 - 第2層へ入力受け付け停止メッセージを送信 (4)
 - 第2層へ終了メッセージを送信 (6)
 - 第2層からユーザ入力 (EMMA) を受信 (4)
 - 第2層から noinput を受信 (5)
 - 第2層から nomatch を受信 (5)
 - 第2層から終了メッセージを受信 (6)
- インタフェースの具体的なフォーマット、パラメータ等の詳細については、今後検討を進めていきたい。

4.2.2 出力分化部

ユーザ情報や端末の種類に基づいた出力生成の手法が数多く提案されているが、入力統合と同様にデファク

トスタンダードとなる手法はないため、第2層、第4層とのインタフェースを中心に仕様化する。以下に、出力分化部の主な処理内容と備えるべきインタフェースを示す。

出力分化部における主な処理

- (1) 初期化処理
- (2) パラメータ更新
- (3) プロンプト提示
- (4) 通常出力提示
- (5) 入力受付時の処理
- (6) retry, reprompt メッセージの受信時
- (7) 終了処理

第4層とのインタフェース

- 第4層から初期化開始メッセージを受信 (1)
 - 第4層からパラメータ更新メッセージを受信 (2)
 - 第4層からプロンプト提示メッセージを受信 (3)
 - 第4層から出力提示メッセージを受信 (4)
 - 第4層から終了メッセージを受信 (7)
- 第4層へデバイス情報を送信 (1)
 - 第4層へ出力完了メッセージを送信 (3)(4)(6)
 - 第4層へ終了メッセージを送信 (7)

第2層とのインタフェース

- 第2層へ初期化開始メッセージを送信 (1)
 - 第2層へパラメータ更新メッセージを送信 (2)
 - 第2層へプロンプト提示メッセージを送信 (3)(6)
 - 第2層へ出力提示メッセージを送信 (4)
 - 第2層へ出力中断、出力内容更新のメッセージを送信 (5)
 - 第2層へ終了メッセージを送信 (7)
- 第2層からデバイス情報を取得 (1)
 - 第2層から出力完了メッセージを受信 (3)(4)(6)
 - 第2層から終了メッセージを受信 (7)

4.2.3 入力と出力の連携およびその他の機能

ユーザ入力の受け付けに伴う出力内容の更新や中断、プロンプトの再提示等の処理は、入力統合部と出力分化部の間で行うことが合理的である。このとき必要となるメッセージを以下に挙げる。

- 入力統合部から出力分化部に入力受信のメッセージを送信
- 入力統合部から出力分化部に reprompt のメッセージを送信
- 入力統合部から出力分化部に retry のメッセージを送信
- 出力分化部から入力統合部に出力完了メッセージを送信

この他、入力統合部、出力分化部は、ユーザや環境に適応した入出力を実現するため、ユーザ情報、環境・端末情報を取得するモジュールとの通信を行うインタフェースを備える必要がある。

4.3 第4層の仕様

第4層はひとまとまりの小さな対話タスクを扱うもので、バックエンドアプリケーションとのやりとりを含まないものと想定している。ただし、単純なエラーハンドリング(例えば、出発時刻<到着時刻のチェック)や、スロット値の更新による振る舞いの変化などはこの層で処理するものと考えられる。全体の処理としては、フォームの充足性を判定し、これが満たされれば呼び出し元の第5層に終了イベントを投げけるものとする。

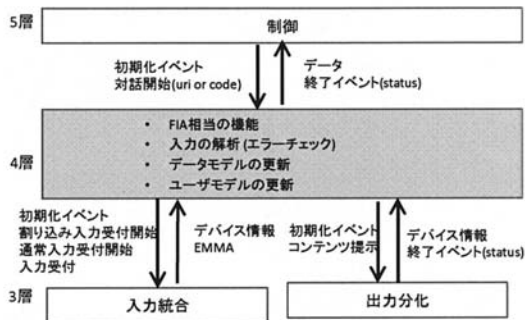


図3 第4層の入出力仕様

第4層の要求を満たす記述言語として、音声モダリティではVoiceXML, GUI入出力と音声ではXHTML+Voiceをそれぞれ拡張するという方向が考えられる。

4.4 第5層の仕様

第5層はタスクの大きな流れを管理するものと想定している。おおむねMVCモデルのC(すなわちコントローラ)に相当するものである。主要な入出力仕様を以下に示す。

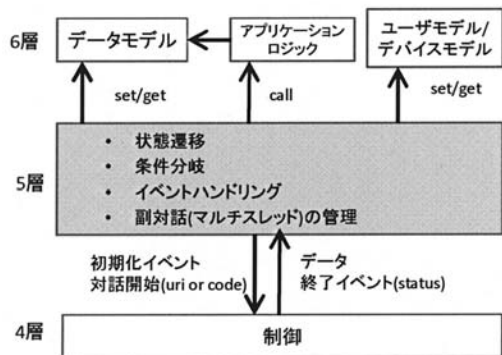


図4 第5層の入出力仕様

第5層の記述言語候補としては、明示的に状態遷移を書く場合はSCXMLが考えられる。また、エントリーポイントとその処理という、簡略化されたMVCモデルのコントローラ記述(例えば Rails フレームワークのスクリプト言語で書かれたコントローラ)を採用することもできる。特に5層の記述言語として Rails のスクリプト言語を採用する場合は、バックエンドデータベースとの連携がシームレスに行われる(いわゆる Object / Relational mapping)ことから、6層のアプリケーションロジック+データモデルを組み込んだ機能が提供できる。

4.5 第6層の仕様

第6層は対話を支えるコンポーネントが配置されている。MVC モデルの M (すなわちモデル) にあたる DB アクセス、Web API アクセスの機能を持つアプリケーションロジックや、対話アプリにユーザ情報・特性やデバイスの状態を通知するコンポーネントなどが想定される。

アプリケーションロジックとデータモデルは通常の Web アプリケーションで採用されているコンポーネントが利用可能であるが、そこに格納されている情報に MMI システムに適した処理が施されていると、想定されるアプリケーションの幅が大きく広がる。例えば MPEG-7 などであノテーションされている動画コンテンツが格納されていれば、ユーザからの要求に応じてダイジェストなどを配信するアプリケーションが考えられるし、SVG で記述された地図はポインティング入力の解釈をより高度なものにすることが考えられる。

一方、ユーザモデル/デバイスモデルコンポーネントは MMI システムに特有のものとなる。ユーザ端末のスクリーンの大きさや、音声入出力機能の有無といった静的情報や、環境雑音やユーザの好みの使用モダリティといった動的情報が管理される。また、場合によっては更新情報がイベントとして必要なコンポーネントに投げられる。

デバイスモデルの標準化に関しては、W3C において、Delivery Context: Client Interfaces (DCCI)¹³として API が検討されている。また、ユーザモデルを複数のアプリケーションから共有する試みとして、[8]などがある。いずれにしても標準コンポーネントとして機能するためには、その API だけではなく、それぞれのモデルのオントロジーを整備することが必要である。

ユーザモデル/デバイスモデルコンポーネントは5層からのみアクセスされるのではなく、ユーザモデルの更新のために第4層からアクセスされたり、出力モダリティを選択するために第3層からアクセスされたりすることも考えられる。

5. 関連研究

対話システムのアーキテクチャとしては、ハブ-スポーク構造の Galaxy アーキテクチャ[9]や、マルチエージェントに基づいた OAA が挙げられる[10]。これらはハブスクリプトやファシリテータが全体の制御を行っており、非同期的なイベント処理が多くなる MMI システムのアーキテクチャには適さない。

SmartKom プロジェクト[11]では Mobile (車載機器)、Home (家電操作)、Public (キオスク端末)という3種のアプリケーションシナリオを設定し、その共通開発基盤の設計からユーザビリティテストやコーパス収集まで様々な研究開発がおこなわれてきた。SmartKom では、まず階層的な基本アーキテクチャを設計し、それぞれのアプリケーションに対して必要なコンポーネントを組み合わせて実装するという考え方に基づいている。特に我々の提案における2層、3層の部分が詳細に検討されているが、4層以上に関しては、切り分けは曖昧である。

W3C の MMI アーキテクチャ¹⁴は、既存のモダリティコンポーネントを最大限利用する立場で考えられてお

り、現状の技術を利用した実用システムの構築としてはよい指針となるが、モダリティ統合・分化の部分があまり検討されておらず、研究用プラットフォームには向いていない。

6. 今後の課題

本稿では MMI システムのための階層的アーキテクチャの提案をおこなった。今後は、各層での記述言語候補を検討し、いくつかのアプリケーションの参照実装などを通じて、これらを試行標準案としてまとめてゆきたい。

謝辞

オブザーバとして委員会にご参加いただき、有益な御意見をいただいた芦村和幸氏 (W3C) に深く感謝いたします。

参考文献

- [1] M. Johnston, et al.: MATCH: An architecture for multimodal dialogue systems. In Proc. of ACL, pp.376-383 (2002)
- [2] J. Gustafson, et al.: AdApt a multimodal conversational dialogue system in an apartment domain. In Proc. of ICSLP, Vol.2, pp.134-137 (2000)
- [3] A. Gruenstein and S. Seneff: Releasing a Multimodal Dialogue System into the Wild: User Support Mechanisms, In Proc. SIGDial 2007, pp.111-119 (2007)
- [4] 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)
- [5] 速水悟, 竹澤寿幸: マルチモーダル情報統合システムの研究動向, 人工知能学会誌 Vol.13, No.2, pp.206-211 (1998).
- [6] D. Gibbon, I. Mertins and R.K. Moore (ed.): Handbook of Multimodal and Spoken Dialogue Systems, Kluwer Academic Publishers, section 2.2, pp.118-122 (2000).
- [7] R. L. Delgado and M. Araki: Spoken, Multilingual and Multimodal Dialogue Systems, John Wiley & Sons, Ltd, section2.2.2, pp.36-40 (2005).
- [8] D. Heckmann and A. Krueger: A User Modeling Markup Language (UserML) for Ubiquitous Computing, In UM 2003, LNAI 2702: Proceedings of the Ninth International Conference on User Modeling, Berlin Heidelberg: Springer, pp.393-397 (2003).
- [9] J. Polifroni, and S. Seneff: Galaxy-II as an Architecture for Spoken Dialogue Evaluation Proc. LREC, pp.42-50 (2000)
- [10] L. Julia, A. Cheyer, L. Neumeyer, J. Dowding and M. Charefeddine, "HTTP://WWW.SPEECH.SRI.COM/DEMOS/ATIS.H TML", Proc. AAAI97, pp.72-76 (1997)
- [11] W. Wahlster (ed.): SmartKom: Foundations of Multimodal Dialogue Systems, Springer (2006)

¹³ <http://www.w3.org/TR/DPF/>

¹⁴ <http://www.w3.org/TR/mmi-arch/>