ヒューマンインタフェース　97－24
音声言語情報処理　40－24
（２００２．２．２）

移植性の高い音声対話システムにおける対話スクリプト構築ツールの試作

小暮 悟† 伊藤 敏彦†† 中川 聖一†

†豊橋技術科学大学 情報工学系
〒 441-8580 豊橋市天伯町字雲雀ヶ丘 1–1
{kogure, nakagawa}@slp.ics.tut.ac.jp

††静岡大学 情報学部 情報科学科
〒 432-8011 浜松市城北 3–5–1
t-itoh@cs.inf.shizuoka.ac.jp

あらまし 音声対話システムにおける音声認識技術や言語処理技術の要素技術に関しては確立しつつあり，実用化に向けシステム開発が進んでいる．実用化などを考慮した場合，今までのような使い易さ，頑健性などに関する技術だけでは不十分であり，拡張性や移植性なども十分考慮する必要がある．我々も移植性の高い音声対話システムについての検討を行ない，音声認識部については言語モデルのタスク適用について未知語の登録を考慮した手法を提案している．また言語理解部についても従来システムと比較して性能を落さずにシステム構築の効率を大幅に削減できたことをすでに報告している．今回，従来システムにおいて不完全であった対話管理部に注目し，データベース検索の分野において，対話を管理する情報をドメイン・タスク独立な情報とドメイン・タスク依存な情報に分割した．アプリケーション構築者はドメイン・タスク依存な情報である XML ベースの対話スクリプトを構築することによって様々な対話主導を実現可能である．さらに，対話スクリプトを編集する GUI ツールを試作したので報告する．

A Prototype Tool of Designing Dialogue Script for Highly Portable Spoken Dialogue Systems

Satoru KOGURE†, Toshihiko ITOH†† and Seiichi NAKAGAWA†

†Department of Information and Computer Sciences
Toyohashi University of Technology,
Tenpaku, Toyohashi, 441-8580, Japan
{kogure, nakagawa}@slp.ics.tut.ac.jp

††Department of Computer Science,
Faculty of Information, Shizuoka University
Johoku, Hamamatsu, 432-8011, Japan
t-itoh@cs.inf.shizuoka.ac.jp

Abstract Recently the technology for speech recognition and language processing for spoken dialogue systems has been improved, and speech recognition systems and dialogue systems have been developed to be practical use. In order to become more practical, not only those fundamental techniques but also the techniques of portability and expansibility should be developed. We already presented the portability of spoken dialogue systems. In our past research, we demonstrated the portability of the speech recognition module and the interpreter. In this paper, we focused on the portability of the dialogue manager that had no enough portability. In a database retrieval system, we distinguish domain/task dependent data sets with domain/task independent ones on a dialogue manager. An application developer can design several dialogue initiative strategies according to building dialogue script based on XML. We also report GUI tools that are used in developing the dialogue script.

1 はじめに

一般に，音声対話システムは，対話の対象となる領域・分野を限定して開発されるのが普通である．また，利用者が解決・達成したい問題・処理の種類によっても音声対話システムの仕様が異なることが多い．ここで，本稿では対話の対象となる領域・分野のことをドメイン，あるドメイン下で利用者が解決・達成したい問題・処理のことをタスクと呼ぶことにする．

通常，音声対話システムの仕様は，ドメインやタスクに大きく依存するが，音声対話システム全体のモジュール性を高め，システムの移植性や拡張性を上げるためには，システムのどの部分がドメインやタスクに依存していて，どの部分がドメインやタスクからは独立であるかを明確に区別する必要がある．しかし，現実世界には様々な対話形式が存在し，そのすべてに対処するようなシステムを構築することには困難が伴う．

一般的に音声対話システムの移植性とは，システムのドメイン・タスクを変更することの容易さを示し，拡張性とは，ドメイン・タスク知識をあとから動的に追加・修正・削除することの容易さを示す．本稿においてはデータベース検索用音声対話システムという広義のドメインに限定して議論を進める．本稿におけるデータベース検索用音声対話システムの移植性とは，観光案内や文献検索といった狭義のドメイン・タスクを変更することの容易さを示し，拡張性とは，ドメイン・タスク知識を後から動的に追加・修正・削除することの容易さを示す．

実際に新しい音声対話システムを最初から構築するには莫大なコストがかかることから，今後，これまで開発してきたシステムを他のドメイン用に変更したり，汎用的なシステムを開発することが重要になってくると予想される．実際,「移植性」や「拡張性」を重要視する研究も行なわれてきている．

PIA [1] というシステムは，複雑な音声対話システムでもプロトタイプを簡単に構築できる．Visual BASIC を用いて実装され，認識のロバスト性と対話の自然さを両立させることに重点をおいている．REWARD(Real World Applications of Robust Dialogue) [2] というプロジェクトで試作しているシステムは，開発者がシステムの開発，デバッグを一括して管理でき，従来の音声対話構築よりも早い時間でシステムを構築することが出来る．OGI (Oregon Graduate Institute) で研究が勧められている CSLU Toolket [3, 4] というシステムは，音声に関する知識を全然持っていなくても音声を使ったアプリケーションを素早く構築することが出来る．DARPA Communicator プロジェクトはマルチモーダルな音声対話を簡単に構築できる環境を提供する目的で 1998 年から始まっており，その基本概念は，MIT の Galaxy-II Hub [5] の枠組を利用して対話システムの各構成要素が情報を送受信するものである．その例として，CU Comunicator system [6] などがあり，このシステムでは実際にカーナビゲーションのプロトタイプを構築している．笹島ら [7] は EUROPA という音声対話構築ルールを提案し，MINOS というカーナビゲーションシステムを実際に構築して評価している．秋葉ら [8] は，マルチモーダルインターフェースの汎用性に関する研究を報告している．彼らは，MILES という マルチモーダル対話記述言語を開発し，ジャンケンや地下鉄乗り換え案内などいくつかの対話タスクを実際に試作している． 荒木ら [9] は，音声対話システムにおけるタスクを「スロットフィリング」「データベース検索」「説明」「それ以外」に分割している．前 3 つのタスクについては，その対話をスクリプト言語で記述できる．

一方，我々も「音声対話システムにおける移植性」に関する研究を行なってきた．既存のシステムである富士山観光案内 [10] を東三河観光案内に実際に適用し，このような似ているドメインへの変更でも 30 日・人かかり，全く違うドメインへの変更にはさらに大幅な作業時間がかかることを示した [11]．本研究では，音声対話システムの構成要素としては音声認識，言語理解，応答生成と対話管理が考えられるが，音声認識においては，少量の対話コーパスからよりロバストな言語モデルを構築するための手法について検討を行なった [12]．さらに，言語理解や応答生成の部分の移植性に関しても検討を行ない [13] ，さらにドメイン・タスク依存部とドメイン・タスク独立部の明確な分割，検索結果表示部，各種データの GUI ツールによる修正などの改良を加えた [14]．

しかしながらこれまでに開発したシステムの対話管理は非常に単純であり，対話主導を変更するのに対話管理をしているシステムコア部の修正が必要であった．今回，この対話管理部に注目し，移植性を考慮して対話管理に必要となる情報をタスク独立な物とタスク依存な情報に分類した．これによりアプリケーション構築者はタスク依存な情報である XML 形式の対話スクリプトを編集するだけで様々な対話主導を実現できる．さらに，対話スクリプトをより簡単に編集するための GUI ツールを試作した．

2 対話管理を考慮した移植性の高い音声対話システム

まず，様々なデータベース検索に関する対話文集合 [15–17] を調べた．データベース検索を対象とした場合，対話は以下に示す 7 つの状態の遷移でモデ

ル化できる．

- 初期状態：対話を開始する際の処理をするための状態．
- 検索条件入力：ユーザが検索したい情報を入力するための状態．
- 検索実行：ユーザの入力に従ってデータベースを検索するための状態
- 検索結果表示：検索された情報をユーザに提示するための状態
- 確認：さまざまな状況においてユーザに確認を取るための状態．
- 説明：システムの使用方法に関する情報を説明するための状態．
- 最終状態：対話を終了する際の処理をするための状態．

さらに，対話の流れを決定する情報（今後，これを対話スクリプトと記す）では，「検索条件入力」，「検索実行」，「検索結果表示」，「確認」，「説明」の 5 つの状態をクラスと考え，実際の対話を，前述の 5 クラスからインスタンス化した任意個数の対話状態間の状態遷移であるとする．なお，「初期状態」クラスと「最終状態」クラスの 2 つは，インスタンス化せず，そのまま使用する．

これに沿って構築したシステムの動作図を図 1に示す．ここで，対話管理部内部の 5 つの円がそれぞれ対話状態インスタンスを示す．なお，図中の対話インスタンスは 5 つ（各対話状態クラスから 1 つずつインスタンス化）であるが，各対話状態クラスについて，必ず 1 つ以上使用しなければならないという制限はなく[1]，どの対話状態クラスについても任意個数インスタンス化可能で，どの対話状態インスタンスからからでもすべてのインスタンスへ遷移可能である．

ここで重要なのは，「初期状態」と「終了状態」以外の 5 つのクラスについてクラスの内部的な処理の遷移がタスク独立であると仮定できる点である．つまり，たとえば，検索条件入力クラスの内部でどのような処理が行われているかはアプリケーション構築者が触れる必要はない．よって，アプリケーション構築者は対話を構築する際には，ここに示す 5 つの対話状態クラスからインスタンス化された任意個数の対話状態間の遷移で対話を表現することができるということになる．

図 2 に各状態インスタンスにおける内部処理のアルゴリズムを示す．ここで，図中の (a),(b),(c),(d),(e) はそれぞれ，検索条件入力，検索実行，検索結果表示，確認，説明の 5 状態における内部処理アルゴリズムである．また，図 2 (a) にある SI モード とは，簡易的なシステム主導を実行するためのフラグで，図にある通り入力プロンプトを表示するかどうかを制御する．

[1] 極端な例を言ってしまえば，対話インスタンスなしで，初期状態と終了状態のみの対話を実現することも可能である

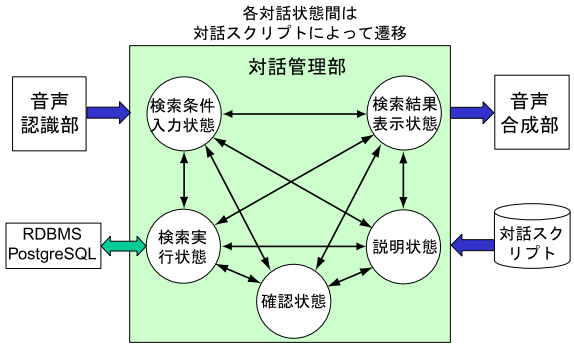

図 1: システムの構成

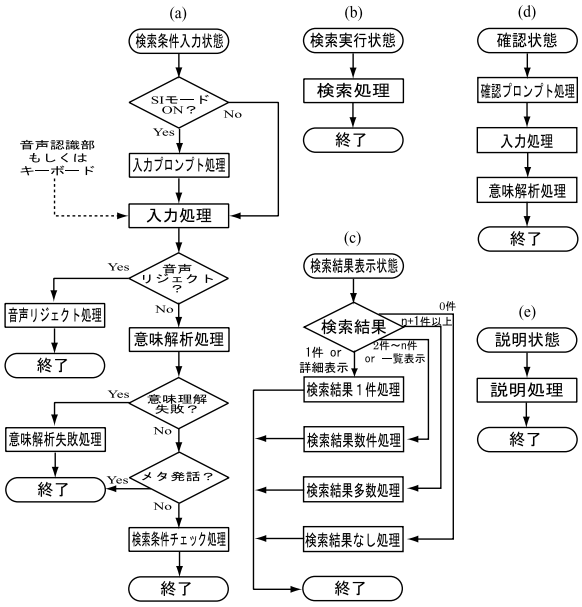

図 2: 各対話状態クラスの処理のアルゴリズム

2.1 対話スクリプト

上記までに示した対話状態を用いて対話を制御するために記述された情報を，対話スクリプトと呼ぶことにする．図 3に対話スクリプトの定義を示す．なお，以下に示す定義の中で，先頭が大文字のものがクラスを，先頭が小文字のものがアトリビュート（属性値）をそれぞれ示す．

```
Dialogue := title kind InitialState State*
            FinalState
InitialState := nextstate Script
Script := (Let|Run)*
Let := name value
Run := command argument
State := name kind InitialScript? ToState+
InitialScript := Script
ToState := nextstate condition Script
FinalState := Script
```

図 3: 対話スクリプトの文法定義

図 3における対話スクリプトの文法を示す．なお，図中における ?, +, *, は正規表現における 0 個か 1 個，1 個以上，0 個以上にそれぞれ対応する．まず Dialogue は対話スクリプト全体を含むクラスで，対話システムのタイトル title と対話システムタスク名 kind 属性，および初期状態クラス InitialState 一つ，対話状態クラス State 0 個以上，最終状態クラス FinalState 一つから構成されることがわかる．同様に InitialState は，初期状態のクラスを表し，遷移先状態 nextstate 属性と処理スクリプトクラス Script から構成される．Script は内部処理の実行の流れを記載する処理スクリプトのクラスで，変数代入クラス Let あるいは コマンド実行クラス Run の 0 個以上から構成される．Let は変数代入のためのクラスで，変数名 name と 変数値 value 属性から構成される．Run はコマンド実行のためのクラスで，コマンド名 command と引数列 argument 属性から構成される．State は対話状態クラスを表し，初期処理スクリプト InitialScript 0 個あるいは 1 個， 状態遷移情報 ToState 1 個以上から構成される．InitialScript は State の開始時に実行される処理スクリプトのクラスで Script と同定義 (変数代入クラス Let あるいはコマンド実行クラス Run の 0 個以上から構成) である．ToState は State 中の状態遷移情報のクラスで，状態遷移先 nextstate と状態遷移条件 condition 属性，および処理スクリプトクラス Script から構成されている．

次に実際の対話スクリプトのデータ形式であるが，システムは基本的には Lisp によって動作しているため，現時点では対話スクリプトのデータ形式は Lisp である．しかし，Lisp 形式のデータは，あまり一般的ではないし，階層構造の終了が ')' のみであり，どの階層のデータ記述が終了したかの情報を含まないという欠点がある．そこで，対話システム構築時には XML 形式によって対話スクリプトを構築し，そのデータを Lisp 形式に変換することで，XML 形式で記述された対話スクリプトをシステムに反映させることができるようにシステムを構築した．そのフォーマットに沿って作成した後述の富士山観光案内における対話スクリプトの一部を図 4に示す．

3 対話主導の構築：富士山観光案内タスク

3.1 対話スクリプトの構築

富士山観光案内システムへの実際の実装において，システム主導，ユーザ主導，混合主導の 3 つの対話主導を実現するための対話スクリプトをそれぞれ用意した．構築は XML の対話スクリプトを手作業で構築し，図 5 の GUI で視覚的にチェックを行なったあと，GUI を使って Lisp 形式の対話スクリプトに変換して行った．これをシステムで使用して，意図した対話の流れで実際にシステムが動作することを確認した．

構築した 3 つの対話主導と対応するタスクの説明を以下に示す．紙面の関係上すべての例を載せられないため，混合主導型についてのみ詳しい説明を載せることにする．

混合主導型

目標とするタスクは，最初はユーザ主導型で行ない，意味理解に連続で失敗した場合にシステム主導型に移行する (ユーザ主導型，システム主導型ともに前節で述べたタスクと同様) システムである．対話の流れと対話の遷移の流れを図 6に示す．対話例において,「富士山の高さはいくらですか」という質問文はシステム構築時には考慮されていないため意味理解に失敗する．2 回意味理解に失敗したところで「検索条件入力－１」の状態から「検索条件入力－場所」に遷移し，システムからユーザへ質問を行なう（ユーザ主導からシステム主導への遷移）．

3.2 対話スクリプトの記述能力の評価

システム主導型，ユーザ主導型，混合主導型の 3 種類の対話主導について，本研究で提案する対話スクリプトですべての対話主導を実現し，対話スクリプトの記述能力を確認した．対話スクリプトは XML で記述でき，編集や対話状態の確認は図 5 の GUI ツールを用いて変更可能である．

```
<?xml version="1.0" encoding="EUC-JP"?>
<dialogue title="富士山観光案内サンプル" kind="富士山観光案内">
<initialstate nextstate="検索条件入力－１">
<script>
<let name="必須検索条件" value="(場所 || 種類 || 施設名 || 行動)" />
<let name="挨拶応答メッセージ" value="こんにちは" />
   :
<let name="音声リジェクトメッセージ" value="入力された音声がリジェクトされました．もう一度発話して下さい" />
<let name="意味理解失敗メッセージ" value="意味理解に失敗しました．もう一度入力して下さい" />
<let name="ＳＩモード" value="FALSE" />
<run command="システム出力" argument="富士山観光案内システムです．ご用件をどうぞ" />
</script>
</initialstate>

<state name="検索条件入力－１" kind="検索条件入力">
<to nextstate="検索結果表示－１" condition="全検索結果表示？ || 詳細表示？ || 番号指定詳細表示？">
<script>
</script>
</to>
  :
  :

<state name="説明－１" kind="説明">
<to nextstate="検索条件入力－１" condition="T">
<script>
</script>
</to>
</state>

<finalstate>
<script>
<run command="システム出力" argument="御利用ありがとうございました" />
</script>
</finalstate>
</dialogue>
```

図 4: XML 形式の対話スクリプトの例

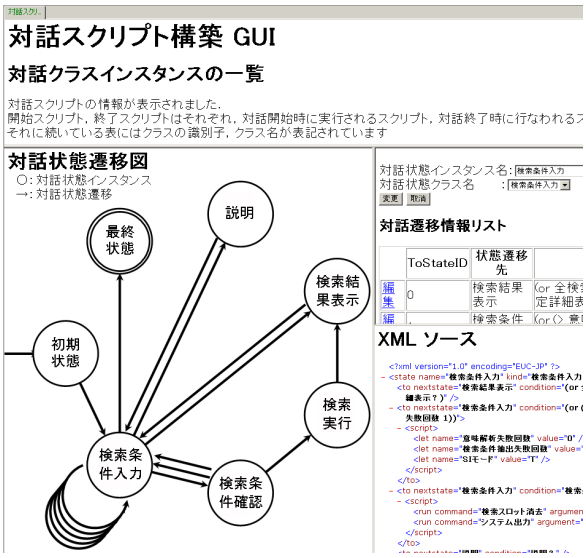

図 5: 対話スクリプトを構築するための GUI ツールの動作例

```
[初期状態]
システム：富士山観光案内システムです．ご用件をどうぞ
[検索条件入力－１]
   ユーザ：富士山の高さを教えて下さい
システム：意味理解に失敗しました．もう一度入力して下さい
システム：検索条件が満たされていません
[検索条件入力－１]
   ユーザ：富士山の高さはいくらですか
システム：検索条件が満たされていません
[検索条件入力－場所]
システム：場所を検索条件として入力して下さい
   ユーザ：富士山
[確認－場所]
システム：抽出された検索条件を以下に示します
システム：場所は以上でよろしいですか？
   ユーザ：はい
[検索条件入力－種類]
システム：種類を検索条件として入力して下さい
   ユーザ：遊園地
[確認－種類]
システム：抽出された検索条件を以下に示します
システム：施設の種類は以上でよろしいですか？
   ユーザ：はい
[検索実行－２]
システム：抽出された検索条件を以下に示します
[検索結果表示－２]
システム：１件見つかりました
```

図 6: 富士山観光案内の混合主導型の対話例

音声対話システム用の対話スクリプト記述手法としては VoiceXML [2] が有名である．これは，XML 規格の対話スクリプトで認識エンジンや認識に使用する文法を XML 中に記載できる．その意味では今回提案している対話スクリプトはその部分に関する記述能力は低い．しかし，VoiceXML は記述能力が高い反面，対話とは直接関係のない，音声認識部や意味理解部についての細かな指定を行なう必要がある．これは，対話システムに詳しくない初心者からみれば VoiceXML は非常に理解しにくい記述言語であると言い替えることも出来る．

3.3 対話スクリプトの構築効率の評価

以前の対話管理を考慮していなかった移植性の高い対話システム [14] において，対話の流れは Lisp の手続きで記載されていた．この対話処理の流れの変更は Lisp のプログラムを変更する必要があり対話管理の観点からは移植性が低かった．一方，今回提案する対話管理を考慮した移植性の高い対話システムにおいては，対話の流れについても移植性を高め，前述の 3 つの対話主導を XML の記述の変更のみで実現できた．この XML の記述の変更には

[2] http://www.voicexml.org/

一つの対話主導について 30 分ほどしかかかっていない．

4　対話主導の構築：文献検索

文献検索タスクについても同様にシステム主導型，ユーザ主導型，混合主導型の対話スクリプトをそれぞれ構築し，実際にシステムで対話スクリプトが意図した通りに動作することを確認した．対話の流れ，対話スクリプトは前述の富士山観光案内のものと大筋で同等であり，紙面の関係上からすべての実例の紹介は省略する．システム主導型における対話の例を図 7に示す．

[初期状態]
システム：文献検索システムです．ご用件をどうぞ
[検索条件入力－著者]
システム：著者を検索条件として入力して下さい
　ユーザ：著者は中川聖一さんです
[確認－著者]
システム：抽出された検索条件を以下に示します
システム：著者は以上でよろしいですか？
　ユーザ：はい
[検索条件入力－キーワード]
システム：キーワードを検索条件として入力して下さい
　ユーザ：キーワードは音声認識
[確認－キーワード]
システム：抽出された検索条件を以下に示します
システム：キーワードは以上でよろしいですか？
　ユーザ：はい
[検索実行－１]
システム：抽出された検索条件を以下に示します
[検索結果表示－１]
システム：３件見つかりました

図 7: 文献検索のシステム主導型の対話例

5　まとめ

移植性の高い対話管理部を構築するにあたり，まず様々な対話文集合を調査した．そして，データベース検索に関する対話システムにおいて必要となる処理をドメイン・タスクに依存する部分と独立である部分に分類した．さらに，依存する部分を XML 形式の対話スクリプトで記述できる手法を提案した．また XML 形式の対話スクリプトの構築および確認が可能な GUI ツールを試作し，構築の効率について述べた．

しかし，XML 記述の定義，および構築の主観的評価は著者の一人が行なっており構築効率の正当な評価とはいえない．そこで，対話システムに関してあまり詳しくない大学院生を対象に，GUI ツールを使った対話スクリプトの構築をしてもらい，構築効率の評価を行なう予定である．

参考文献

[1] S. Kaspar and A. Hoffmannn, “Semi-automated incremental prototyping of spoken dialog systems ” , Proc. ICSLP’98, Vol. 3, pp. 859–862, 1998.

[2] T. Brondsted, B. Bai and J. Olsen, “The REWARD Service Creation Environment. An Overview”, Proc. ICSLP’98, Vol. 4, pp. 1175–1178, 1998.

[3] S. Sutton, R. Cole, J. de Villiers, J. Schalwyk, P. Vermeulen, M. Macon, Y. Yan, E. Kaiser, B. Rundle, K. Shobaki, P. Hosom, A. Kain, J. Wouters, D. Massaro and M. Cohen, “Universal Speech Tools: The CSLU Toolkit”, Proc. ICSLP’98, Vol. 7 pp. 3221-3224, 1998.

[4] B. Serridge: “An Undergraduate Course on Speech Recognition Based on the CSLU Toolkit”, Proc. ICSLP’98, Vol. 4, pp. 1663–1666, 1998.

[5] S. Seneff, E. Hurley, R. Lau, C. Pao, P. Schmid and V. Zue, “GALAXY-II: A Referece Architecture for Conversational System Development”, Proc. ICSLP’98, pp. 931–934, 1998.

[6] B. Pellom, W. Ward, J. Hansen, R. Cole, K. Hacioglu, J. Zhang, X. Yu and S. Pradhan, “University of Colorado Dialogu Systems for Travel and Navigation”, Proc. Int. Conf. on Human Language Technology’2001, pp.362–367, 2001.

[7] M. Sasajima, T. Yano, and Y. Kono, “Europa: Generic Framework for Developing Spoken Dialogue System”, Proc. EUROSPEECH’99, pp.1163–1166, 1999.

[8] 秋葉 友良, 伊藤 克亘, “スクリプト言語を用いたマルチモーダル対話記述の試み”, 情報処理学会, 情処研報, SLP-23-1/HI-80-1, 1998.

[9] 荒木 雅弘, 駒谷 和範, 平田 大志, 堂下 修司, “音声対話システム構築のための対話ライブラリ”, 人工知能学会, SIG-SLUD-9901-1, 1999.

[10] S. Nakagawa, S. Kogure and T. Itoh, “A semantic interpreter and a cooperative response generator for a robust spoken dialogue system”, IJPRAI, Vol. 14, No. 5, pp. 553–569, 2000.

[11] 小暮　悟, 伊藤 敏彦, 中川 聖一, “音声対話システムの移植性に関する考察 –観光案内システムとデータベース検索システム–”, 情報処理学会, 情処研報, SLP-25-3, 1999.

[12] 小暮　悟, 堀　賢史, 中川 聖一, “音声対話システムのための未知語の登録を考慮した言語モデルの構築”, 情報処理学会, 情処研報, 99-SLP-31, pp. 39–44, 2000.

[13] S. Kogure and S. Nakagawa, “A portable development tool for spoken dialogue systems”, Proc. ICSLP’2000, vol. I, pp. 238–241, 2000.

[14] 小暮　悟, 中川 聖一, “データベース検索用音声対話システムの移植性の高い意味理解部・検索部の構築と評価”, 情報処理学会論文誌, Vol. 43, No. 3, 採録決定, 2002.

[15] 文部省重点領域研究 [音声対話],CD-ROM Vol.1, 1994.

[16] 文部省重点領域研究 [音声対話],CD-ROM Vol.2-4, 1995.

[17] 電総研道案内対話音声コーパス,CD-ROM Vol.1-7, 1998.