社団法人 情報処理学会 研究報告
IPSJ SIG Technical Report

2007－SLP－67 (20)
2007／7／21

アプリケーションの機能構造に基づく音声インタフェースの提案と評価

熊井朋之† 中野 鐵兵† 小林 哲則† 石川泰‡
†早稲田大学,‡三菱電機株式会社

あらまし:
複数の機能の選択肢から，特定の機能の容易かつ効率的な選択を可能にする音声インタフェースとして，“機能構造” と” 連続キーワード入力” を用いたインタフェースを提案する．機能構造とは，システムで実行可能な機能を木構造で表現したもので，提案手法では，これを音声操作のための言語資源として利用する．連続キーワード入力とは，複数のキーワードを連続して一度に入力するための枠組みであり，意図する機能への効率的な到達を可能にする．本稿では，このインタフェースの定式化を行う．また，提案手法の有効性を確認するために開発したアプリケーションと，音声コマンドをすべての機能に対して一意に決定し，状況に依存せず，全コマンドを利用する従来のインタフェースとの比較実験を紹介する．実験では，タスク達成率と主観評価の結果で，提案手法の優位性が確認できた．また，アプリケーションを初めて利用するユーザでも，提案手法では容易に操作可能であることが示された．さらに，頻度の多い操作の場合には，連続キーワード入力によって効率的なコマンド選択が可能であることも示された．

A Proposal and Evaluation of New Speech User Interface Based on Functional-Structure

Tomoyuki KUMAI†, Teppei NAKANO†, Tetsunori KOBAYASHI†, Yasushi ISHIKAWA‡
†Waseda University, ‡Mitsubishi Electric corp.

Abstract:
As a speech user interface that support easy and effective selection of expected functions, we propose a new speech interface using “functional-structure” and “continuous keyword input”. “functional-structure” is a tree structure of executable functions of the system, and used as a linguistic resource. “continuous keyword input” is a framework to input a series of keywords at a time. Probabilistic formulation is also presented. Experimental result show that our approach is superior to traditional approach, command and control approach, in terms of accuracy for experts and novices.

1 はじめに

複数の機能の選択肢から，特定の機能の容易かつ効率的な選択を可能にする音声インタフェースを提案し，その検証結果を報告する．音声インタフェースは，人の精通度の高さや習得・利用の容易さ，入力速度の速さ，他のデバイス・操作との共存可能性など備え，HCI における有効な入力手段の１つとして捕らえられる．機能選択に音声インタフェースを用た例は，テレフォニーシステムでのメニュー選択，ハンドヘルド端末用アプリや携帯電話，カーナビゲーションシステムでのコマンド選択等，様々な分野で多数報告されている [2][8][3][1].

これらのシステムは，ユーザの自由な発話を認め，広い範囲の音声入力をシステムが処理してナビゲーションを行うスタイルと，使用可能な語彙や文法をシステムがユーザに明示し，与えられた自由度の中で発話された音声入力をシステムが処理するスタイルのインタフェースに分類される．ここでは，前者のユーザインタフェースを自然言語スタイル (natural-spoken-language style, NSL Style)[5]，後者をシステム指向言語スタイル (system-oriented-language style; SOL Style) と呼ぶ．このうち本研究では，期待される音声認識エンジンのパフォーマンスや，実装・実用化の容易性，インタフェースに対する人の適応力の観点から，SOL Style のユーザインタフェースを検討する．

SOL Style のユーザインタフェースでは，そのアプリケーションで利用可能な語彙や文法，発話様式をまとめた部分集合言語 [6] が定義される．従来の機能選択用音声インタフェースの多くは，孤立単語認識用の特定のコマンド語彙の集合や，複数の小さな文脈自由文法が定義されるが，部分集合言語はこれらを包含するものと考える．一般に，部分集合語彙は NSL Style の語彙セットと比較して小さな集合となるため，音声認識エンジンの誤認識の問題 [7] の影響が小さく見積もられる．

Stifelman et al. は，特定の操作を直接マッピングした語彙をコマンドとして定義し，音声メモアプリケーションの音声による機能選択を可能にした [3].

このアプローチはCommand and Controlと呼ばれ，実装の容易さや，語彙数が少なく誤認識が起きにくいことから，多くの音声アプリケーションで同様の手法が用いられている．しかしながら実際には，開発者が用意した語彙とユーザが実際に利用する語彙にミスマッチが起こりやすく，ユーザが何と発話すれば良いのか分からないという問題が発生しやすい．語彙数の増大によるミスマッチの解決手法も考えられるが，孤立単語認識では，語彙数が認識率に対して与える影響が大きく，それだけでは有効な解決手段となりにくい [1]．

コールセンターの自動双方向音声応答システム (Interactive Voice Response System; IVR System) のように，階層化されたメニューから，意図する機能に到達するために複数の項目を順に選択していく音声インタフェースも多く採用されている [8]．このインタフェースでは，メニューを選択する毎にそのサブメニューが提示され，逐次的にサブメニューを選択していく．そのため，サブメニュー毎に利用可能な語彙がわかりやすく，待ち受け語彙も少ないため誤認識の問題も起きにくい．また，毎回メニューが提示されるインタフェースの場合は，利用するための学習も必要としない．しかしながら，メニューを逐次選択していくために，音声の利点である入力の速さが活かしきれない．また，メニュー構造によっては，ユーザが意図する機能がどのメニューに含まれているかの推定が困難となり，ユーザビリティは低くなりやすい [9]．

本研究では，木構造で表されたアプリケーションの “機能構造” を言語資源として利用し，“連続キーワード入力” によって意図する機能への到達を可能にする音声インタフェースを提案する．この手法では，目的とする機能へすばやく到達することが可能な Command and Control の利点と，逐次的に選択していくメニューベースのインタフェースのわかりやすさと学習の容易性を同時に実現する．さらに，あいまい性を回避するための枠組みにより，音声認識システムの誤認識問題に対してシームレスな解決手段を提供する．

本稿では，まず 2 節において、提案手法の基本的なアプローチを述べ，3 節でそのモデル化・定式化を行う．4 節で提案モデルの実装として開発した実験用アプリケーションについて述べ，5 節以降で実験の結果を報告する．

2 基本アプローチ：機能構造と連続キーワード入力

音声インタフェースを備えたアプリケーションでは，ユーザは特定の機能の実行を意図した発話を行い，その認識結果に従ってシステムが実行する機能を選択する．このとき，ユーザにどのような発話を許すかを決定するのが部分集合言語であり，本稿で提案する手法では，アプリケーションの機能構造からこれを定義する．機能構造とは，木構造で表現されたシステムで実行可能な機能の構造を表し，以下のように定義される．

- 各々のノードには 1 つ以上の名前 (キーワード) が与えられている
- 葉ノードにはそれぞれ呼び出し可能な機能が割り振られている
- 葉以外のノードには機能の種類やカテゴリなどの分類が割り振られている

機能構造を用いると，1 つの機能を複数のキーワード系列で表現することができる．また，キーワードを指定することで機能の絞込みが可能となる．メニューベースのインタフェースのメニュー構造は，機能構造の実現例の 1 つと考えられる．

機能構造に基づくユーザインタフェースでは，ユーザがキーワードの系列を与えることで特定のノードが選択される．ユーザは現在選択されているノードという観点で状態を保持し，選択されたノードに対して状態遷移を行う．遷移先のノードが葉ノードであれば，そのノードに関連付けられた機能が選択される．遷移先のノードが葉以外のノードであれば，その子ノードに関連付けられたキーワード群がユーザに提示される．入力されたキーワードの系列から選択可能なノードが複数存在する場合，システムは遷移先のノードの明示的な選択をユーザに要求する．

部分集合言語は，機能構造から得られるキーワードの，連続キーワード入力が可能となるように定義される．この枠組みを用いて特定の機能を一意に表すキーワード系列を 1 度に入力すると，Command and Control と同様，特定機能の音声による直接呼出しが可能となる．それに対して，子ノードのキーワードを逐次的に入力していくと，IVR システムのような，対話的な機能選択が行われる．また，実行を意図する機能が機能構造のどこに位置するか不明確な場合は，その機能に関連するキーワードを複数適当に入力してみることで，該当の機能の検索が可能となる．さらに，特定の機能を選択する為に複数のキーワードの入力を繰り返すことで，ユーザは特定の機能を一意に選択するためのキーワード系列を学習し，Command and Control のような操作が可能となることが期待される．

このように，機能構造に基づく連続キーワード入力をサポートしたインタフェースを提供することで，特定の機能の容易かつ効率的な選択が可能となる．

3 モデルとアルゴリズム

提案手法では，システムの機能構造を言語知識として動作する音声認識システムを利用し，連続キーワード入力により得られたキーワード系列から選択対象となるノードを決定する．すなわち，音声入力による実行コマンドの推定問題が，機能構造上の移動先ノードの決定問題として置き換えられる．ここではまず，木構造上のノードを音声入力によって移動させるモデルにおいて，次の移動先のノードを音声入力から決定する問題として定式化を行う．

3.1 定式化

音声入力を X，現在のノードを n_s，移動先のノードを n_g とすると，この問題は $P(n_g|n_s, X)$ を最大

にする最尤な n_g^* を求める問題として捉えられる．

$$n_g^* = \underset{n_g}{\operatorname{argmax}} P(n_g|n_s, X) \quad (1)$$

(1) 式はベイズの定理により，

$$n_g^* = \underset{n_g}{\operatorname{argmax}} P(X|n_g, n_s)P(n_g|n_s) \quad (2)$$

のように変形される．また，n_s から n_g に到達するために入力可能な単語列の集合を $\Omega = \{W_i\}$, (W_iは単語列, $i < C, C$ は全ての単語から構成される単語列の組み合わせの数) とすると，(1) 式はさらに以下のように変形される．

$$n_g^* = \underset{n_g}{\operatorname{argmax}} \sum_i P(X|W_i)P(W_i|n_g, n_s)P(n_g|n_s) \quad (3)$$

(3) 式では，第一項が音響モデル，第二項が状態・意図依存の言語モデル，第三項が状態依存の意図モデルを表す．この式をそのまま解くことは困難であるため，ここでは

$$P(W_i|n_g, n_s) = \frac{P(n_g|W_i, n_s)P(W_i|n_s)}{P(n_g|n_s)} \quad (4)$$

であることを用いて，(3) 式をさらに (5) 式のように変形する．

$$n_g^* = \underset{n_g}{\operatorname{argmax}} \sum_i P(X|W_i)P(W_i|n_s)P(n_g|W_i, n_s) \quad (5)$$

(5) 式では，第一項が音響モデル，第二項が状態依存の言語モデル (状態依存言語モデル)，第三項が状態と手段 (入力可能単語列) 依存の意図モデルを表す．今，(5) 式の第一項と第二項から得られる事後確率の N-Best を与える単語列の集合 $\Omega' = \{W_j'\}$, ($\Omega' \subset \Omega, j < N$(from N-Best)) を考える．

$$\Omega' = \underset{W_i}{\operatorname{nbest}} P(X|W_i)P(W_i|n_s) \quad (6)$$

このとき，(6) 式から得られる W_j' を用いて (5) 式を以下のように近似する．

$$n_g^* \simeq \underset{n_g}{\operatorname{argmax}}\{\max_{W_j'} P(X|W_j')P(W_j'|n_s)P(n_g|W_j', n_s)\} \quad (7)$$

また，(5) 式の第三項を，意図関数として適当な関数 $f(n_s, n_g, W_i)$ で定式化し，W_j' の尤度を尤度関数として $L(W_j')$ と表すと，n_g^* は以下の式で与えられたものとなる．

$$n_g^* \simeq \underset{n_g}{\operatorname{argmax}}\{\max_{W_j'} L(W_j')f(n_s, n_g, W_j')\} \quad (8)$$

すなわち，(1) 式で与えられた問題の最尤な解は，(9) 式を最大にする n_g^* として近似的に得られる．

$$score(n_g) = \max_{W_j'} L(W_j')f(n_s, n_g, W_j') \quad (9)$$

特に，N-Best として 1-Best を用いた場合，n_g^* は以下のような計算で算出が可能となる．

$$W^* = \underset{W_i}{\operatorname{argmax}} P(X|W_i)P(W_i|n_s) \quad (10)$$

$$n_g^* \simeq \underset{n_g}{\operatorname{argmax}} f(n_s, n_g, W^*) \quad (11)$$

3.2 アルゴリズム

前述の定式化に従って作成したアルゴリズムを以下に示す．ここでは，ユーザのフィードバックを前提としたあいまい性解決の枠組みも導入される．あいまい性は，音声認識の認識結果に加え，移動先のノードの候補にも含まれる場合がある．しかしながら提案する手法では，いずれのあいまい性も同一の枠組み内でシームレスに解決することが出来る．

1. 現在のノード n_s から状態依存言語モデル $P(W_i|n_s)$ を決定
2. ユーザから連続キーワードの入力を受け付ける
3. $P(W_i|n_s)$ を用いて入力音声から候補となる単語列の集合 Ω' を選択
4. 全てのノードに対して (9) 式で与えられるスコアを計算．
5. ある閾値 θ を超えるスコアのノードを全て選択し，その数 k に従って以下の処理を行う
   - $k < 0$ の場合　候補がなかったとして，再度入力を促して終了
   - $k = 1$ の場合　そのノードを n_g^* として選択し終了
   - $k > 1$ の場合　スコアに応じて結果をソートし，候補をユーザに提示．ユーザの選択結果を n_g^* として選択し終了

4 実験用アプリケーションの開発

提案モデルに従って，音声インタフェースを備えた GUI アプリケーションの開発を行った．概観を図 1 に示す．このアプリケーションは，機能構造を表す XML に従って選択肢を表示する．ユーザはタッチペンか，連続キーワード音声入力を用いて項目を選択する．画面中央部に機能構造の表示領域が用意され，現在選択されているノードが開かれた状態で表示される．開かれたノードは子ノードの一覧が表示される．現在の状態の表示と認識可能単語の提示は，このインタフェースを用いて行われる．機能構造の表示領域の下には，認識結果，一つ前の状態，その他メッセージを表示するテキストボックスを置いた．その下に音声認識を開始するボタン，一つ前の状態に戻すことが出来る Undo ボタン，Undo 動作を取り消すやり直しボタン，ルートノードまで戻るリセットボタンを置いた．音声認識は逐一ボタンを押すことで開始する．音声認識エンジンには，Sphinx-4[10] を当研究室で日本語用に拡張したものを用いた．また，あいまい性の解決に必要な候補は，選択肢をリスト表示にしたダイアログを用いて表示し，そのダイアログからユーザが最終的な移動先を選択するようにした (1(c))．

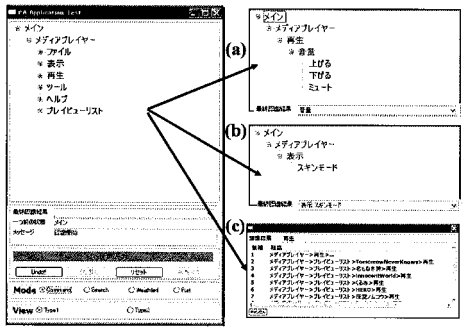

図 1: GUI Overview:(left) child-nodes are listed, (right) After the state transition; (a) with single keyword - grand-child nodes are listed, (b) with two continuous keywords - the command is selected, (c) with ambiguous keyword - confimation dialog is shown

4.1 音声認識とスコア関数

今回のシステムでは，N-Best は用いずに，1-Best でアルゴリズムを実装した．そのために，スコア関数に尤度は必要なく，以下の式で求められるコストを利用した意図関数の値のみを用いる．

コスト関数 現在のノードから移動先候補のノードに至るのに，木を下る回数を d_{n_s}，登る回数を u_{n_s}，またルートノードから現在地に至るのに木を下る回数を d_{n_o} 回とする．この時，(i) $u_{n_s} = 0$ ならば

$$cost(n_g) = 2 - 0.5^{d_{n_s}-1} \tag{12}$$

(ii) $u_{n_s} > 0$ ならば

$$cost(n_g) = \begin{cases} 3-0.5^{d_{n_o}} & \text{if } u_{n_s} > 2 \\ u_{n_s} + 1 - 0.5^{d_{n_s}} & \text{otherwise} \end{cases} \tag{13}$$

4.2 状態依存言語モデル

状態依存言語モデルとして，実験用に 2 種類のモデルを利用可能にした．

順序制約付文脈自由文法 このモデルでは，機能構造上の現在のノードの子孫ノードのみを待ちうけ語彙とした文法が用いられる．文法はノード毎に用意される．連続入力可能なキーワードはノードの親子順序の制約を受けるが，途中のノードのキーワードを省略可能とした (図 2)．このモデルでは，自由度はあまり高くないが，探索範囲を狭めることで，認識可能単語を絞り込み，認識精度を上げることが出来る．また，操作対象が木構造であることをユーザに強く印象付けることを期待する．

孤立単語認識モデル このモデルでは，現在のノードの子供のノードのみを待ち受け語彙とし，孤立単語認識を行う．連続キーワード認識をサポートせず，従来の音声インタフェースを模擬する為に用いる．

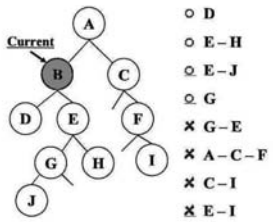

図 2: Order-Constrained CFG example

4.3 機能構造

実験用の機能構造として，音声チャット用アプリケーション (Skype™) と，メディア再生用アプリケーション (Windows Media Player™) のそれぞれのメニューの構造に従い構築した．各ノード名も，それぞれのメニューで使用されている単語をそのまま用いた．

比較対象として，用意した機能構造を元にして，全てのコマンドが並列に置かれている音声インタフェースを構築した．これは，全ての機能が葉として根の直下に配置されているものである．各コマンド名は，木構造での同等のコマンドの名称に準拠した．但し，「終了」など，複数のアプリケーションにおいて同様のコマンド名を持つものについては，「アプリケーション名+終了」などのコマンド名にし，コマンド名の重複は避けた．以降これをフラット構造と呼び，通常の木構造の機能構造と区別する．

5 フラット構造のインタフェースとの比較実験

提案手法の有効性を検証するため，前述のアプリケーションを用いて，木構造の機能構造を持つアプリケーションと，フラット構造のアプリケーションの比較実験を行った．状態依存言語モデルは，機能構造を持つアプリケーションでは順序制約付文脈自由文法を，フラット構造のアプリケーションでは孤立単語認識モデルを用いた．実験では，1 人あたり 2 つのインタフェース，1 つのインタフェースにつき 2 セッション，1 セッションあたり合計 24 問のタスクを課した．また，1 つのインタフェースが完了すると，被験者の主観的評価を得るためにアンケートを取った．インタフェースの主観的評価を測る方法として，Hone らが提案した音声インタフェースについての評価法である SASSI[11] の質問群を日本語化して用いた．

5.1 タスク設定

被験者は各タスク毎に 1 つの機能を実行するように指示を受け，音声もしくはタッチペンを用いて機能の選択を行う．選択する機能の提示はアプリケーションが自動で行い，被験者ごとのタスク内容や順序は変更しない．

タスクは，被験者から最初に選択された機能が，設問の指示と合致している場合に成功とし，それ以

外は全て失敗とした．タスクのやり直しは認めず，制限時間も設けていない．

タスクの完了時間は，問題文を表示してからある葉ノードに辿り着くまで時間とした．問題文を見てから，該当するコマンド名を考える時間も完了時間に含まれる．

設問は，以下に示す3つのカテゴリに分けて設計した．カテゴリ1に属する問題が10問，カテゴリ2に属する問題が10問，カテゴリ3に属する問題が4問，計24問でタスク1セットを構成した．

カテゴリ1 該当するコマンド名が問題文中にほぼそのまま現れる．ユーザが想像するコマンド名と実際のコマンド名が一致する場合に相当．
- 例：「再生速度を速くしてください」
正解は，TREEでは「メディアプレイヤー＞再生＞再生速度＞速く」，FLATでは「再生速度速く」

カテゴリ2 該当するコマンド名の問題文からの類推が必要．ユーザが想像するコマンド名と実際のコマンド名が異なる場合に相当．
- 例：「映像をディスプレイいっぱいに広げてください」
正解は，TREEでは「メディアプレイヤー＞表示＞全画面表示＞オン」，FLATでは「全画面表示オン」

カテゴリ3 1セット内複数回繰り返される簡単な質問．繰り返し良く用いるコマンドに相当．
- 例：「音量を上げてください」

5.2 実験設定

実験はタブレットPCを用いて行った．TREE，FLAT，どちらの場合でも，試験を開始する前にシステムについて一通りの説明をし，被験者にも簡単にシステムを操作してもらってから実験を開始した．入力については，どちらの場合でも，タッチペンによる項目の選択と，音声入力の両方を許可したが，音声入力を積極的に利用してもらうことを促した．

実験は研究室の学生12名を被験者として，6人ずつ2グループに分けて行った．以降それぞれグループ1，グループ2とする．グループ1は，前半にフラット構造について，メディア再生用アプリケーションを操作するタスクを2セット行い，後半に木構造について，音声チャット用を操作するタスクを2セット行う．グループ2は，タスクの順番は同じに，フラット構造と木構造の順番を逆にして同様の手順で実験を行った．

5.3 実験結果

タスク達成率と設問あたりの完了時間を条件毎に比較したものを図3に，また，対象としたアプリケーション毎の問題カテゴリ別タスク達成率を表1，2に示す．これらの結果から以下のことが言える：

- タスク達成率の観点から，提案手法の優位性が認められる
- 提案手法では，初めてアプリケーションを利用する人でも，従来手法である程度なれたユーザと同等の操作が可能 (TREEの1回目とFLATの2回目の比較)
- “なんと言ってよいかわからない” 状態のユーザでも，提案手法では目的の機能への到達が比較的可能 (表1，2のカテゴリ2の設問において，FLATが27.12%，23.33%に対して，TREEが62.71%，53.33%)
- 頻繁に利用するコマンドの場合，連続キーワード入力による操作は，従来手法と同等の速度でコマンド発行が可能 (表1，2のカテゴリ3のtimeより)

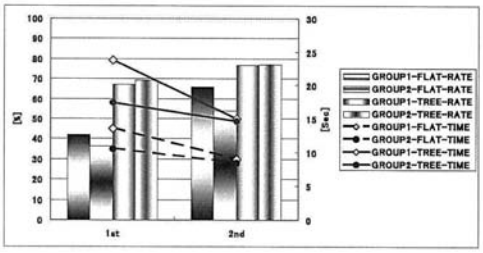

図 3: achievement rate and completion time

表 1: achievement rate by category[%] with completion time on category 3(Media Player)

Category	1st		2nd	
	FLAT	TREE	FLAT	TREE
1	45.00	66.67	53.33	76.67
2	27.12	62.71	68.33	73.33
3	70.63	91.67	91.67	87.50
(time[sec])	(9.9)	(10.2)	(5.8)	(5.9)
Average	41.96	69.23	65.97	77.08

表 2: achievement rate by category[%] with completion time on category 3(Voice Chat)

Category	1st		2nd	
	FLAT	TREE	FLAT	TREE
1	49.15	71.19	58.62	79.66
2	23.33	53.33	55.93	66.10
3	37.50	91.67	37.50	95.83
(time[sec])	(6.9)	(14.9)	(5.1)	(7.7)
Average	36.36	67.13	53.90	76.76

5.4 主観評価分析

アンケートの結果を図4に示す．ほとんどのカテゴリでFLATよりTREEの評価の方が高かった．特にLikabilityは，FLATとTREEで評価の差が大きかった．また，実験の完了時間を単純比較すると，実際にはFLATの方がTREEよりも早く完了しているのだが，Speedの評価からはその有意差が見られなかった．結果として，複数のキーワードを使い，階層を追うことで目的のコマンドに辿り着け

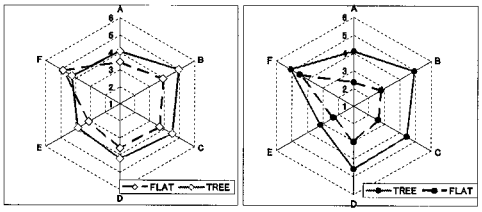

図 4: SASSI subjective assessment; A: System Responce Accuracy, B: Likability, C: Cognitive Demand, D: Annoyance, E: Habitability, F: Speed

る可能性が高い機能構造 (木構造) の方が，多少時間がかかっても主観的にはそのように感じず，好感度の高いインタフェースとみなされたことが分かる．

6 議論

今回の実験では，提案手法のタスク達成率が総合的な結果では 70%前後と低い値となっている．これは，タスク達成率を First Trial の結果のみを用いて結果を算出したからであり，従来のようにやり直しを認め，制限時間内で達成できるか否かを表す手法にした場合は，さらに高い達成率が期待される．実際のアプリケーションでは，試しに音声インタフェースを用いたユーザに取って重要なのが，自分の意図した操作を簡単に実行できるかであり，その印象としての First Trial の重要性から，今回はこのような実験設定を行った．この観点からすると，コマンド名がわからないときの実験結果 (設問カテゴリ 2) から，従来のインタフェースと比較すると First Trial の際のタスク達成が大幅に改善されており，提案手法では，これまで音声インタフェースの利用をあきらめていたユーザを取り込めることが期待できる．

また，今回はメニューベースの IVR との比較実験は行わなかった．これは，提案手法は従来のメニューベースを包含し，達成率の改善が明らかだったからである．Out-of-turn Interaction という，提示されたメニューよりも後のメニューで有効な入力を常時受け付ける手法でユーザビリティを改善した報告もある [8]．しかしながらが，この報告では比較対象が従来の IVR だけであり，Command and Control と同等の操作速度の実現は困難である．それに対して提案手法では，“連続キーワード入力” の枠組みによりこれを可能にする．

さらに，SOL Style のインタフェースとして，自然言語のサブセットを定義し，そのテンプレートを被験者に学習させて利用する手法 [4] もある．しかしながら，[4] では大域的なナビゲーションに関する手法は未だ示されておらず，提案されている主張だけで複雑なアプリケーションから意図した機能を選択可能かは疑問が残る．ただ，本手法も機能選択後の操作に関する言及はしておらず，このような手法によって補完可能なもの考えている．

最後に，本手法では認識結果のあいまい性と，選択肢のあいまい性を統合してユーザに解決を依頼する枠組みを提供し，音声認識システムの誤認識問題の解決手法とした．音声認識システムのあいまい性を解決する手法として，音声入力の段階で N-Best を表示し，ユーザに選択を促す手法 [7] はこれまでにも用いられて来たが，提案手法のように，アプリケーションのユーザインタフェースのモデルと統合して問題解決を一段階で済ませるものではない．

7 むすび

複数の機能の選択肢から，特定の機能の容易かつ効率的な選択を可能にする音声インタフェースとして，“機能構造” と” 連続キーワード入力” を用いたインタフェースを提案し，その定式化を行った．音声コマンドがフラットに用意された従来のインタフェースとの比較実験を行い，タスク達成率と主観評価の結果で，提案手法の優位性を示した．

謝辞 本研究は，経済産業省，平成 18,19 年度戦略的技術開発委託費「音声認識基盤技術の開発」の一部として実施されたものである．

参考文献

[1] 古井 貞熙, 他, “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, 2006.

[2] Nicole Yankelovich, et al., “Designing SpeechActs: Issues in Speech User Interfaces,” Proc. of CHI95, pp.369–376, 1995.

[3] Lisa J. Stifelman, et al., “VoiceNotes: A Speech Interface for a Hand-Held Voice Notetaker,” Proc. of CHI93, pp.179–186, 1993.

[4] Stefanie Tomko, et al., “Towards efficient human machine speech communication: The speech graffiti project,” ACM Tran. on Speech and Language Processing (TSLP), v.2 n.1, p.2-es, February 2005.

[5] Susan J. Boyce, “Natural spoken dialogue systems for telephony applications,” Communications of the ACM, v.43 n.9, p.29-34, 2000.

[6] Candace L. Sidner, Clifton Forlines, “Subset Languages for Conversing with Collaborative Interface Agents,” Proc. of ICSLP2002, pp.281–284, 2002.

[7] Jennifer Mankoff, et al., “Interaction techniques for ambiguity resolution in recognition-based interfaces,” Proc. of UIST2000, 2000.

[8] Saverio Perugini, et al., “A Study of Out-of-turn Interaction in Menu-based, IVR, Voicemail Systems,” Proc. of CHI2007, 2007.

[9] Min Yin, Shumin Zhai, “The benefits of augmenting telephone voice menu navigation with visual browsing and search,” Proc. of CHI2006, 2006.

[10] Lamere, et al., “Design of the CMU Sphinx-4 decorder,” Proc. of EUROSPEECH-2003, pp.1181–1184, 2003.

[11] K.S. Hone, R. Graham, “Towards a tool for the subjective assessment of speech system interfaces(SASSI),” Natural Language Engineering, vol. 6, Issue 3–4, pp.287–303, 2000.