Vol.2009-SLP-78 No.5
2009/10/23

階層的 MMI アーキテクチャに基づくプラットフォーム実装方法の検討

荒木 雅弘[†]　西本 卓也[††]　桂田 浩一[†††]　新田 恒雄[†††]

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」(WG4)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，階層的マルチモーダル対話システムアーキテクチャを提案した．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．本稿では，提案アーキテクチャに基づくマルチモーダル対話システム開発のためのプラットフォームの各種実装について報告する．

Studies on implementation methods of a development platform based on Hierarchical MMI Architecture

Masahiro Araki[†]　Takuya Nishimoto[††]
Kouichi Katsurada[†††]　and　Tsuneo Nitta[†††]

The speech interface committee W4G under ITSCJ (Information Technology Standards Commission of Japan) proposed a hierarchical architecture of multimodal dialogue systems for the purpose of standardization of multimodal interaction technology. The aim of the proposed architecture is to support practical system development by complying with the existing markup language and development framework, and to function as a research platform by specifying the role of each component. This paper reports some implementations of platform based on the proposed architecture.

1. はじめに

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」(WG4)では，MMI (MultiModal Interaction)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．検討の結果，図 1 に示すような階層的 MMI アーキテクチャを試行標準案として提案した[1].

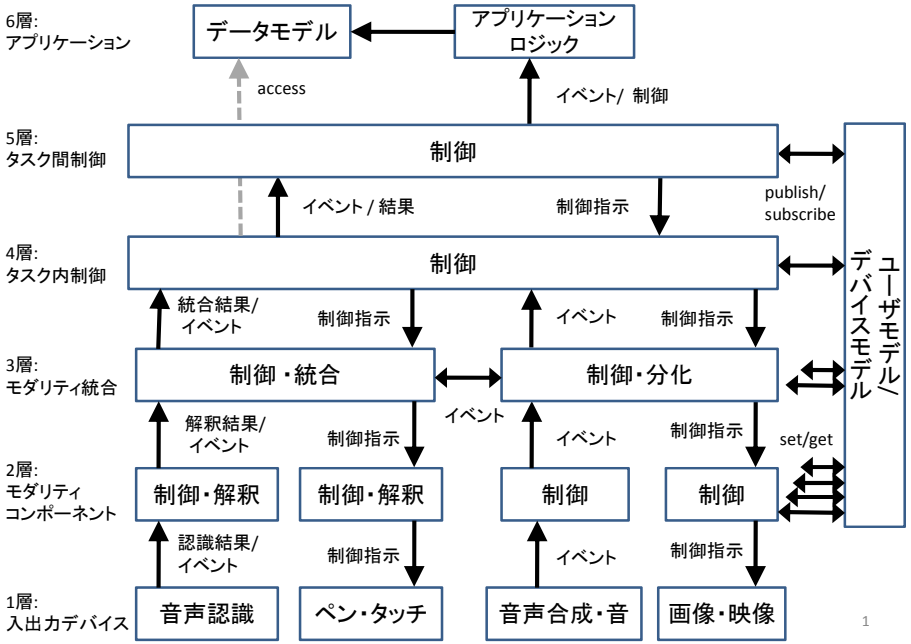

図 1　階層的 MMI アーキテクチャ

† 京都工芸繊維大学
Kyoto Institute of Technology
†† 東京大学
The University of Tokyo
††† 豊橋技術科学大学
Toyohashi University of Technology

提案アーキテクチャは 6 階層から構成されている．1 層は個別モダリティの認識・合成を行う既存モジュールであり，標準化の対象外である．2 層はそれらに標準 API を持たせるためのラッパーである．擬人化エージェントのリップシンクを伴う音声合成など，低レベルのモダリティ統合・分化はこの 2 層で行う．3 層は各モダリティから入力された情報を統合したり，出力の際に各モダリティに情報を分化させる役割を果たし，モダリティの違いをここで吸収する．4 層はモダリティに依存しないインタラクションのパターンを記述する．HTML や VoiceXML の Form 処理に相当する．5 層は大きなタスクの流れを記述し，4 層でのインタラクションの結果に応じて 6 層のアプリケーションとの連携を行う．6 層はデータベースなどをバックエンドとしたビジネスロジックを実装したアプリケーションである．ユーザモデル／デバイスモデルは現在使用中のデバイスの情報やインタラクション中のユーザの情報を管理することで，環境や使用状況に適応的なインタラクションを実現する．

以下本稿では，2 章で提案アーキテクチャに基づいたシステム開発用プラットフォームについて，3 章でユーザモデル／デバイスモデルコンポーネントの実装について，4 章で上位層での知識駆動開発について報告する．最後に 5 章でまとめと今後の課題について述べる．

2. 6 階層モデルに準拠した Web ベース MMI システムの開発

豊橋技術科学大学では，昨年度までに Web ブラウザをインタフェースとする MMI システム[2]を開発してきた．このシステムは，JavaScript などの標準技術のみを用いているため，特別なソフトウェアや高性能端末なしに MMI を提供できるのが特長である．本年度はこのシステムを 6 階層モデルに準拠するよう再構築した[3]．以下，Web ベース MMI システムの概要を述べた後に 6 階層モデルに準拠したシステムについて説明する．

2.1 Web ベース MMI システム

Web ベース MMI システムは，Galatea toolkit [4]を基にして構築された MMI システムである．このシステムは Ajax および Comet の技術を利用してサーバとブラウザを連携させ，高負荷な処理をサーバ上で，低負荷な処理をブラウザで行なうよう設計されている．図 2 にシステム構成を，表 1 にシステムが扱うことができるモダリティを示す．以下では Web ブラウザ上での処理とサーバ上での処理に分けて説明する．

2.1.1 Web ブラウザでの処理

ユーザからの入力取得

音声認識は複雑な処理を必要とするため，ブラウザ単体で低負荷に実装するのは困難である．そのため録音をブラウザで，認識をサーバ上で行なうことにより，ブラウザへの負荷が少ない音声認識を実現した．この手法は西村らの w3voice [5]で実用性が確認されている．ブラウザ上での録音は Java Applet を用いた音声録音器 Sound Recorder によって行なわれる．録音した WAV 音声データは Base64 エンコードされた後，JavaScript で構築されたブラウザ制御器である Browser Controller によってサーバに送られ，音声認識処理にかけられる．また，ポインティングなど音声以外の入力についても Browser Controller によって取得され，サーバに送信される．

表 1　利用可能なモダリティ一覧

	利用可能なモダリティ
入力	音声，ポインティング，キーボード
出力	ブラウザ，エージェント(動画)，合成音声

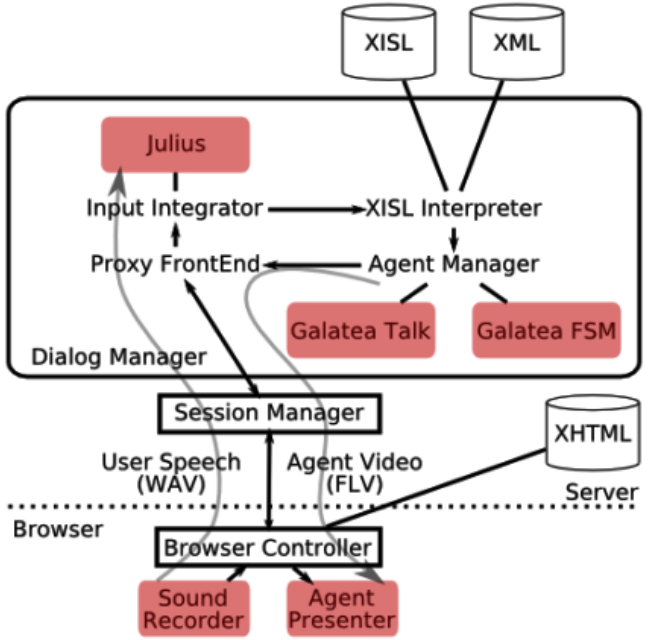

図 2　Web ベース MMI システムの構成図

ユーザへの出力

顔画像合成と音声合成によるエージェント出力は高負荷な処理となる．そこで，顔画像と音声をサーバ上の Agent Manager において 1 つの FLV 形式の動画に結合し，ブラウザ上でその動画を再生することでエージェント出力を低負荷に実現する方法を採った．動画再生には Adobe Flash を用いた動画再生器である Agent Presenter を使用した. また, ページ遷移などのブラウザ出力は Browser Controller によって実現している．

2.1.2 サーバ上での処理

ブラウザからのマルチモーダル入力データは，図 2 に示すサーバ上の Session Manager が受け取り，統合前処理（音声入力は音声認識結果に変換）の後，Dialog Manager 内の Input Integrator に送られ，統合処理が行なわれる．統合結果は，MMI 対話シナリオ記述言語 XISL で記述された文書を解釈する XISL Interpreter に渡され，対話シナリオに沿った出力命令が生成される．出力がエージェントの場合は，Agent Manager で動画が生成され，Session Manager を通じてブラウザに送信される．

その他の出力（Web ページの表示など）の場合，そのまま Session Manager を通じてブラウザに Web ページ表示などの命令が送られる．

2.2 Web ベース MMI システムの 6 階層モデルへの対応

これまで開発してきた Web ベース MMI システム（以後，従来システムと呼ぶ）を図 3 に示すような，MMI 6 階層モデルに準拠する形で再構築した．図 3 中の 4 層は，MMI 6 階層モデルにおける 4 層と 5 層を一つの層に統合したものになっている．
従来システムと今回開発したシステムの処理を比較すると，Web ブラウザのモジュール構成は同様であるが，サーバ上のモジュール構成が大きく異なる．

従来システムの入力統合および統合前処理の音声認識は，全て入力統合器 Input Integrator で行なわれていた．これに対して，今回開発したシステムの入力部は，統合前処理の音声認識を 2 層の入力管理部 Modality Input Manager で行ない，入力統合を 3 層の統合管理部 Input Integrate Manager で行なうように設計している．

また, 従来システムの対話管理および出力管理は, 対話管理部である XISL Interpreter が逐次出力などのタイミングを管理し，エージェント生成器 Agent Manager に出力の指示を出していた．これに対して，今回開発したシステムでは XISL Interpreter には対話管理だけを行なわせ，出力タイミングの管理は 3 層の出力管理部 Output Control Manager が行なうように設計した．また Agent Manager については，2 層の出力部 Modality Output Manager のモジュールの一つとして構築した．

以上の改良によって，モダリティ追加・変更などの拡張性が向上したといえる．例えば，音声認識エンジンを変更する場合は認識モジュールを差し替えるだけでよい．このような柔軟な変更が可能な理由は，各層の独立性が高く，一つの層に対する変更が他の層に影響を与え難いという特徴が MMI 6 階層モデルにあるためである．

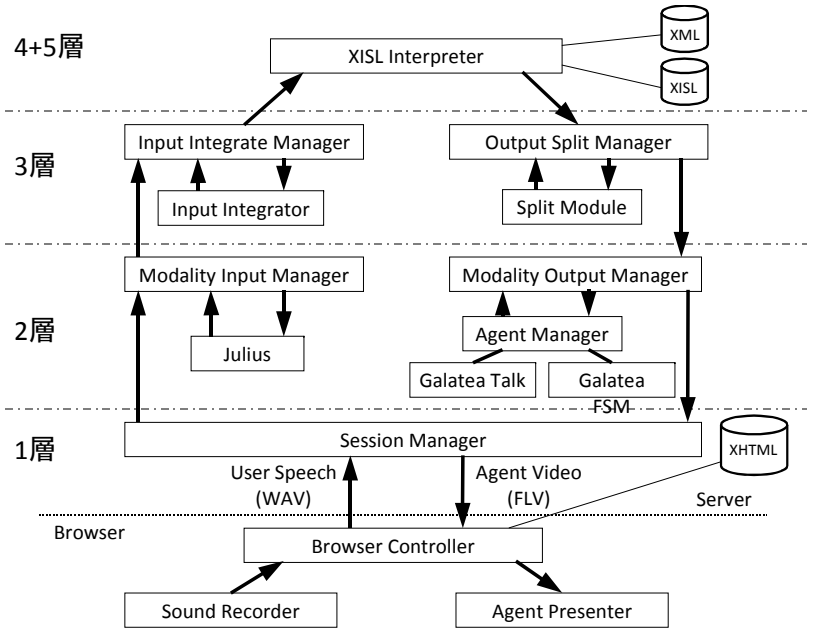

図 3　6 階層モデルに準拠した Web ベース MMI システム

3. ユーザモデル／デバイスモデルコンポーネントの実装

MMI システムはモバイル環境や家庭用ロボットとのインタフェースとして利用されることが想定されており，端末機やユーザエージェント(GUI における Web ブラウザに相当する)は多様なものが想定される．それらの多様な端末に対して，それぞれの利用状況に適したインタラクションを行うためには，端末機やユーザエージェントに関する詳細な情報を管理するコンポーネントが必要になる．

また，モダリティが多様化すればユーザとのやりとりも多様化するため，ユーザが好むモダリティの情報やユーザの習熟度に応じた対話制御などを実現するために，ユーザ情報を管理するコンポーネントも重要な役割を果たす．

本章ではこのユーザモデル／デバイスモデルコンポーネントの実装について，W3C における標準化動向と比較しながら，その概要を説明する．

3.1 ユーザモデルコンポーネント(UM)の要求仕様

UM はマルチモーダルインタラクションの過程におけるユーザ情報を管理できる必要がある．管理すべきユーザ情報としては，静的なものとして音響的特徴，言語的特徴，特定の入力モダリティへの習熟度，利用頻度の高いモダリティ，特定のドメインに関する知識，興味のある分野など，また動的なものとして現在用いているモダリティ，現在のインタラクションに対する没入度，感情などがある．他のコンポーネントがこれらの情報を読み書きできること，また動的な情報の変化を他のコンポーネントに通知する必要がある．

3.2 デバイスモデルコンポーネント(DM)の要求仕様

DM の要件は，W3C の UWA(Ubiquitous Web Applications) WG が目的としているものと同様で，様々なユーザクライアントやその使用状況に対して，適したコンテンツを配信するための情報を管理することである．静的なデバイス情報としては利用可能なモダリティ，画面サイズ，対応している記述言語の種類とバージョンなどがあり，動的な情報には背景雑音，通信速度などがある．

3.3 UM/DM の位置付け

ユーザ毎の音響モデル適応情報や言語モデルは 2 層からアクセスする．また，インタラクションログも 2 層で取得し，ログデータの URI (Universal Resource Identifier)を UM に通知することで，話者適応プロセスにデータを渡すことができる．3 層では統合・分化の過程で現在利用可能なモダリティの情報を DM から取得する．特に分化においてはユーザが現在好むモダリティを UM からも取得する．4 層では UM からユーザの習熟度を取得し，システム主導・混合主導を切り替えることができる．そのインタラクションの成否をもとにユーザの習熟度を更新する．5 層はユーザの興味の測定結果を UM から取得し，動的に提示コンテンツを変える処理を行う．このように UM/DM は 2 層から 5 層の各コンポーネントからアクセスされることになるので、提案アーキテクチャでは各層を縦断する形で位置付けられている．

3.4 RDF データストアによる UM/DM の実装

W3C UWA から提案されている CC/PP (Composite Capabilities/Preference Profiles) Structure and Vocabularies 2.0 [6]では，ユーザエージェントの特性を記述する方法として，セマンティック Web のデータ表現である RDF (Resource Description Framework)を採用している．CC/PP では，特定の利用状況を表現するトップレベルの構造は，ccpp:component 属性の目的語としてハードウェア・OS・ユーザエージェントを持つ．そしてそれぞれの目的語が画面サイズやバージョン番号など，インタラクションを特定するのに必要な情報を持つ．

また，ユーザモデルに関しては Heckmann ら[7]が同じく RDF の利用を前提とし，オントロジー記述言語 OWL (Web Ontology Language)に基づくユーザモデリングオントロジーGUMO を提案している．

これらを参考にして考案したデバイス／ユーザプロファイルの例を図 4 に示す．MMI のセッションを一つの URI で示し，デバイス／ユーザプロファイルと関連付ける．デバイスプロファイルは W3C 標準に準拠し、端末ハードウェア・端末ソフトウェア・ユーザエージェントをそれぞれ構成要素として持ち、さらに詳細な情報を RDF グラフで表現する．ユーザプロファイルは感情状態・個性・特性・物理状態の基本語彙は GUMO[6]に準拠し，その他のモダリティ選好性やタスク依存のユーザモデリング変数は独自拡張とする．ユーザが不特定の場合は RDF の空白ノードを用いてそのセッション中のみ有効な一時ユーザ変数管理を行う．

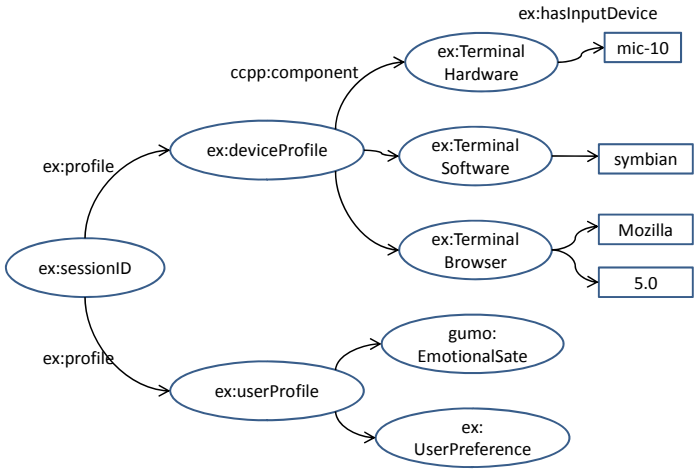

図 4　RDF に基づくデバイス／ユーザプロファイルの例

RDF によるデバイス情報・ユーザ情報の表現は，その表現が Web 標準であるという利点も当然であるが，セマンティック Web 技術の様々な特徴を生かすことができることが大きな利点となる．

ひとつには共通オントロジーによる語彙の標準化が期待できるという点がある．文書情報においては，RSS (RDF Site Summary)でも利用されている Dublin Core が標準となっているため，特定の文書のタイトルを検索するには，そのリソースの dc:Title プ

ロパティの値を調べればよいということになっている．MMI に関するデバイス情報・ユーザ情報の標準オントロジーが普及すれば, MMI に関しても同様な状況が期待できる．デバイス情報のオントロジーに関しては，W3C で標準化が進められており[8], ユーザ情報に関しては Heckmann ら[6]の試みがある．

また，RDF による表現では推論が可能であるという点も，実装に適した選択であるといえる．特にユーザ情報に関して，特定の情報が検索できなくとも，他の情報から推定するなど，様々な応用が考えられる．

4. 音声対話技術の普及促進と進化

4.1 Ruby による対話記述の検討

さまざまな機能を持つ Web ベースのアプリケーションが広く使われるようになった現在こそ，たとえその一部でも音声インタフェースを介して利用できることの意義は大きい．ブラウザのフォームに情報を埋める作業を繰り返していると，もっと効率よく，あるいは，キーボードやマウスに頼らずに操作したい，と感じるのではなかろうか．

システム記述言語の設計においては，実績のある成功事例（ベストプラクティス）が有用である．我々の 6 階層アーキテクチャにおける第 5〜6 層（タスク間制御，データモデル，アプリケーションロジック）について，既存の Web アプリケーション開発から借用できる成功事例の一つとして，Rails (Ruby on Rails) に着目している．

VoiceXML は第 5 層と第 4 層の界面に対応しており，これは一般的な Web における HTML に相当する．Web アプリケーションではテンプレートエンジン（HTML に埋め込まれたスクリプト言語を実行する処理系）が一般的である．「階層の界面が記述言語に対応し，各階層がテンプレートエンジン処理系に対応する」という構図は 6 階層モデルの随所に当てはまる．テンプレートエンジンにはさまざまな技術や記述言語が乱立しているが，Rails による Web アプリケーション開発ではオブジェクト指向が徹底され，MVC のすべての要素が Ruby で記述され，一貫性がある．Ruby 言語はコードブロックによって手続き型言語と宣言型言語の記述の混在が可能になり, いわゆる「ドメイン記述言語」への流用が容易とされる．

Model においては SQL データベースを簡潔な記述で操作できるクラスライブラリがある．

View においては HTML に Ruby の記述を埋め込むテンプレートエンジン機能 (ERB)がある．

Ruby による VoiceXML アプリケーションの事例は, 階層モデルにおける記述言語やアーキテクチャの詳細を考える出発点になるという立場から, Galatea Dialog Studio の開発は Ruby on Rails との互換性を重視して進めている[9][10].

4.2 知識からの対話生成

音声対話システムによって「どのようなインタラクションを実現すべきか」を議論することは重要である．目標が定まらない段階でアーキテクチャや記述言語を検討するのは時期尚早という意見もある．これまでの標準化活動では，できるだけ先進的なユースケースを取り入れることでこの問題を克服してきたが，ユースケースもやがて時代遅れになる懸念がある．

これに対して，時代遅れになりにくい「抽象的で普遍的な情報構造」に着目して，インタラクションの詳細を後から開発・標準化する，というアプローチがある．多くの実現例が報告されている「一問一答型の対話システム」も一例と言える．書籍のメタファで音声対話コンテンツを記述する提案はこれまでにも行われてきた[11][12].

近年「情報提供型の音声対話」の要素技術として注目に値するのは，障害の有無にかかわらず読書ができる環境を実現する「マルチメディア DAISY」[13]である．

その派生技術である「テキスト DAISY」はテキスト音声合成技術の新しい応用分野である．ハイパーリンクや検索といった電子書籍の操作手段としての音声対話にも期待が高まる．

4.3 コミュニケーションの効率性

擬人化音声対話エージェント技術は「人間が声で会話したいと感じるような人工物をいかに実現するか」という問題への一つの回答だと考えられてきた[14].

その目標を真に達成するためには「対人コミュニケーション」を形式的な問題として捉えるのではなく，高品質の映像や音声を高速に制御し，豊かな情報の伝達を可能にし，コミュニケーションの効率性を本質的に高める必要があろう．

エージェント制御に力学や物理学のモデルを取り入れる試み[15]，音声インタフェースを「実時間の効率性」という観点から構成要素に分解する検討[16]などはマルチモーダル対話アーキテクチャに今後必要となる視点を与えるだろう．

5. おわりに

ISTC (音声対話技術コンソーシアム)，および学会試行標準委員会(WG4)を中心に策定した，6 階層モデルに準拠したシステム実装と，検討結果を述べた．今回の階層モデルは，多くのユースケースを基に，そこに現れる複数のモダリティを含む対話を，システム上で如何に確実に動作させるかを中心に討議した結果の「叩き台」である．試行標準は，実装評価をもとに随時改定することが可能なため，今後，機能追加やモデル改良に向け，多くの研究者・開発者の方達の助力をお願いしたい．

MMI の記述言語は，現在，個別モダリティに対応する様々な言語から成る複合ドキ

ュメント形式が，W3C を中心に討議されている．我々も，今後，様々な記述言語による 6 階層モデルの実装と評価を行い，その結果を基に W3C ほかへ向けて，言語仕様に関する提案を行っていきたい．

謝辞 学会試行標準委員会(WG4)の委員として，試行標準案の作成にご尽力いただいた甘粕哲郎氏(NTT)，川本真一氏(ATR)に感謝いたします．また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) 新田恒雄, 桂田浩一, 荒木雅弘, 西本卓也, 甘粕哲郎, 川本真一: 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, 情報処理学会研究報告, 2007-SLP-68-2, (2007).
2) 桐畑輝樹，工藤正志，高田淳貴，桂田浩一，新田恒雄：ウェブブラウザ上で動作可能なマルチモーダル対話システム，情報処理学会研究報告 2008-SLP-73，pp.35-40 (2008).
3) 工藤正志, 桂田浩一, 入部百合絵, 新田恒雄 : MMI6 階層モデルに準拠した Web ベース MMI システムの開発， FIT2009 情報科学技術フォーラム，E-039 (2009).
4) S. Kawamoto, et al.: Galatea: Open source software for developing anthropomorphic spoken dialog agents, in Life-Like Characters, ed. H. Prendinger and M. Ishizuka, pp.187-212, Springer-Verlag (2004).
5) 西村竜一，他： 音声入力・認識機能を有する Web システム w3voice の開発と運用，情報処理学会研究報告，2007-SLP-68-3，pp.13-18 (2007).
6) Kiss, C.: Composite Capability/Preference Profiles (CC/PP): Structure and Vocabularies 2.0, W3C Working Draft 30 April 2007, http://www.w3.org/TR/CCPP-struct-vocab2/
7) Heckmann, D., Schwarzkopf, E., Mori, J., Dengler, D. and Krner, A. : The user model and context ontology GUMO revisited for future web 2.0 extensions. In Proceedings of 3rd Contexts and Ontologies Workshop, Roskilde, Denmark. pp. 37-46, (2007).
8) Fonseca, J. M. C. and Lewis, R.: Delivery Context Ontology, W3C Working Draft 16 June 2009, http://www.w3.org/TR/dcontology/
9) http://ja.nishimotz.com/dialogstudio_rails
10) http://sourceforge.jp/projects/galatea/wiki/JapaneseTutorial
11) Nishimoto, T., Araki, M. and Niimi, Y.: RadioDoc : A Voice-Accessible Document System, Proc. ICSLP2002, pp.1485-1488, Denver, (2002).
12) 西本卓也, 荒木雅弘, 新美康永: 擬人化音声対話エージェントのためのタスク管理機能, 日本音響学会 2002 年春季研究発表会, 1-5-15, pp.29-30, (2002).
13) DAISY 研究センター http://www.dinf.ne.jp/doc/daisy/
14) 嵯峨山茂樹, 西本卓也, 中沢正幸: 擬人化音声対話エージェント, 情報処理学会誌, Vol.45, No.10, pp.1044-1049, (2004).
15) 中沢正幸, 西本卓也, 嵯峨山茂樹: 視線制御モデルによる擬人化音声対話エージェントの制御, 2005 年度人工知能学会全国大会(第 19 回)論文集, 3B2-07, (2005).
16) 西本卓也，岩田英三郎，櫻井実，廣瀬治人: 探索的検索のための音声入力インタフェースの検討，情報処理学会研究報告 2008-HCI-127(2), pp.9-14, (2008).