2006－SLP－62　(16)
2006／7／8

静的・動的情報を利用した MMI システムの設計と実装

桂田 浩一*　大隈 祐治*　矢野 誠*　入部 百合絵*　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

〒441-8580　愛知県豊橋市天伯町雲雀ヶ丘１－１
Email: {katurada, okuma, yano, iribe, nitta}@vox.tutkie.tut.ac.jp

あらまし：　本論文では web ベースのマルチモーダル対話(Multi-Modal Interacton: MMI)システムにおいてユーザの視線や表情，嗜好やプロファイル，端末周辺の雑音状況といった静的・動的情報を取り扱うフレームワークを提案する．これらの情報を対話で利用することにより，ユーザに適用した対話や状況に応じたモダリティ変更が可能になるため，より自然な対話を実現できる．本研究ではこれらの情報を管理するために，MMI システム内に静的・動的情報管理部を実装した．静的・動的情報管理部は静的・動的情報を一元管理するモジュールで，情報の収集と管理，および外部モジュールへの情報提供を行う．情報提供の方法として Pull 型（情報参照型）と Push 型（情報変化のイベント通知）の二種類を用意し，各モジュールにおける情報の利用方法に応じて選択できるようにした．本論文では静的・動的情報管理部の構成について述べ，アプリケーションの実装例を示すとともに，実験による評価と関連技術・研究との比較を行った．

キーワード：　静的情報，動的情報，マルチモーダル対話システム

Design and Implementation of Multi-Modal Interaction System using Static/Dynamic Information

Kouichi KATSURADA*, Yuji OKUMA*, Makoto YANO*,
Yurie IRIBE* and Tsuneo NITTA*

*Graduate School of Engineering, Toyohashi Univ. of Technology

1-1 Hibarigaoka, Tempaku-cho, Toyohashi 441-8580, JAPAN
Email: {katurada, okuma, yano, iribe, nitta}@vox.tutkie.tut.ac.jp

Abstract： This paper provides a mechanism to deal with static/dynamic information in a web-based Multi-Modal Interaction (MMI) system. The static/dynamic information in this paper includes user preference, user's facial expression, environmental information, and so on. By employing these properties, the MMI system can make interaction natural based on context or situation. To consolidate these properties we have designed and developed a static/dynamic information manager on our MMI system. The manager provides two types of interfaces to access the information: pull type interface and push type interface. Each module in the MMI system can receive the information through either of interfaces as the need arises. We prototyped a user navigation system that employs user's facial expression and GPS information as static/dynamic information. The evaluation test showed that the manager makes it easy to handle static/dynamic information when designing MMI systems. We clarified the characteristics of static/dynamic information manager by comparing with other researches and techniques.

Key words： Static information, Dynamic information, Multimodal interaction system

1. はじめに

ネットワーク技術の発展と web アクセスの多様化に伴い，マルチモーダル対話(Multi-Modal Interaction: MMI)の研究分野では，多様な端末機器で様々なモダリティを扱う web ベース MMI システムが活発に検討されるようになった. Web 技術の標準化団体である W3C が web ベース MMI システムのフレームワーク[1]とアーキテクチャ[2]を提案したのを始めとして，その他の機関においても独自に SALT[3] や XHTML+Voice[4]といった MMI 記述言語が検討されている. 筆者らの研究グループでも MMI 記述言語 XISL2.0[5]を提案するとともに，MMI システムのアーキテクチャ[6]を開発してきた.

こうした従来の web ベース MMI システムでは，音声やポインティングのように，ユーザの能動的(active)な入力操作を取り扱うことが検討の中心となっている. しかしながらユーザビリティを向上させるにはユーザの視線や表情といった非能動的(passive)な情報を併用することが重要であると指摘されている[7]. 本論文では視線や表情の他, 対話のユーザ適用に利用可能なユーザの嗜好やプロファイル, モダリティ変更の契機となる周辺の雑音状況やデバイス状況等（以上をまとめて本論文では静的・動的情報と呼ぶ）を web ベース MMI システムで扱うためのフレームワークを提案する.

静的・動的情報を取り扱う一つの方法として，ユーザ入力と同様に捉えてこれらの情報を処理することが考えられる. しかし対話進行の直接のきっかけとなる能動的な入力操作と, 付加的な情報である非能動的な情報を同一レベルで取り扱うと，対話進行の見通しが不透明になる. 特に動的情報には GPS 情報やユーザの視線のように絶え間なく変化するものもあるため, 一定のインターバルがある能動的入力がこれらの処理の中に埋もれる可能性がある. 一方で，動的情報の中には対話の特定の状況において参照できればよく，普段は参照されないような情報もある. 静的情報についてはそもそも情報に変化がないため, 入力操作と捉えるのは不自然である. こうした理由から本研究では能動的な入力操作と非能動的な静的・動的情報を区別し，後者を静的・動的情報管理部において管理することにした.

静的・動的情報管理部は静的・動的情報を一元管理するモジュールで，情報の収集と管理，および外部モジュールへの情報提供を行う. 情報提供の方法として Pull 型（情報参照型）と Push 型（情報変化のイベント通知）の二種類を用意し，

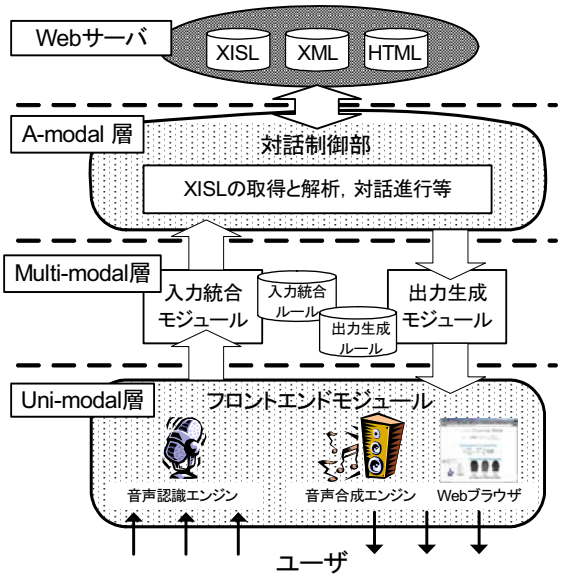

図 1: MMI システムの構成

各モジュールにおける情報の利用方法に応じて選択できるようにしている. また，情報収集／提供のインタフェースの統一により新規情報の導入を容易にしている. 以下，まず 2 節において本研究の基盤となる web ベース MMI システムについて概説する. 続く 3 節では静的・動的情報の詳細，および静的・動的情報管理部の構成について述べ，4 節においてアプリケーションの実装例を示す. さらに 5 節で静的・動的情報管理部の実験評価を行い，6 節において関連技術・研究との比較を行った後，7 節で本論文のまとめと今後の課題について述べる.

2. マルチモーダル対話システム

2.1 システムの概要

筆者らの研究グループではモダリティ拡張性の高い web ベース MMI システムのアーキテクチャを検討してきた[6]. 図 1 に示すようにシステムは Uni-modal 層, Multi-modal 層, A-modal 層の 3 つの階層から構成されている.

Uni-modal 層は入出力モダリティを扱う層で，筆者らが開発したシステムではフロントエンドモジュールとして実装されている. フロントエンドでは音声認識エンジンや音声合成, web ブラウザ，擬人化エージェント等を個別に制御する. フロントエンドが他のモジュールと分離されているため, システム全体を変更することなく新規モダリティの追加やモダリティの修正を行うことが可能になっている. Multi-modal 層はモダリティの統合・分化を行う層で，それぞれ入力統合モジュール, 出力生成モジュールとして実装されて

```
<?xml version="1.0" encoding="Shift-JIS"?>
<!DOCTYPE xisl SYSTEM "xisl.dtd">
<xisl version="2.0">
  <body>
  <form id="Shop">…… (1)
   <fe><!--html 文章--></fe>…… (2)
   <fe><!--音声認識文法--></fe>…… (3)
   <initial> …… (4)
    <prompt>
      <fe><!--「商品と個数を入力して下さい」--></fe>
    </prompt>
   </initial>
   <field name="ITEM">……(5)
    <prompt>
      <fe><!--「商品を入力してください」--></fe>
    </prompt>
    <filled>…… (6)
      <backend action="set.cgi"
                namelist="ITEM"/>…… (7)
          :
    </filled>
   </field>
   <field name="QTY">
          :
   </field>
   <filled namelist="ITEM QTY">
    <fe><!--「QTY 個の ITEM ですね？」--></fe>
          :
   </filled>
  </form>
  </body>
</xisl>
```

図 2: XISL2.0 の対話記述例

いる．これらのモジュールでは入力統合ルール，出力生成ルールを用いて統合，分化を行う．A-modal 層は入出力モダリティに非依存の情報を扱うモジュールで，本システムでは対話制御部として実装されている．対話制御部は MMI 記述言語 XISL2.0[5]に記述された対話シナリオに従って対話を進行する．XISL2.0 文書およびその他の文書は Web サーバに格納されており，必要に応じて各モジュールにダウンロードされる．

2.2 MMI 記述言語 XISL

XISL2.0[5]は，音声対話記述言語 VoiceXML[8]をベースにこれまで我々が策定を進めてきた MMI 記述言語 XISL1.1 を改良したものである．XISL2.0 ではスロットフィリングタイプのマルチモーダル入力を受け付ける他，対話遷移，条件分岐，数値演算など，マルチモーダル対話の進行に必要な様々な処理が記述可能になっている．図 2 に XISL2.0 の記述例を示す．

XISL2.0 では VoiceXML と同様の対話進行アルゴリズムを採用している．まず一纏まりのスロット群を表す<form>（図 2(1)）を訪れると，最初に(2)，(3)の<fe>が実行される．<fe>は XISL2.0 のオリジナルタグで入出力モダリティの動作を設定するために用いる．この例では HTML の表示と音声認識文法の設定を行っている．続いて初期処理である(4)の<initial>が実行され，プロンプトが出力される．(5)の<field>は入力を待ち受ける一つのスロットを表す．<field>が埋められると，システムが実行するアクションである(6)の<filled>が実行される．(7)の<backend>は，CGI などの外部プログラムを実行するためのタグである．<form>内に埋まっていないスロット（<field>要素）が存在すると，自動的に<field>内の<prompt>要素が実行され，ユーザに入力を促す．XISL2.0 の詳細については文献[5]を参照されたい．

3. 静的・動的情報の管理

3.1 静的・動的情報

静的･動的情報とは，静的情報（ユーザの嗜好，プロファイルなど）と動的情報（ユーザの表情，視線，位置など），および環境情報（ノイズレベル，端末のデバイス情報など）を総称するものである．これらの情報は MMI システムの様々なモジュール内で利用される可能性がある．例えばユーザの嗜好は対話制御部において対話コンテンツを選択する際に利用できる．また，ユーザの表情は入力統合モジュールにおける意図解釈に，ユーザプロファイルの一部である年齢･性別等は出力生成モジュールで出力のスタイリングを行う際に利用可能である．周辺のノイズレベルはフロントエンドモジュールが入出力の音量レベルを変更する場合や入出力モダリティを変更する際に用いることができる．本研究ではこれらの情報を静的・動的情報管理部において一元管理する．

3.2 静的・動的情報管理部の設計

ユーザの位置情報を利用するアプリケーションを想定した場合，リアルタイムに地図を更新するアプリケーションでは一定間隔ごとに位置情報を地図更新部に通知するインタフェースが有用であるが，例えば音声入力の「ここ」という発話とユーザの位置情報を統合する場合には，発話があった時点で位置情報を参照できればよい．また動的情報や環境情報には頻繁に変化するものもあるため，更新が容易なデータ構造を用いる必要がある．以上より，次の条件を満たすよう静的・動的情報管理部を設計した．

- 新規情報の追加や更新が容易であるデータ構造を持つ．
- 情報の変更を MMI システム内の他のモジュールへ通知する Push 型の情報提供機能を持つ．
- 他のモジュールからの情報参照を可能に

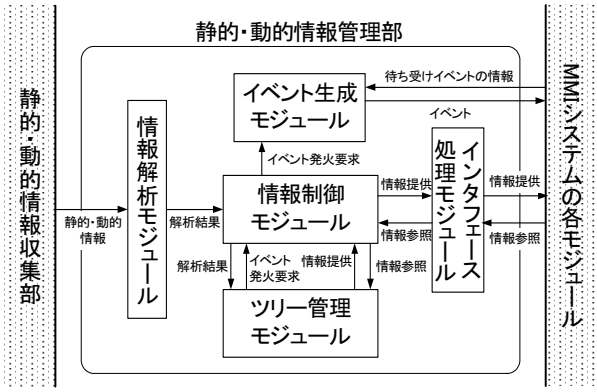

図 3: 静的・動的情報管理部

する Pull 型の情報提供機能を持つ.

これらの条件を満たすため, web ベースシステムにおいて静的・動的属性を扱うためのフレームワークである DCI(Delivery Context Interfaces) [9]を参考に静的・動的情報管理部を設計した. DCI の枠組みでは DOM ツリーの構造で属性が管理され, スクリプト言語等から属性を参照するための API や, 動的な属性変化を通知するための XML イベントに基づいた方法が規定されている. 本研究では DCI と同様の処理を検討した.

図 3 に静的・動的情報管理部の構成を示す. 図に示すように, 静的・動的情報管理部は情報解析モジュール, 情報制御モジュール, ツリー管理モジュール, イベント生成モジュール, インタフェース処理モジュールの 5 つのモジュールから構成される. 静的・動的情報収集部から情報が送られると, まず情報解析モジュールがこれを受け取り, データ解析を行った後に情報制御モジュールに送信する. 情報制御モジュールは受け取った情報をそのままツリー管理モジュールに送信する. ツリー管理モジュールは保持している情報と受け取った情報を比較し, もし異なっていれば情報制御モジュールを通してイベント生成モジュールに変化の内容を伝える. イベント生成モジュールは MMI システム内の各モジュールにイベントを送信する. 一方, MMI システム内のモジュールから情報参照のリクエストが来た場合には, インタフェース処理モジュールがそのリクエストを受理し, 情報制御モジュールを通してツリー管理モジュールから情報を取得し, 要求元に情報を伝える.

静的・動的情報は MMI システムの各モジュールの他にも, 対話制御部を通して XISL2.0 文書においても利用することができる. 上述の Push 型と Pull 型の双方の情報取得方法が利用できるように 2 種類のインタフェースを用意した. 一つ目のインタフェースは Push 型の情報通知を受け

```
<?xml version="1.0" encoding="Shift-JIS"?>
<!DOCTYPE xisl SYSTEM "xisl.dtd">
<xisl version="2.0">
  <body>
    <catch event="SE.chg_user_face_exp"
           return="user_face_emo"> …… (1)
      <if cond="session,SE.user#state
                eq 'dissatisfied' ")…… (2)
        <then>
          <fe><!--何かお困りでしょうか? --></fe>
          <goto next="#search_landmark">
        </then>
      </if>
    </catch>
  </body>
</xisl>
```

図 4: 静的・動的情報への XISL からのアクセス

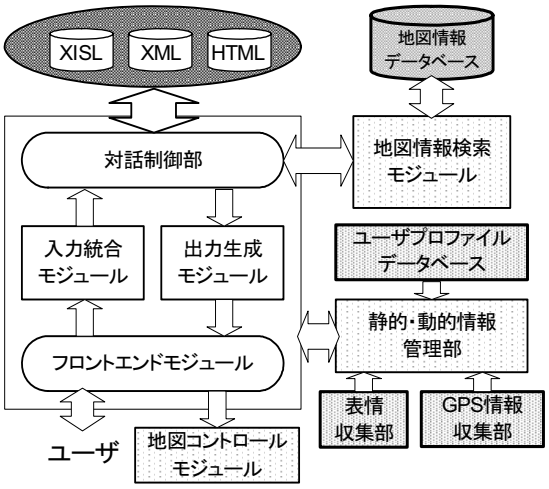

図 5: 音声対話ナビゲーションシステムの構成

取るための<catch>要素である. 図 4 の(1)に示す部分が XISL2.0 での記述例である. <catch>要素は動的情報に限らず一般的なイベントを取得するための要素であるが, 本研究では<catch>の event 属性に"SE....."という記述をすることにより静的・動的情報管理部からの情報を取得できるようにした. 一方, 二つ目のインタフェースは Pull 型の情報参照を可能にするためのもので, XISL2.0 では専用の変数を介して取得することにした. 図 4 の(2)に示す"session.SE...."の部分がツリーの該当部分を指定する変数である. これらのインタフェースにより XISL2.0 から静的・動的情報を利用できるようになり, ユーザビリティの高い web ベース MMI アプリケーションが作成可能になった.

4. アプリケーションの実装例

前節までに述べた MMI システムおよび静的・動的情報管理部を基盤として音声対話ナビゲー

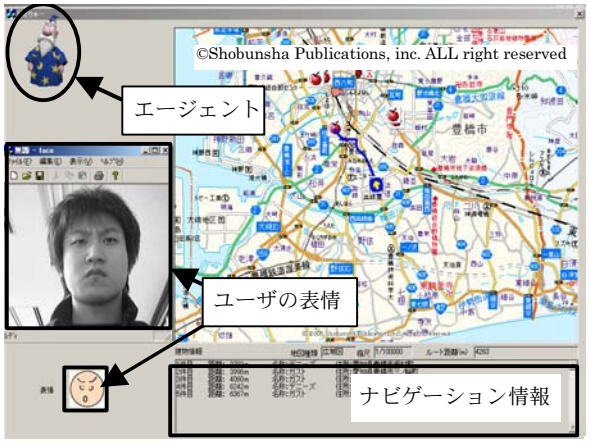

図 6: 音声対話ナビゲーションシステムの画面例

ユーザが運転中にレストランを検索
ユーザ： 腹減ったな.
システム：すき家が逆方向○○m の所にあります.
ユーザ： （不満げな顔）
システム：では，進行方向で検索しますか？
ユーザ： そうして.
システム：一番近いのは△△m 先のデニーズです.
ユーザ： じゃ，そこまでナビして.

図 7:ナビゲーションシステムとの対話例

ションシステムを構築した. システムはユーザの嗜好や表情に応じてユーザを様々な施設まで案内する．図 5 に示すように，2 節で述べた MMI システムに地図情報検索モジュール, 地図コントロールモジュールを追加し，さらに静的・動的情報収集部として GPS 情報収集部，表情収集部，ユーザプロファイルデータベースを用いた．図 6 に実行画面を示す.

GPS 情報収集部はユーザの位置，移動速度，移動方向を取得する. ユーザの位置は地図情報検索の中心座標を決定するために用いる. 移動速度はユーザが徒歩であるか, 交通機関を用いているかを推定する際に用いる. 移動速度からユーザが車に乗車していると推定できる場合, システムは検索範囲を広げ, 移動方向に基づいてできるだけ順方向の施設を提供するよう検索結果を調整する. 表情収集部ではユーザが不満げな顔をしたかどうかを取得する. ユーザが不満げな顔をした場合, ナビゲーション内容に対して不満があると判断し, システムは地図情報の検索範囲を変更するなどして対処する. 図 7 に音声対話ナビゲーションシステムでの対話例を示す.

5. 静的・動的情報管理部の評価

静的・動的情報管理部の評価を行うために，4

表 1:システム開発の容易さ

項目	評価
システムを開発しやすいのはどちらですか？	4.5
新たな静的・動的情報収集部を導入するとき，どちらを使いたいですか？	4.5
データの流れが把握しやすかったのは？	4.25
開発時間が短かったように感じたのは？	4.5
開発工程が複雑でなかったのは？	4.5

1 2 3 4 5
システム B ◀——▶ システム A

表 2:静的・動的情報管理部の機能評価

項目	評価
通信メッセージと XISL の変数名との関係の分かりやすさ	2.5
通信メッセージと XISL のイベント待ち受けとの関係の分かりやすさ	2.5
静的・動的情報管理部へ送信するメッセージの作成	3.25
メッセージの送信	4.25

1 2 3 4 5
理解が難しい ◀——▶ 理解が容易

名の被験者に 2 種類の MMI システムを開発させた．一つは静的・動的情報管理部を利用したシステム（システム A）であり，もう一方は静的・動的情報管理部を用いずに MMI システムの各モジュールと情報を直接やり取りするシステム（システム B）である．静的・動的情報としては表情認識と個人認証を用いた．システム A では被験者は上記の情報の収集部と静的・動的情報管理部の間の通信プログラムのみを実装すればよいのに対し，システム B では静的・動的情報管理部が提供している機能の一部（各モジュールとのインタフェースやイベント生成，情報管理など）を実装する必要がある. 実装の順序効果をなくすために，2 名の被験者には先にシステム A を，残る 2 名には先にシステム B を開発させた.

実験ではシステム実装に関する主観評価を行った．結果を表 1，表 2 に示す．表 1 は実装の容易さに関する主観評価，表 2 は静的・動的情報管理部の機能に関する主観評価である. 評価はそれぞれ 5 段階で行わせた．まず表 1 によると，システム A が全ての項目において高い評価を得ていることが分かる．結果より，静的・動的情報管理部の導入によってシステム開発が容易になっ

たことが示された. 一方, 表 2 によると, XISL2.0 の変数やイベントと通信メッセージの関係が理解しにくいとの結果が得られた．今後は変数名/イベント名と通信メッセージの対応の明確化により，静的・動的情報管理部の利便性を向上させたい.

6. 関連技術との比較

6.1 MMI システムにおける静的・動的情報管理

従来の MMI システムにおいても静的・動的情報は数多くのシステムで扱われている．例えば DFKI の SmartKom プロジェクト[10]で開発された SmartKom Public ではユーザの表情や音韻情報が，SmartKom Mobile では GPS 情報が利用されている．また，AT&T の MATCH システム[11]ではユーザプロファイルに基づいたナビゲーションを行っている．この他にも多様な MMI システムにおいて視線や顔画像認識等が取り扱われている[12]. これらのシステムの多くでは，代表的な音声対話アーキテクチャである GALAXYII[13]や上述の MATCH が採用する Open Agent Architecture（一体の対話管理エージェントが中心となって入力情報を管理する）方式か，あるいは SmartKom 等が採用する Blackboard（各モジュールが書込／閲覧できる共通掲示板で入力情報を管理する）方式を入力処理に用いており[14]，動的情報もこの枠組みで処理されることが多い．本研究では静的・動的情報を入力と捉えないためこれらの方式とは根本的に異なるが，静的・動的情報管理部における処理はどちらかといえば集中管理を行う Open Agent Architecture 型に近い.

6.2 Web 技術との関連

W3C では静的・動的情報の管理や端末情報の通知のための標準が策定されている．CC/PP[15]はデバイスの構成やユーザの嗜好・プロファイルをネットワークを介してサーバに送信する際のデータ形式を規格化したものである．CC/PP によって送信元, 送信先の情報や端末情報等をどのようなフォーマットで送付するかが規定されるが, 動的なイベント通知のタイミングや情報参照の方法を提供するものではない．同様に W3C で策定された DCI は 3.2 節で述べたように本研究のベースとなる技術である．DCI では DOM ツリーによる静的・動的情報の管理や，プログラミング言語から属性を参照するための API，XML イベントによる属性変化の通知方法が規定されているが，具体的な MMI システムの構成は示されていない．本研究では DCI を参考に実際に静的・動的情報管理部を設計・構築し，アプリケーション開発を行った.

7．まとめ

Web ベース MMI システムにおいて静的・動的情報を利用するために静的・動的情報管理部を設計・開発し，音声対話ナビゲーションシステムとして実装するとともに, その有効性を実験的に検証した．文献[8]でも述べられているように，静的・動的情報を利用すると周辺状況やユーザの状態を推定できるため, よりユーザに適応した対話を実現できる．今後の課題としては，5 節の実験において被験者に指摘された静的・動的管理部の通信メッセージの理解し難さを改善するとともに，静的・動的情報を用いた発話文生成/スタイリングシステムを構築することが挙げられる.

8. 参考文献

[1] http://www.w3.org/TR/mmi-framework/

[2] http://www.w3.org/TR/mmi-arch/

[3] http://www.saltforum.org/

[4] http://www.voicexml.org/specs/multimodal/x+v/12/

[5] Katsurada, K., et al.: Reducing the Description Amount in Authoring MMI Applications, Proc. of INTERSPEECH2005, pp.873-876 (2005).

[6] Katsurada, K., et al.: A Modality Independent MMI System Architecture, Proc. of ICSLP'02, pp.2549-2552 (2002).

[7] Oviatt, S., et al.: Designing the User Interface for Multimodal Speech and Pen-Based Gesture Applications: State-of-the-Art Systems and Future Research Directions, Human-Computer Interaction, Vol.15, No.4, pp.263-322 (2000).

[8] http://www.w3.org/TR/voicexml20/

[9] http://www.w3.org/TR/DPF/

[10] Reithinger, N., et al.: SmartKom - Adaptive and Flexible Multimodal Access to Multiple Applications, Proc. of ICMI'03, pp.101-108 (2003).

[11] Johnston, M., et al.: MATCH: An architecture for multimodal dialogue systems, Proc. of the Annual Meeting of the Association for Computational Linguistics}, pp.376-383 (2002).

[12] Gibbon, D., et al. (ed.): Handbool of Multimodal and Spoken Dialogue Systems, Kluwer Academic Publishers, section2.2, pp.118-122 (2000).

[13] Seneff, S., et al.: Galaxy II: A Reference Architecture for Conversational System Development, Proc. of ICSLP98, pp.931-934 (1998).

[14] Delgado, R. L. and Araki, M.: Spoken, Multilingual and Multimodal Dialogue Systems, John Wiley & Sons, Ltd, section3.2.3, pp.67-70 (2005).

[15] http://www.w3.org/TR/CCPP-struct-vocab/