2003－SLP－47　(8)
2003／7／19

CTI 向け自由発話対応 音声対話システム RexDialog

平沢 純一 †　山本 俊一郎 †　堀 貴明 ‡　大附 克年 †

日本電信電話株式会社

†NTT サイバースペース研究所　‡NTT コミュニケーション科学基礎研究所

〒 239-0847 神奈川県横須賀市光の丘 1-1　〒 619-0237 京都府相楽郡精華町光台 2-4

E-mail: hirasawa.j@lab.ntt.co.jp

あらまし

音声ポータルが実用化されるなど，CTI（Computer Telephony Integration）向けに音声認識を利用したサービスが始まっている．現状のサービス以上に利便性を高めるため，話題展開の自由，発話表現の自由，発話タイミングの自由，を実現した” 自由発話” 対応の音声対話システム RexDialog を開発した．自由発話に対応するための技術の中から，(1) クラス言語モデルの利用と作成環境，(2) 音声認識エンジンの連続駆動制御，(3) 対話制御規則の記述におけるスロット値の変化タイプ属性の導入，を紹介する．

RexDialog :
CTI-based Spoken Dialog System for Spontaneous Utterances

HIRASAWA Jun-ichi†, YAMAMOTO Shun-ichiro†,
HORI Takaaki‡, and OHTSUKI Katsutoshi†

†NTT Cyber Space Laboratories
1-1 Hikari-no-oka, Yokosuka, Kanagawa, 239-0847 Japan

‡NTT Communication Science Laboratories
2-4 Hikari-dai, Seika-cho, Souraku-gun, Kyoto, 619-0237 Japan

NTT Corporation

E-mail: hirasawa.j@lab.ntt.co.jp

Abstract

Today, some voice-portal services have started in Japan. However, callers do not always use those services comfortably, because the current voice-portal services have some constraints for callers' utterances. In order to avoid these constraints and allow the callers' spontaneous utterances, we have been developed a CTI-based spoken dialog system; RexDialog. In this paper, we focus on (i)adopting class N-gram language models with its generating tool, (ii)method of determining the end of user's utterances, and (iii)introducing the new feature for representing the changes of domain-slot values.

1 はじめに

米国で 2000 年に音声ポータルサービスが始まったのに続いて，2001 年には日本でも 7 月に日本テレコム社（voizi）が，8 月には NTT コミュニケーションズ社（V ポータル）が音声ポータルサービスの試験提供を開始した．V ポータル [10] は翌 2002 年 1 月に商用サービスに移行し，CTI（Computer Telephony Integration）向けに音声認識を利用したサービスが実用化され始めてきている．

現状の音声ポータルサービスは，技術的には，離散単語認識のための単語リストか簡単な定型文を記述した文法によって音声認識エンジンを制御し，VoiceXML[13, 14] のコードを記述することで対話のシナリオ（コールフロー）を制御する枠組で提供されている．これにより，サービス提供者（開発者）は電話回線や呼を制御するプログラミングから解放され，あたかも HTML ドキュメントを作成するだけで web ページを公開するように，音声サービスの提供が可能となっている．

しかしながら，現状の技術によるサービスは，人間同士の日常の対話の形式には必ずしも合致しておらず,「特別な利用法の習得が不要」という音声対話インタフェースの利点が活かせていない．サービスの利用法を熟知しているユーザか，ガイダンスを通じたシステムからの指示にまごつかずに対応できるユーザだけしか使えないのでは，音声ポータルサービスは普及が妨げられてしまう．

NTT サイバースペース研究所は，これに対して，利用者が話題を自由に展開でき，発話できる表現への制約が少なく，自由なタイミングで発話できることを目的として，自由発話対応の音声対話システム RexDialog を開発した．本稿では，CTI 向けの自由発話対応 音声対話システム RexDialog の概要を紹介（3 章）した上で，4 章で自由な発話に対応するために導入された RexDialog システムの特徴, すなわち, (1) 多彩な表現を認識するためのクラス言語モデルの導入と，言語モデルを簡易に作成するための環境, (2) 考えながらや言い淀みながらの利用者の発話にも話者交代を混乱させないための音声認識エンジンの連続駆動制御の方法, (3) 柔軟な対話の展開を許すために複雑かつ膨大になりがちな対話制御規則を，効率よく作成することに貢献する，ドメインスロット値の変化タイプ属性の導入，などについて述べる．

2 現状の課題

現状の技術レベルの範囲で音声ポータルサービスを提供する場合，例えば，図 1のような対話が行われる．

残念ながら，図 1のような対話は利用者（ユーザ）が日常行う対話の形式とは合致しておらず，必ずしもユーザに使いやすいとは言えない．具体的な問題を以下に示す．

- ユーザはシステムが予め定めた対話進行の順序通りに沿ってしか用件を遂行できない．離散単語認識のシステムでは一度に複数の項目をまとめて伝えられない．
- ユーザはシステムからの指示（プロンプト/ガイダンス）に従う内容しか発話できず，システムが想定している以外の表現は，認識されないか，誤認識されてしまう．
- 発声してよい（認識が動いている）箇所はシステムにより予め定められており，それ以外のタイミングでは，たとえ涙ながらに発話しても黙殺される

現状技術の課題は，ユーザ発話に対して 3 つの自由，すなわち, (a) 話題展開の自由, (b) 発話表現の自由, (c) 発話タイミングの自由，を提供できるようにすることと言える [4]．達成すべき目標は以下の通りとなる．

(a) 話題展開の自由： ユーザ発話が，システムガイダンスで指定される入力項目に従っていなくても，対話が破綻することなく用件が遂行される．

(b) 発話表現の自由： ユーザが発話中に用件（キーワード）以外のことを話しても，適切に用件を理解できる．

(c) 発話タイミングの自由： ユーザは対話中，どんなタイミングでも発話を開始することができる．また，話者交代（turn-taking）が乱れても，混乱することなく，柔軟に対話を進行できる．

3 自由発話対応 音声対話システム RexDialog

NTT サイバースペース研究所では，これらの自由な発話に対応できる CTI 向け自由発話対応 音声対話システム **RexDialog** を開発した．RexDialog では，確率言語モデル（クラス N-gram）による音声認識を採用し，入力場面ごとに認識文法を切り替

SYS	音声ポータルサービスです。お聞きになりたいコンテンツ名をおっしゃってください [beep] **発話タイミングの制限**
USR	天気案内 **コマンド名(コンテンツ名)の事前把握が必要**
SYS	○月○日、天気案内です。お知りになりたい都道府県名をおっしゃってください **システムにより入力順が指定される**
USR	神奈川県 **1発話＝1用件(キーワード)の制限**
SYS	神奈川県、のお知りになりたい地域を、東部、西部、全域、の中から選択しておっしゃってください
USR	東部 **1発話＝1用件(キーワード)の制限**
SYS	お知りになりたい日を、今日、明日、のようにおっしゃって下さい **システムにより入力順が指定される**
:	:

図 1: **現状技術による対話例．**

SYS	店舗検索システムです。お探ししたいエリアとジャンルをおっしゃってください **・複数スロットの同時待ち受けが可能**
USR	えーと、それじゃぁ、**横浜**の**映画館**だとどうですか？ **・複数入力項目の同時発声が可能** **・周辺表現や不要後に対応**
SYS	エリア横浜、ジャンル映画〔館ですね？〕←出力の途中停止 **・任意箇所で割込み可能** **(フルタイムバージイン)**
USR	いや、やっぱり**関内**でお願いします **・入力済の値の修正.変更が可能**
SYS	では、**横浜ではなく関内**、ジャンルは**映画館**でよろしいですね？ **・スロット値の変化を把握**
USR	はい、お願いします
SYS	それですと、6件になります…
:	:

図 2: RexDialog **での対話例．**

えない．その結果，ユーザ発話がガイダンスから逸脱しても認識でき，発話に用件（キーワード）以外の周辺表現や不要語があっても適切に理解できる．さらに，システムからの音声出力中も原則として常に認識を行う（フルタイムバージイン）ことで自由な発話タイミングを実現している．これらにより，RexDialog では図 1のような対話例とは異なり，図 2のような対話を実現することができる．

以下では，まず 3-1 節で音声対話システムの典型的な基本アーキテクチャについて述べ，3-2 節で自由発話に対応する RexDialog システムで基本アーキテクチャがどのように実現されているか述べる．

3.1 音声対話システムの基本アーキテクチャ

音声対話システムのアーキテクチャは，複数の要素技術を適切に機能分化させるモジュラリティを実現することが重要である．以下に，典型的な音声対

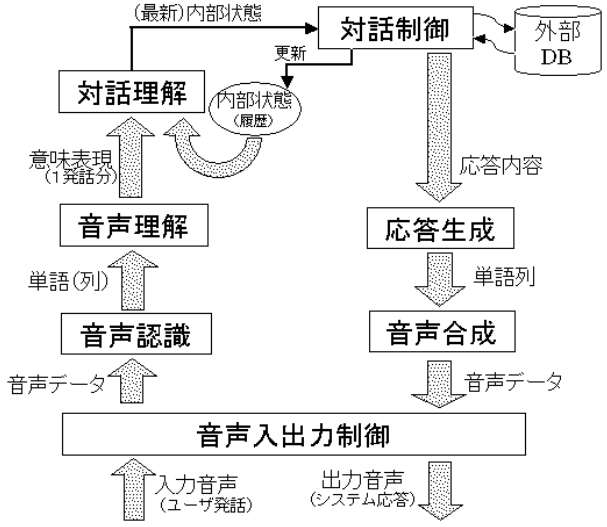

図 3: **音声対話システムの基本構成．**

話システムのモジュール構成を述べる（図 3）[1, 6].

音声認識部： Speech Recognition（SR）．入力は音声信号，出力は単語（列）．ユーザ発話の音声入力信号を単語（列）に変換する．

音声理解部 [1] **：**Speech Understanding（SU）．入力は単語列，出力は意味表現[2]．出力の意味表現とは，対話の履歴（文脈）から独立した 1 発話分の認識結果に対する，変数と値の組のリストによる表現である．

対話理解部 [3] **：**Dialog Understanding（DU）．処理すべきユーザ発話が入力される直前までの，対話システムが内部に保持している状態（文脈）と当該のユーザ発話 1 発話分の音声理解結果（意味表現）を掛け合わせて，最新の状態を得る処理．入力は音声理解部の理解結果（意味表現）とシステムの内部状態，出力は最新のシステム内部状態である [3].

対話制御部 [4] **：**Dialog Control（DC）．最新のシステム状態（対話理解部の出力結果）を用いて，システムが次に行うアクションを決定する処理．システムの次のアクションとは，ユーザへの応答の内容を決める，システムの内部状態を更新する，などである．

応答生成部　：対話制御部で決定された，出力すべきシステム応答内容を，具体的に発話すべき単語列の表現に変換生成する処理．

[1] 意味解析，言語理解，文理解とも呼ばれる．
[2] 対話行為表現（ユーザ発話の意図）とも呼ばれる．
[3] 文脈理解，談話理解，談話解析とも呼ばれる．
[4] 対話管理とも呼ばれる．

音声合成部　：応答生成部で生成された単語列から，実際に出力される音声を合成する処理.

音声対話システムは上記の構成を基本とするが，例えば，音声認識部が 1 単語しか出力しないのであれば単語列を意味表現に変換する音声理解（SU）部は実質不要となる．あるいは，対話の履歴（文脈）を活用せず，ユーザの 1 発話を DB への検索式（クエリー）に変換するだけの一問一答システムであれば，対話理解（DU）部は重要でなくなる[5].

3.2 RexDialog システムにおける構成

音声入出力： CTI 向けシステムである RexDialog は，音声の入出力制御，電話系の回線/呼制御に市販の CTI プラットフォーム[6] を用いている.

音声認識： 入力音声は NTT サイバースペース研究所の VoiceRex[7, 8] で単語列に変換される．VoiceRex は確率言語モデルを用いた音声認識を行える [8] ので，RexDialog では，タスクドメイン遂行に必要となる単語（キーワード）をクラスとしてまとめ，クラス N-gram により認識を行う．クラス N-gram の作成も含めた利用法は，4-1 節で詳しく述べる.

音声理解： 音声認識結果の単語列から，音声認識でのクラス名の情報などを用いて必要なキーワードを抽出し，適切なスロットに埋めて対話理解部へと出力する．また，RexDialog では話者交代を円滑にするため，音声認識エンジンの駆動/停止を音声理解部が連続的に制御している（詳細は 4-2 節で述べる）.

対話理解： 最新のシステム状態を算出する．その際，ドメインスロット（出発地/到着地/日にち etc.）に「値が埋まっているかどうか」だけでなく，スロットの値に「どんな変化が生じたか」を把握するために，内部状態のひとつとして「スロット値の変化タイプ属性」を導入し，対話制御規則の記述を容易にしている（詳細は 4-3 節）.

対話制御/応答生成： 対話進行のシナリオを定める対話制御規則は XML 形式で記述され，VoiceXML による記述と同様にプログラム言語での記述が不要である．対話制御規則は，if-then 規則の集合であり，条件節では内部状態の属性（変数）と値を参照し，条件に合致するとシステムの応答内容[7] と次の認識待ち受けのリソースが指定される.

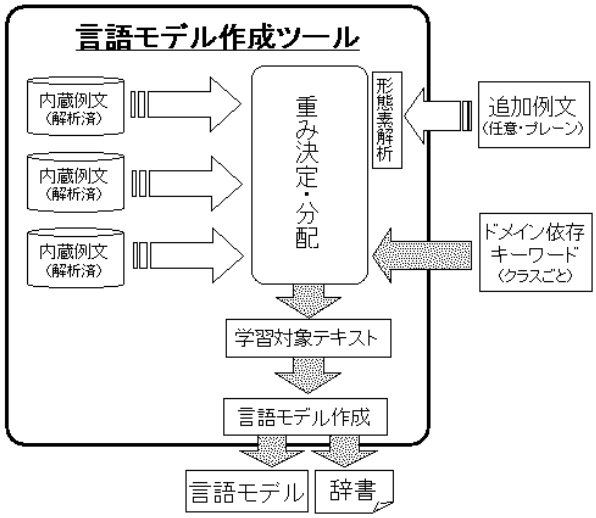

図 4: **クラス言語モデルの作成環境.**

対話制御/外部 DB の参照： ユーザから入力された値を引数として，外部 DB（cgi）を呼び出すことができる．これにより，検索条件に合致する件数や具体的な店舗情報を提供したり，DB 内の情報に依存した対話制御も記述できる [2, 5].

音声合成： 規則音声合成方式と録音編集方式（音声ファイルの再生）を利用することができる.

4 RexDialog の特徴

4.1 クラス言語モデルと作成環境

自由な発話表現（話し言葉）を認識するのに，人手による文法記述では対応しきれないばかりか，音声ポータルではコンテンツ提供者（開発者）は文法記述の専門家でないこともありえるため，RexDialog では発話例文テキストから N-gram 言語モデルを作成して音声認識を行う．また，対話システムなので完全なディクテーションは不要であり，ドメイン依存のキーワードでは更新作業が行われる（駅名を追加する etc.）ことから，単語 N-gram ではなく，クラス N-gram を用いている.

[5] まだ入力されていない項目を順に辿っていく FIA（Form Interpretation Algorithm）に基づいて挙動する VoiceXML 処理系は，対話理解（DU）のための明示的な仕様を持たない．但し，ユーザ（開発者）定義変数や（ECMA）スクリプトを駆使すれば，同等の機能を実現することは不可能ではない.

[6] NTT アイティ社 音声コミュニケーション事業部 CTI プラットフォーム Advice[9].

[7] 応答内容は直接，文字（単語）列で指定されるため，独立した応答生成部は存在しない.

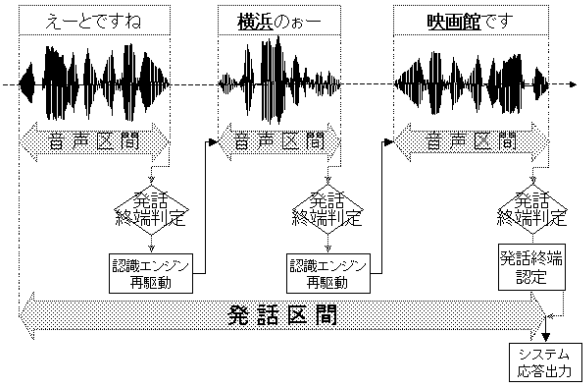

図 5: **音声認識エンジンの連続駆動制御．**

さらに，一般のコンテンツ提供者は，コストの面からタスクドメイン依存の例文テキストを調達することすら困難な場合がある．従って，各ドメインごとの言語モデル作成に当たっては新たに用意できる学習テキストは任意（少量もしくは無し）と仮定しなければならない．そこで，RexDialog システムでは言語モデル作成のための GUI ツールを提供した上で，ツール内部に予め大量の例文（形態素解析/クラス付与済テキスト）を内蔵しておく方式を採用した（図 4）．言語モデル作成ツールは，コンテンツ提供者が用意するドメイン依存のキーワード（及びクラス名）と任意で与えられる追加例文（プレーンテキスト）を元に，ツール内蔵の大量の例文からテキスト選択を行い，対象ドメインに合った重み付けで学習テキストを得て，言語モデルを作成する [11]．

4.2 音声認識エンジンの連続駆動制御

コマンド発声の場合にはユーザ発話の区切り（終端）は一定時間の無音で明確に検出できる．ところがユーザに自由発話を許す場合，ユーザは考えながら発話したり言い淀むので，発話中に一定時間の無音を挟んで発話を続けることがある．つまり，音声区間の終端がそのままユーザ発話の終端（システム応答の開始タイミング）とは限らない．ここで，無音区間を含むかもしれないユーザ発話の終端を確実に検出させたいために，音声の終端を検出する無音時間の閾値を長く設定してしまうと，通常のシステム応答のレスポンスが遅くなってしまう．従って，音声の終端検出の閾値を長く（システム応答のレスポンスを遅く）することなく，ユーザ発話中の短い無音でユーザ発話の終端としない（システムが応答を開始しない）ことが必要となる．

表 1: **スロット値の変化タイプの一覧．**

変化タイプ名	直前のスロット	音声理解結果	値の一致	意味
null to A	−	○	−	空(null)だったスロットに、(現発話による入力で)スロットに値が埋まった
null to notA	−	○(否定)	−	空(null)だったスロットに、値が埋まったが、その値は否定されていた
A to A	○	○	一致	既に値が埋まっていたスロットに、同じ値が再度、入力された
A to B	○	○	不一致	既に値が埋まっていたスロットに、異なる値が新たに入力された
A to notA	○	○(否定)	一致	既に埋まっていたスロットの値が、否定された
A to notB	○	○(否定)	不一致	既にスロットに値が埋まっていた状態に、否定されている異なる値が入力された
notA to A	○(否定)	○	一致	否定された値が埋まっていた状態に、同じ値が否定されずに入力された
notA to B	○(否定)	○	不一致	否定された値が埋まっていた状態に、否定されていない異なる値が入力された
notA to notA	○(否定)	○(否定)	一致	否定された値が埋まっていた状態に、同じ値が否定された状態で再度入力された
notA to notB	○(否定)	○(否定)	不一致	否定された値が埋まっていた状態に、否定されている異なる値が入力された
no change	−	−	−	スロットが空(null)だったが、現発話での入力にも値がなく、変化が無かった
no change	○	−	−	既にスロットに値が埋まっていたが、現発話の入力には値が無く、変化は無かった

そこで RexDialog では，ユーザ発話中に無音があればまず音声区間の終端として検出し，一旦，認識結果を出力する．しかし，この音声区間の終端をそのままユーザ発話の終端として扱わず，音声理解部がその認識結果を吟味してからユーザ発話の終端としてよいか判定する．例えば，最初に音声区間の終端を検出しても認識結果に内容語（キーワード）が含まれずフィラーだけの場合は，まだ発話が続くと判定して，もう一度認識エンジンの駆動を再開させる．その後の音声区間の終端で認識結果に 1 発話として十分な内容が含まれていれば，ユーザ発話の終端と判定し，システム応答を開始する（図 5）．これにより物理的な音声区間の終端とは別に，意味的な発話区間の終端を判定することができ，システム応答を遅らせることなく，ユーザ発話中の短い無音に対して応答を開始してしまう危険を減らせる[8]．現在，発話終端の判定規則は人手でチューニングしているが，対話データから学習することも可能である [12]．

4.3 スロット値の変化タイプ属性の導入

ユーザが自由に話題転換を行える場合，既に入力済のスロット値が変更されたり，修正されることがある．従って，対話制御規則は単に「スロットが埋まっているかどうか」の情報だけで対話を進めるの

[8] 人間同士の対話でも発話の衝突が避けられないように，連続駆動制御を導入しても話者交代の混乱が皆無になるとは限らない．

ではなく,「スロット値に変化があったかどうか」, 変化があった場合「どのような変化か（ex. 変更された，否定された etc.)」を把握して対話を進める必要がある．しかし，対話制御規則でこれらの状況を表現しようとすると，条件節の記述は繁雑になってしまう．

そこで RexDialog では，対話理解部で予め上記の判定を行っておき，当該スロット値の変化を「スロット値の変化タイプ」というひとつの属性で保持するようにした（変化タイプの一覧を表 1に示す)．これにより，ユーザ発話により生じた現在の状況が 1 つの属性の値だけで表現でき，対話制御規則の記述が簡素になった．スロット値の変化タイプ属性の導入は，柔軟な対話制御を行えるシナリオを見通しよく記述しやすくするだけでなく，スロット変化の希少事態（ex. 以前に言及されていない値がいきなり否定された [A to notB] etc.）を把握しやすくする効果もある．

5 おわりに

本稿で述べなかった話題： 電話サービスのシステムは現在の業界標準でもある VoiceXML 処理系（ボイスブラウザ）の上で提供される．ここで，N-gram による音声認識の認識リソースの指定を（VoiceXML ドキュメント中で）どのように記述するか，VoiceXML 処理系の組込み文法をどう扱うか，などは別の機会に改めて論じることにしたい．さらには，音声理解部で抽出したキーワードと VoiceXML シナリオ記述中の <field> タグとの関連の付け方や，対話理解（履歴）を考慮する柔軟な対話制御規則を VoiceXML 処理系にどのように組み込むか，などについても別の機会に述べることにする．

今後の課題： 自由発話に対応したコンテンツで，一定の認識精度を保ち，安定した品質でサービスを提供するためには，コンテンツ提供者が用意するドメイン依存の追加例文に関して，どのような質の例文をどれくらいの分量，用意すればよいのか，その質と量の関係を明らかにしていく必要がある．また，複雑になりがちな対話シナリオコンテンツを簡易に提供するための環境も今後の課題となろう．

まとめ： 定型発声や一問一答の対話進行による従来技術による音声ポータルサービス以上にユーザの利便性を高めるため，話題展開の自由/発話表現の自由/発声タイミングの自由を実現する，自由発話対応音声対話システム RexDialog を紹介した．特に，自由発話への対応として,（1）クラス言語モデルとその作成環境,（2）音声認識エンジンの連続駆動制御,（3）スロット値の変化タイプ属性の導入, などの特徴を紹介した．

謝辞 日頃よりご指導いただく NTT サイバースペース研究所メディア処理プロジェクト 小原永プロジェクトマネージャー，有益な示唆をいただく音声対話インタフェースグループの諸氏，松永昭一さん，に感謝致します．また，NTT 西日本 法人営業本部 吉岡理さん，関西学院大学 川端豪教授，NTT アイティ社 岡田政裕さん，井上歩さんはじめ，音声コミュニケーション事業部のみなさんにお世話になりました．

参考文献

[1] 荒木雅弘: 音声対話システムと VoiceXML. 人工知能学会研究会資料 SIG-SLUD-A103. pp.39–44,2003.

[2] 堂坂浩二, 安田宣仁, 相川清明: 試行型対話戦略による音声対話システムの確認発話削減. 言語処理学会 第 9 回年次大会発表論文集, pp.63–66, 2003.

[3] 東中竜一郎, 中野幹生, 相川清明: 複数文脈を用いる音声対話システムにおける統計モデルに基づく談話理解法. 情報処理学会研究報告, SLP-45-17, pp.101–106, 2003.

[4] 菊池英明, 工藤育男, 小林哲則, 白井克彦: 音声対話インタフェースにおける発話権管理による割込みへの対処. 電子情報通信学会論文誌, Vol. J77-D-II, No.8, pp.1502–1511, 1994.

[5] 駒谷和範, 上野晋一, 河原達也, 奥乃博: バス運行情報案内システムにおける適応的な対話管理を行うユーザモデルの評価. 言語処理学会 第 9 回年次大会発表論文集, pp.59–62, 2003.

[6] 中野幹生, 堂坂浩二: 音声対話システムの言語・対話処理. 人工知能学会誌, Vol.17, No.3, pp.271–278, 2002.

[7] 野田喜昭, 山口義和, 大附克年, 小川厚徳, 中川聡, 今村明弘: 音声認識エンジン VoiceRex の開発. 日本音響学会 秋季講演論文集 2-1-19, 1999-9.

[8] 野田喜昭, 山口義和, 大附克年, 今村明弘: マルチメディア時代を支える音声認識技術. NTT R&D, Vol.49, No.3, pp.142–148, 2000.

[9] NTT アイティ社 CTI プラットフォーム Advice. http://www.ntt-it.co.jp/goods/cts/index_cts.html

[10] NTT コミュニケーションズ社 V ポータル. http:// www.ntt.com/v-portal

[11] 大附克年, 堀貴明, 松永昭一, 川端豪: テキスト選択に基づくタスク依存言語モデル構築の検討. 日本音響学会 秋季講演論文集 1-5-20, 2000-9.

[12] Sato,R., Higashinaka,R, Tamoto,M., Nakano,M., and Aikawa,K. : Learning Decision Trees to Determine Turn-Taking by Spoken Dialogue Systems. *Proceedings of the 7th International Conference on Spoken Language Processing, (ICSLP-2002)*, pp.861–864, 2002.

[13] VoiceXMLForum: Voice extensible markup language VoiceXML. http://www.voicexml.org

[14] W3C Voice Browser Activity. http://www.w3.org/Voice