Vol.2009-SLP-78 No.8
2009/10/23

音声インタフェース普及促進のため開発支援技術

中　野　鐵　兵[†1]

音声インタフェースの普及に必要な技術として研究・開発が進められている，音声認識アプリケーションの開発支援技術について述べた．具体的には，音声認識技術を用いたユーザインタフェースの設計・開発・評価手法，音声認識システムを組み込んだアプリケーションの開発支援ソフトウエア，音声認識アプリケーションの開発・保守・運用で利用可能なサービスを紹介した．さらに，音声インタフェース普及の阻害要因の 1 つである，他のインタフェースに対する優位性のなさを解決するための手法として，簡易コントローラを援用した音声インタフェースについて述べ，パタン・ランゲージの形式で知見を共有可能となることを示した．また，他のインタフェースに対する優位性を検証するための客観的評価手法の具体例として，Time-AchievementRate Graph を用いた評価手法を紹介し，この評価手法によりシステム同士の特徴の差異を読み取ることが出来ることを示した．

Technologies for Encouraging Broad Use of Speech User Interface

Teppei Nakano[†1]

This paper presented development supporting technologies for speech recognition applications, which were necessary for the spread of speech user interfaces. These technologies included the following technologies. 1. Methodologies to design, develop, and evaluate speech user interfaces. 2. Software to support developments of speech applications. 3. Web-based services to develop, maintain and operate speech applications. Furthermore, the speech interfaces that quoted a facility controller as technique to solve lack of the superiority for the other interfaces, and the evaluation method to explain the difference of effectiveness were also presented.

†1 早稲田大学
Waseda University

1. はじめに

今日までに音声技術の実用化が進み，カーナビやゲームを通して以前よりも多くの人が音声インタフェースの使用を経験するようになったにも拘らず，民生分野において音声インタフェースが十分に普及したと言える状況には至ってない．このような，音声インタフェース普及の阻害要因の 1 つとしては，他のインタフェースに対する優位性のなさが挙げられている[1)]．すなわち，これまで利用者に与えられてきた音声インタフェースでは，GUI 環境におけるインタフェースやリモコンにおけるボタン操作と言った，従来のインタフェースに対して優位性を示すことが困難であり，積極的に音声インタフェースを利用したいと思わせることが出来ていなかった．このような問題を解決するためには，音声インタフェースを備えたアプリケーションの開発をより一般化させ，音声インタフェースをどのように使えば効果的なのか，提供した音声インタフェースに対して利用者はどのような振る舞いをするのか，またそれらをどのように評価すれば良いのか等，アプリケーション開発において重要な知見の収集と共有を進めることが重要である．本稿では，このような音声認識アプリケーションの開発をより一般化させるために必要な技術として研究を進めている，音声認識開発支援技術を紹介する．特に，音声認識アプリケーション開発支援技術の具体例として，音声インタフェースが他のインタフェースに対して優位性を持つために必要な，効果的な音声インタフェースの構築手法と評価手法について述べる．

2. 音声認識アプリケーション開発支援技術

音声認識開発支援技術に関する研究では，より高品質な音声インタフェースの，より多くの開発者による，より簡単な開発を可能にするために必要な技術の確立を目指している．音声認識アプリケーション開発支援技術を確立することで，開発者の新規参入を容易にし，音声認識アプリケーション開発のコミュニティの活発化を図る．

従来のグラフィカル・ユーザ・インタフェース (Graphical User Ingterface,GUI) を備えたアプリケーション開発と比較して，音声インタフェースを備えた音声認識アプリケーションの開発は，設計や利用方法の多様性，言語資源の必要性等から，難易度が高いものとなる．例えば，同じ音声インタフェースと言っても，機能選択のための音声インタフェースであったり，ディクテーションやデータ入力のための音声インタフェース，または音声対話システムであったりと，それぞれ全く違ったインタフェース設計が必要となる．さらに基本的な入出力方式が定義されておらず，標準的な使い方も確立されていない．例えば GUI の場

Vol.2009-SLP-78 No.8
2009/10/23

合，スクリーンデバイスによる情報提示，ポインティングデバイスによる項目選択，キーボードデバイスによるデータ入力，と言った共通の概念が存在する．それに対して音声インタフェースの場合，このように抽象化してインタフェースを表現することが出来ない．さらに，その結果，実利用環境における実ユーザの振る舞いを正確に予測することが困難となり，音声認識エンジンの性能・特性とアプリケーション設計とのミスマッチ，あるいは開発サイドの期待した使い方と利用者の望む使い方のミスマッチが発生してしまう．また，音声認識システムが使用する言語資源としては，読み情報を含んだ語彙情報，言語モデルや文法，音響モデル等があり，開発者はアプリケーションに適したこれらのモデルを用意しなくてはならない．これらの作業には音声認識技術に関する高い専門性が求められ，適切なモデルの構築には時間もコストも要求される．さらに，語彙情報のように，インタフェースの性能・性質に直接関係する言語資源の場合，そのモデルは一度構築したら完了というものではなく，継続的な維持・拡張が求められる．

このような問題に対する音声認識アプリケーション開発支援技術として，以下の分野に関してそれぞれ独立した単独の技術として確立すると同時に，効果的に統合された枠組みの実現を目指している．

- 音声認識技術を用いたユーザインタフェースの設計・開発・評価手法
- 音声認識システムを組み込んだアプリケーションの開発支援ソフトウエア
- 音声認識アプリケーションの開発・保守・運用で利用可能なサービス

例えば，音声インタフェースの設計手法としては，どのような環境で，どのような利用者が，どのような制限で音声を利用可能かを明確にした上で，それぞれに適切な設計をパタン・ランゲージ[2] を用いて統合するインタフェースの設計手法を提案している．パタン・ランゲージとは，ある状況下で繰り返し発生する問題と，熟練者によって得られるて解決策のセットであるパターンの集合であり，特定の分野で発生する複数の問題に対して一般的で抽象的な解法を提供する．全てのパターンには名前が付けられ，通常どのパターンも同一のフォーマットで記述される．また，それぞれのパターンはその問題の前提条件や解法によって新たに発生する問題によって互いに関係を持ち，全体の問題領域に対して最適な解法として体系立てる．音声認識アプリケーション設計のためのパタン・ランゲージの場合，問題をマルチモーダルインタフェース設計として捉え，問題の前提条件として，ポインティングデバイスの有無，キーボードデバイスの有無によってパターンをそれぞれ記述する．また，孤立単語認識を利用した Command and Control や，大語彙連続音声認識を利用した音声対話システム等，様々なアプローチをそれぞれ別の問題に対する別のパターンとして記述す

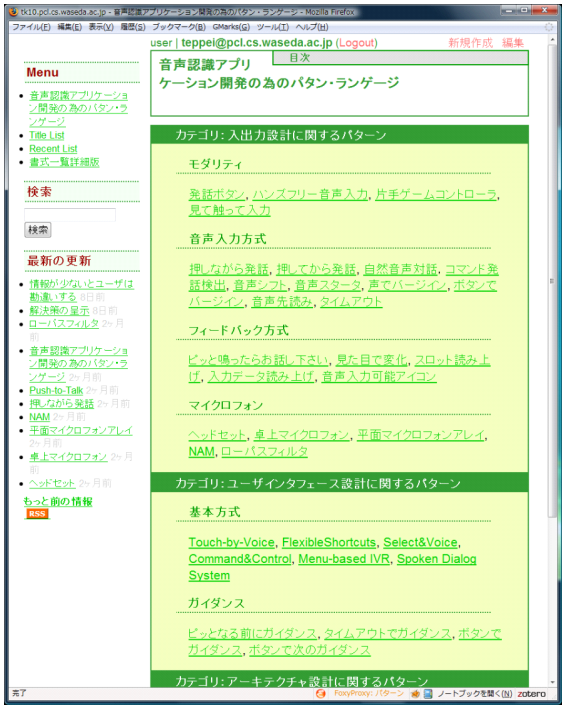

図 1 パタン・ランゲージを用いた設計指針の知見共有サービス

る．さらにそこから発生する別の問題に対する解法をさらにそれぞれ別のパターンとして記述することで，音声認識アプリケーション設計のための共通の解法として提供することが可能となる．また，解法をパターンとしてまとめそれを知見として共有することで，特定の特徴をもったインタフェースに明確な名前が定義されるようになり，開発者間のコミュニケーションコストの低下を実現する (図 1).

また，アプリケーションの開発支援ソフトウエアとして，音声認識アプリケーションにおける汎用的な機能拡張の枠組みを備えた新しいプラットフォームである，Proxy-Agent と呼ばれるミドルウエアを提供している[3]．Proxy-Agent を用いることで利用者の実際の振る舞いのモニタリング機能が効率的かつ効果的に実現され，ランタイム (アプリケーション利用時) のユーザの振る舞いに関するデータの開発サイドへのフィードバックが可能となる

Vol.2009-SLP-78 No.8
2009/10/23

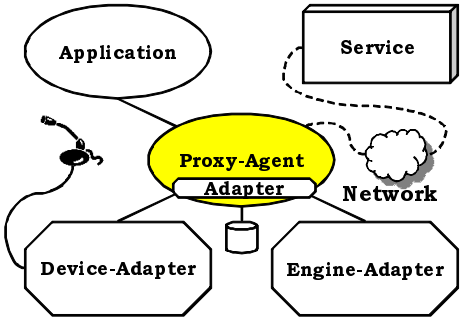

図 2 Proxy-Agent 概要

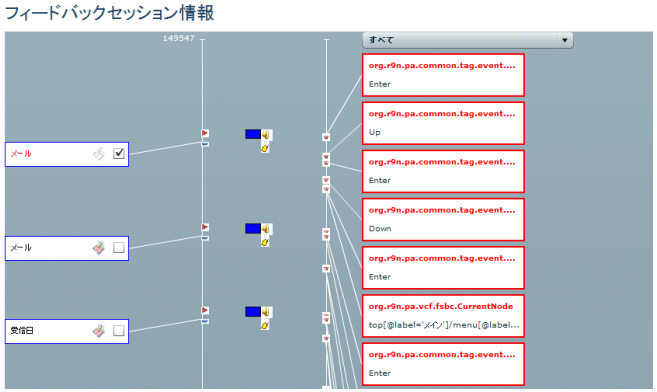

図 3 フィードバックデータの動作解析サービス

(図 2)．フィードバックデータの動作解析サービスを用いることで，ユーザからのフィードバックに基づいてどのように認識器を改良するべきか，あるいはどのようにユーザに改善指示を出すべきかを判断するための支援を行う (図 3)．これらによって，開発サイドでのシステムの改良，ユーザからのデータフィードバックに基づくシステムのチューニングなどの影響が利用者側に対し再度フィードバックされる枠組みが実現され，ユーザが常に最適な状態で音声認識システムを利用できるようになることを可能とする[4]．

音声認識アプリケーションの開発・保守・運用で利用可能なサービスとしては，音声認識アプリケーション開発における最も重要かつ困難な作業の一つである，システムが認識可能な語彙の適切な設計と，実際に利用されている語彙のメンテナンスに関するサービスを提案

図 4 ウェブベースで提供される語彙情報サービス (左：トップ画面，右：語彙の検索結果表示画面)

している[5]．このサービスでは，あらゆる分野の語彙情報がオンラインデータベース上に一元化され，共有のウェブサービスとして提供される (図 4)．集合知を実現するための枠組みを備え，語彙に関する情報の一元管理と半自動更新を可能にし，語彙開発の作業コスト削減と品質向上を可能にする．

3. 音声インタフェースの設計手法の具体例

他のインタフェースに対する優位性を持った音声インタフェースの具体的例として，簡易コントローラを援用した音声インタフェースを紹介する．

この手法は，次のような**文脈・前提条件**が与えられたときに有効である．

- 運転中や移動中でも利用可能なシステムを構築するために，音声入力を用いた効率的なインタフェースを構築したい．
- この場合，ポインティングデバイスやキーボードのような視覚フィードバックが必要なデバイスの利用は困難であるが，完全なハンズフリー環境を必要としている訳ではなく，簡単な操作も可能である．

特に，下記の**問題**を解決する．

- 音声入力の不正確性に起因する，ユーザビリティの低下にどう対処すべきか？
- 音声認識に要する処理の遅延を如何に抑えるべきか？

具体的な **解決策**を，下記に示す．

- 複数のボタンを備えた，片手で操作可能な小型コントローラを利用し，敏速かつ正確性

Vol.2009-SLP-78 No.8
2009/10/23

図 5 コントローラ

が要求される操作をコントローラで行い，音声入力の不正確性を補完する (図 5).

- 音声入力は，コントローラでは入力操作が困難な個所で使用する．
- コントローラとしては，ゲームで使用されているような一般的なボタン (上下左右と 2~6 個程度の押しボタン) を備えたものとし，その操作の習得に練習をほとんど必要としないものとする．
- ボタンの数やボタンに対する機能の割り当ては，標準的なものがあればそれに従う．

この手法により，下記のような **効果・結果**を得る．

- 発話タイミングの指定ややり直し発話等，音声入力に関する処理をコントローラで，データ入力のみを音声入力でと，必要に応じて処理を分割できる．
- 必ずボタン操作が必要となるため，ハンズフリー音声認識の利用はできなくなる．
- コントローラの操作が複雑になったり標準的な操作と異なると，ユーザビリティが大きく低下する．
- コントローラに慣れていないユーザの場合，コントローラの操作ミスにより，期待通りの操作ができなくなる．

提案手法では，このようにある前提条件が与えられたときに適したインタフェース設計を，前記のようなパタン・ランゲージ[2] を用いて記述する．さらにこのパターンを複数用意し，それぞれを **関連するパターン**として統合することにより，誰がいつどのように音声インタフェースを使用すれば良いかに関する知見をまとめる．

このパターンの具体的な**実装例**としては，機能選択用のインタフェースである Flexible Shortcuts[6] とデータ入力用のインタフェースである Select&Voice[7] を組み合わせた，片手で操作が可能な簡易コントローラを併用するインタフェースがある (図 6, 図 7)．このイ

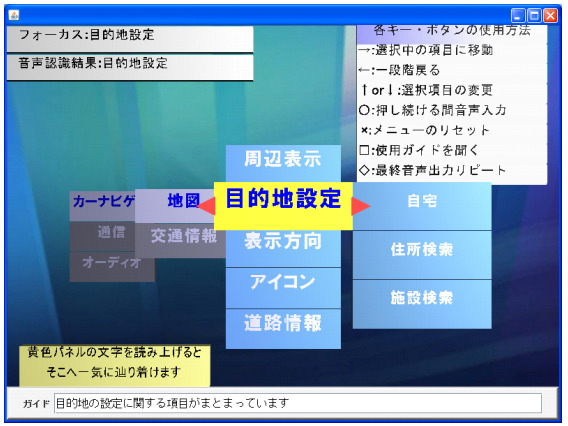

図 6 機能選択用インタフェース

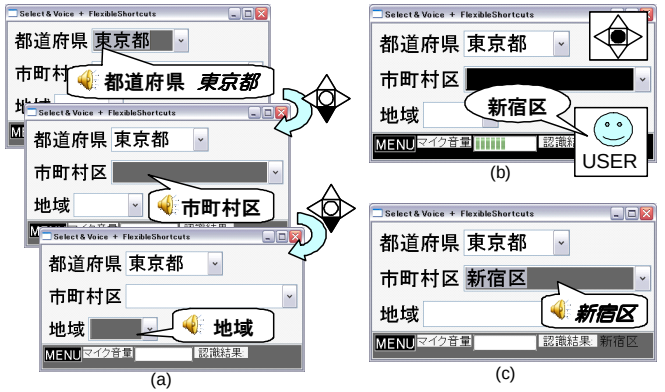

図 7 データ入力用のインタフェース．(a) 上下ボタンで項目を選択，システムは選択された項目名と入力されている値を読み上げる．(b) 発話ボタンを押しながら選択した項目に対し音声入力をする．(c) 認識結果を表示し，読み上げる．

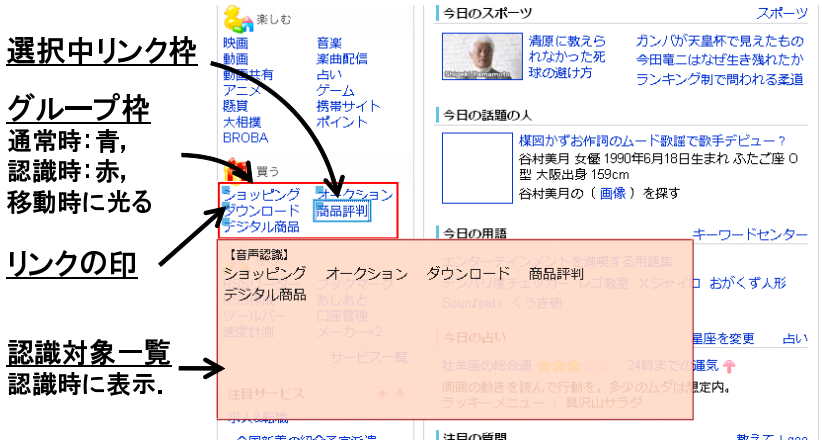

図 8　ボタンと音声を用いた Web リンク選択インタフェース

ンタフェースでは，コントローラを用いて素早く正確な項目選択を行いながら，音声を用いて複数のキーワードや項目の値を入力する手段を提供することで，初心者にも利用可能でありながら，熟練者にとっては効率的に操作が可能な新しいインタフェースを実現している．

また，別の実装例として，複数のグループに分割されたウェブページからコントローラを用いてグループの選択を行い，グループ内の項目 (リンク) の選択を音声を用いて行うインタフェース[8] がある．このインタフェースでは，グループの選択をコントローラで行うことで注目領域の選択と，そこで利用可能な語彙の絞り込みを同時に行うことが出来る (図 8)．語彙が絞り込まれることで，音声認識の精度・速度が向上し，ユーザは効率的な項目選択を実現している．

4. 新しい客観評価手法の具体例

他のインタフェースに対する優位性を検証するための客観的評価手法の具体例として，Time-AchievementRate Graph(以下 T-A グラフ) を用いた評価手法を紹介する．T-A グラフは，客観的な評価尺度を与えるために考えられた，時間に対する平均タスク達成率の変化を描いたグラフである．このグラフを利用する際は，実験時に被験者に制限時間を設けず，実験後にある時間の中で達成されたタスクの割合を算出する．T-A グラフを用いることで，システム同士の特徴の差異を読み取ることが出来る．

一般に，平均タスク達成率を算出する時，適切な制限時間が必要となる．しかし，アプリケーションやタスクによって，適切な制限時間は異なる．もし，ある適当な時間制限を定義し，それによって得られた平均タスク達成率やタスク達成時間と言った標準的な評価を行った場合，制限時間にに左右される情報は失われる恐れがある (もしくは分散として与えられ，差異がわかりにくくなる)．また，定められた制限時間ぎりぎりで達成されたタスクと，制限時間以内に達成されなかったタスクをはっきり区別するのが妥当であるのか，どのように取り扱えばよいかという議論も発生する．以上のような問題を解決するために，制限時間を変数とした，T-A グラフ用いる．

T-A グラフは，仕事の達成率を，制限時間の関数として描かれるグラフである．この手法を利用する際は，被験者には制限時間を与えず，達成するまでタスクに取りかかる．そのかわりに，タスクに対していつでもギブアップすることを許可する．実験終了後，平均タスク達成率は制限時間によって計算され，X 軸を時間，Y 軸を達成率とした直交座標系としてプロットする．最後に，グラフの形状を見ることで，そのシステムの特性や有効性を確認する．

このグラフは単調増加の関数であり，y の最大値は 1.0 となる．T-A グラフの例を図 9 に示す．T-A グラフの中で，ある時間 τ 与えた時の，達成率を A_τ と定義する．例えば図 9(上) では，制限時間が 30[秒] の時のタスク達成率は 0.66 として得られる ($A_{30} = 0.66$)．また，制限時間をなしとした時の達成率を，最大タスク達成率 A_∞ と定義する．例えば図 9(上) では，最大タスク達成率は 0.85 として得られる ($A_\infty = 0.85$)．最大タスク達成率は，ギブアップが行われる確率の低さを表す．またタスク達成時間の上位 α のうちの最大のタスク達成時間を T_α と定義する．これを用いると，平均タスク達成時間は，$\overline{T}_\alpha = \frac{1}{\alpha}\int_0^\alpha T_a da$ で求められる．これは，曲線と，$y = T_\alpha$ で囲まれた領域の面積を意味する．例えば図 9(上) では，上位 80%の最大タスク達成時間は 81 秒 ($T_{0.8} = 81$) であり，平均タスク達成時間 ($\overline{T}_{0.8}$) は図中の斜線部分の面積に比例する形で得られる．

T-A グラフは，複数の異なるシステムの特性の差異を比較するために用いることが出来る．比較の図を T-A グラフの例を図 9(下) に示す．このグラフでは，$A^X(\tau)$ と $A^Y(\tau)$ の 2 つが比較されている．システムの有効性は，このグラフを比較することで評価される．注目する点は 2 つ観点から得られる．一つは制限時間の観点であり，もうひとつは達成率の観点である．前者に注目するときは，ある時間内に達成されたタスクの数が重要な場合である．例えば，解析者が，短時間に達成できるタスクの数に着目したいときなどが挙げられる．このようなときは，ある時間 t に対する達成率 A_t を比較することが有効である．例と

Vol.2009-SLP-78 No.8
2009/10/23

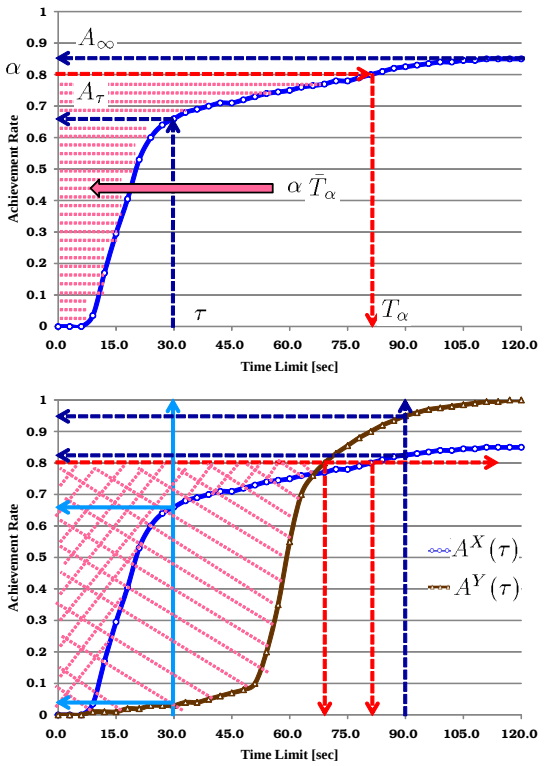

図 9 Time Achievement グラフの例. (上) 値の定義. (下) 比較の例.

して図の T-A グラフにおいて，制限時間を 30 秒と定めた場合の A^X_{30}，A^Y_{30} を比較すると，A^X_{30} の方が大きい．よってごく短い時間においてはシステム X の方が有効であることが分かる．後者は注目するときは，全体的な効率が重要な場合である．例えば，全体の 8 割のタスクを達成するための平均の所要時間に着目するときなどが挙げられる．このようなときは，$\overline{T}_\alpha$ に注目することが有効である．図のグラフを例にすると，T^X_α と T^Y_α では T^Y_α の方が小さいが，$\overline{T}^X_\alpha$ と $\overline{T}^Y_\alpha$ では $\overline{T}^X_\alpha$ の方が小さい．

注目すべき点を決めれば，各被験者ごとの実験結果を利用することで，ANOVA などの標準的な統計的検定を実行する．全ての値は同条件下における平均値として与えられるため，これらの要素は，制限時間とは無関係な公平な平均達成時間を得るために有効である．

5. むすび

音声インタフェースの普及に必要な技術として研究・開発が進められている，音声認識アプリケーションの開発支援技術について，以下の 3 つの分野の紹介を行った．

- 音声認識技術を用いたユーザインタフェースの設計・開発・評価手法
- 音声認識システムを組み込んだアプリケーションの開発支援ソフトウエア
- 音声認識アプリケーションの開発・保守・運用で利用可能なサービス

音声インタフェース普及の阻害要因の 1 つである，他のインタフェースに対する優位性のなさを解決するための手法として，簡易コントローラを援用した音声インタフェースを具体例として紹介し，パタン・ランゲージの形式で知見を共有方法を示した．また，他のインタフェースに対する優位性を検証するための客観的評価手法の具体例として，Time-AchievementRate Graph を用いた評価手法を紹介し，この評価手法によりシステム同士の特徴の差異を読み取ることが出来ることを示した．

参考文献

1) 古井貞熙ほか：音声認識技術実用化に向けた先導研究，NEDO 平成 17 年度成果報告書 100007350，早稲田大学 IT 研究機構 (2006).
2) Alexander, C.: *A Pattern Language*, Oxford University Press (1977).
3) Nakano, T., Fujie, S. and Kobayashi, T.: Extensible speech recognition system using proxy-agent, *ASRU2007*, pp.601–606 (2007).
4) 中野鐵兵，小林哲則：Proxy-Agent を核とした双方向型音声認識アプリケーション開発支援の実現，情報処理学会研究報告. SLP, 音声言語情報処理， Vol.2009, No.10, pp. 63–68 (2009).
5) 中野鐵兵，佐々木浩，藤江真也，小林哲則：集合知を利用した語彙情報の収集・共有・管理システム (音声言語処理)，情報処理学会研究報告. SLP, 音声言語情報処理，Vol.2008, No.46, pp.77–84 (2008).
6) Nakano, T., Kumai, T., Kobayashi, T. and Ishikawa, Y.: Design and Formulation for Speech Interface Based on Flexible Shortcuts, *Interspeech 2008*, pp.2474–2477 (2008).
7) 梅本暁，中野鐵兵，小林哲則：GUI とのアナロジーに基づいた音声インタフェースの提案と評価，日本音響学会 2007 年秋季研究発表会，Vol.2-3-5 (2007).
8) 秋元啓孝，中野鐵兵，小林哲則：音声による Web リンク選択インタフェースの検討，情報処理学会全国大会講演論文集 (2009).