

拡張性を備えたオープンな電話対話システム開発ツール TEDDI

伊藤 和明[†] 山口 由紀子[‡] 河口 信夫[‡] 松原 茂樹[‡] 稲垣 康善[†]

[†]名古屋大学大学院工学研究科 〒464-8603 名古屋市千種区不老町
[‡]名古屋大学情報連携基盤センター 〒464-8601 名古屋市千種区不老町
E-mail: kazu_ito@inagaki.nuie.nagoya-u.ac.jp

あらまし 電話を介した対話システムの多目的利用を目指し、オープンで拡張性を備えた開発支援ツール TEDDI の構築を行った。本開発支援ツールは、電話を利用する対話システムに必要な、音声認識や音声合成、電話の制御など9つの基本コンポーネントを備えている。対話システム設計者は、これらのコンポーネントのインスタンスであるブロックを GUI 上で接続していくことにより、システムを開発することができる。また、システム設計者が必要に応じて機能を拡張するための機構として、スクリプト記述コンポーネントとクラス定義コンポーネントを備えている。このうち、クラス定義コンポーネントでは、オブジェクト指向における継承を利用することによって、効率のよい機能拡張が可能である。本ツールで開発した対話システムは、実際に電話を介して動作することができる。本稿では、対話システムの開発支援環境に望まれる項目について考察し、本開発支援ツール TEDDI の概要と実装について述べる。また、本開発支援ツールの実用例として、電話音声収集システムとネットワーク状況応答システムの開発について述べる。

キーワード 音声対話システム、開発支援、GUI、オブジェクト指向、電話音声収集システム、ネットワーク状況応答システム

TEDDI: Open and Extensible Tool for Designing Telephone-based Spoken Dialogue Systems

Kazuaki ITO[†] Yukiko YAMAGUCHI[‡] Nobuo KAWAGUCHI[‡]

Shigeki MATSUBARA[‡] and Yasuyoshi INAGAKI[†]

[†] Graduate School of Engineering, Nagoya University Furo-cho, Chikusa-ku, Nagoya, 466-8603 Japan

[‡] Information Technology Center, Nagoya University Furo-cho, Chikusa-ku, Nagoya, 466-8601 Japan

E-mail: kazu_ito@inagaki.nuie.nagoya-u.ac.jp

Abstract We have developed TEDDI, a design and development tool for telephone-based spoken dialogue systems. TEDDI consists of nine components, such as telephone control, speech synthesis, speech recognition, and two enhancing components. Using TEDDI, a dialogue system can be developed in a GUI environment, in fact, examined and executed. We have implemented TEDDI in an object-oriented language Ruby, and a designer can write an original function efficiently by using inheritance from a basic component. This paper describes TEDDI and its application to the development of a telephony voice collection system and a network information system.

Keyword Telephone-based Dialogue System, Design and Development Tool, GUI-based Design, Object-Oriented Design, Telephone Voice Collection System, Network Information System

1. はじめに

電話音声認識技術の向上や携帯電話の普及に伴い、電話を利用した音声対話システムが実用化されるようになった。製品メーカーによるユーザサポートのためのコールセンターや、各種情報を提供する音声ポータルサービスに利用されている。また、それに伴い、これらの CTI(Computer Telephony Integration)サービス構築のための商用環境が様々な企業から提供されて

いる。これらの商用環境は、CTI サービスの構築を目的としており、電話対話システムを用いて研究する場合には、導入費用や汎用性などが問題点となり、汎用的に利用可能でオープンな開発環境が提供されていることが望ましい。

そこで本研究では、電話を利用する対話システムの開発を支援するツール TEDDI を構築した。本開発支援ツールでは、電話対話システムの構築に必要な基本コ

コンポーネントをブロックとして表現し、設計者がそれらをGUIキャンパス上で接続することにより対話システムを設計できる。また、基本コンポーネントで提供されていない機能については、対話システム設計者がスクリプトを記述することにより追加できる。さらに、各コンポーネントをオブジェクト指向におけるクラスとして実現しているため、クラスの継承を利用することによって、新しい機能の追加や既存の機能の再定義など、効率的な機能拡張が可能であり、多様なシステム開発に利用可能である。本ツール上で開発した対話システムは、実際に電話を介した動作が可能である。

本稿では、電話対話システムの設計支援環境に望まれる項目について考察し、本開発支援ツール TEDDI について述べる。また、TEDDI を用いた対話システム開発の例として、電話音声収集システムとネットワーク状況応答システムの開発について述べる。

2. 対話システム開発支援環境

2.1. 電話対話システムの研究利用

電話を介した対話システムは、被験者実験のインタフェースとして各種研究に利用できる。例えば、音声対話システムや音声認識エンジンの評価実験、電話音声の収録、電話アンケート調査の自動化などが挙げられる。電話は多くの人が手軽に利用できるインタフェースであり、大量の被験者データを集めたい場合には有効な手段となる。また、電話アンケートによる世論調査など、対話システム研究以外にも利用できる。

しかし、電話を利用した対話システムを開発するためには、対話制御や音声認識、電話制御などに関する知識が必要であり、その開発には多大なコストを要する。一方、コールセンターなどで対話システムによる自動応答が導入され始め、それに伴って CTI サービス構築のための商用環境が提供されている。上記のような研究を目的とした対話システムの開発に、これらの商用環境を導入することは、費用や手続きの点で難しい。また、商用環境には、拡張性、開放性の欠如などの問題がある。音声認識エンジンだけを独自のものに取り替えたり、既存の対話システムを組み込んで利用したりすることを容易に実現できなければ、研究を目的とした対話システムの開発に適しているとはいえない。

2.2. 開発支援環境に望まれる要素

前述したような電話対話システム開発支援環境には、以下のような要素を備えていることが望まれる。

- **モジュール性**：電話対話システムの必要な各基本機能に関する専門知識が無くても対話システムの開発ができるように、電話を介した対話

システムの基本機能をすべて備え、機能ごとにコンポーネントを構成すること。基本機能とは、音声認識、音声合成、電話制御などである。

- **統合性**：対話システムの設計段階から、実行、メンテナンスまでをサポートすること。
- **操作性**：直感的にわかりやすいインタフェースを備えること。
- **柔軟性**：必要に応じた機能の追加や既存機能の入れ替え、再定義を効率よく行うことができ、汎用的に利用できること。

対話システムの開発支援環境に関する研究として、これまでに LOTOS[1]、CSLU Toolkit[2][3]などが構築されている。これらはいずれも、GUI環境で音声認識、音声合成などのコンポーネントを組み立てることによって、対話システムを開発するツールである。前者は、市内情報案内のための対話システム開発を支援する目的で構築されたものである。また、後者は、対話システムを研究目的で利用するための開発支援環境の提供を目指しており、広く公開されている。両ツールとも、スクリプトを記述するコンポーネントが用意されており、必要な機能を追加することはできるが、音声認識エンジンを変更するといった、既存の機能の拡張や再定義は難しく柔軟性が無い。

また、音声コンテンツを記述するための言語として、VoiceXML[4]が開発されており、電話音声ポータルサービスの構築などに利用されている。VoiceXMLによって記述された対話は、基本的にシステムプロンプト音声と利用者の音声入力またはDTMF入力の繰り返しにより進行する。VoiceXMLはボイスブラウザと呼ばれるサーバでの動作を前提として設計されており、そこでの機能を拡張することは難しく、柔軟性に欠ける。

対話システム開発支援環境が柔軟性を持つためには、スクリプトの記述のような機能拡張のためのコンポーネントを提供するだけでなく、新しいコンポーネントの追加や、既存のコンポーネントの再定義ができることが望ましい。

3. 開発支援ツール TEDDI

3.1. TEDDI の概要

本開発支援ツール TEDDI は、設計、動作、変更、改良など、電話を利用する対話システムの開発サイクルをサポートする。

音声認識、音声合成などの対話システムの一般的な機能をコンポーネントと呼ぶ基本単位に分割し、コンポーネントのインスタンスをブロックとして GUI キャンパス上に表示する。ブロックをキャンパス上でマウス操作により接続することにより、対話システムの処

理の流れを設計できる。このことにより、それぞれのブロックの機能を明確化でき、ブロックの入れ替えなどによる変更が容易になる。また、処理の流れを可視化することにより、全体の構成を把握しやすくなる。

また、本ツールは、機能の拡張性の高さに重点を置いて構築している。コンポーネントをオブジェクト指向におけるクラスとして表現し、ブロックを3つのクラスのオブジェクトとして実現した。オブジェクト指向に基づく方法ですべてのブロックの内部構造を統一し、機能拡張のための機構を用意することによって、柔軟な機能の変更や追加が可能となっている。

3.2. TEDDI の実装

開発支援ツール TEDDI は Windows2000 上で Ruby[5] を使用して実装した。Ruby はオブジェクト指向のスクリプト言語であり、簡単な記述で豊富な機能を実現できる。多くのクラスライブラリを持ち、C 言語による拡張ライブラリの作成も可能である。また、GUI ライブラリも提供されており、TEDDI では Ruby/Tk ライブラリによって GUI を作成した。本設計支援ツールは Ruby インタプリタ上で動作する。本ツールで開発した対話システムは、Dialogic 社製電話音声処理ボード (D/4PCI、D/41JCT-LS) によって動作し、実際に電話を介して試用することができる。また、開発した対話

システム設定の読み込みと保存の機能、及び、設計キャンバスを PostScript ファイルとして保存する機能がある。

図 1 に開発支援ツール TEDDI のインタフェースを示す。ウィンドウの中央には設計用の GUI キャンバスがある。左側にブロック追加ボタンがあり、対話システム開発者が左のブロック追加ボタンから必要な機能を選択すると、そのインスタンスが設計キャンバス上に四角いブロックとして表示される。キャンバス上に表示されたブロック間をマウス操作によって接続することによって、対話システムの構成を設計する。また、ウィンドウの右側にはブロックリスト、システム変数リスト、選択中のブロック情報が表示されている。

本ツールでは、電話を利用した対話システムの開発に必要な機能として、以下にあげる 9 種類の基本コンポーネントを備えている。

- **音声再生**: PCM 音声ファイルを電話音声処理ボードにより電話音声として再生する。
- **音声合成**: 音声合成エンジンによって日本語のテキストを PCM に変換し、電話音声として再生する。今回の実装では、音声合成エンジンとしてドキュメントトーカ[6]を利用した。また、音声合成するテキストには、システム変数を埋め込むことができる。

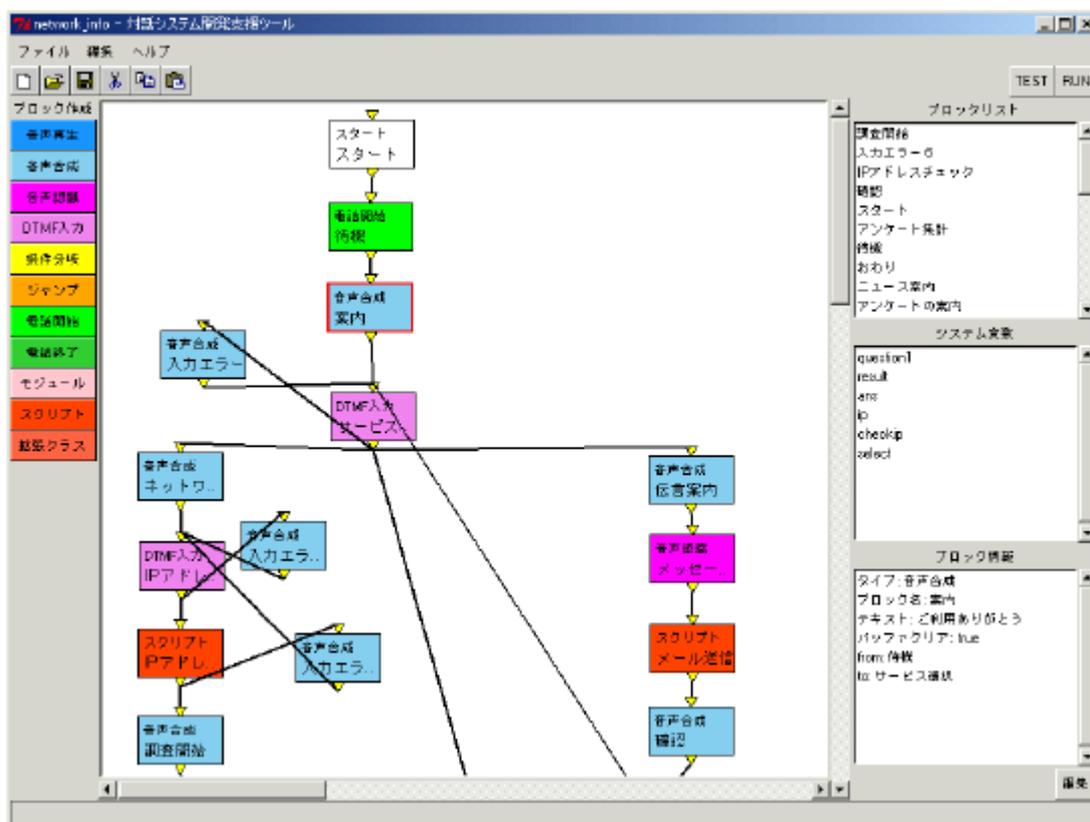


図 1. 開発支援ツール TEDDI のインタフェース

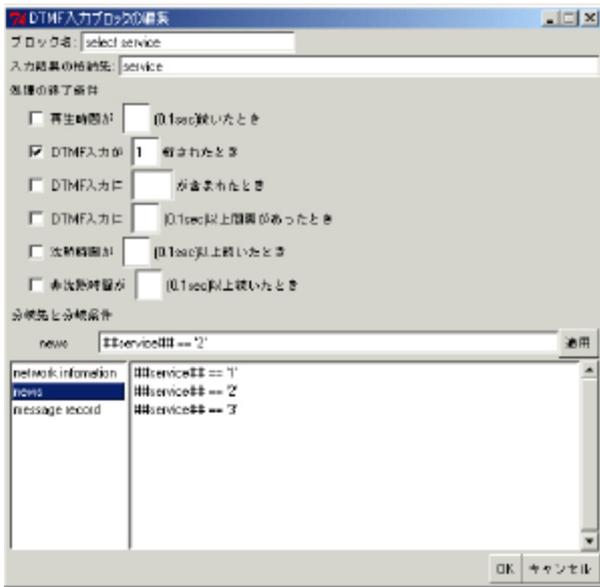


図 2 . ブロック設定ウィンドウ

- **音声認識**：システム利用者が電話で話した発話を電話音声処理ボードによって PCM 音声として録音し、その音声を認識エンジンにより日本語文字列に変換し、その結果を変数に格納する。今回の実装では、音声認識エンジンとして日本語ディクテーション基本ソフトウェア[7]を拡張し、音声認識エンジンの動作するサーバと TCP ソケットによって通信している。また、録音した音声はファイルとしても保存される。
- **DTMF 入力**：利用者が行った DTMF 入力を電話音声ハードウェアで獲得し、変数に文字列として格納する。
- **条件分岐**：各ブロックへの分岐条件を Ruby の式として記述する。分岐条件式には、システム変数を利用することができる。
- **ジャンプ制御**：通常はマウス操作でブロック間を接続して制御の流れを指定するが、このブロックでは直接次ブロックを指定する。
- **モジュール呼び出し**：別に作成し保存された対話システム設定をモジュールとして呼び出す。
- **電話待機**：電話がかかってくるまで待機し、電話処理ボードで電話呼び出しを検知したら電話を受けて次の処理へ移行する。
- **電話切断**：電話ボードで電話を切断の処理を行う。

各ブロックには、ブロック名や次ブロックへの分岐条件などの共通の設定事項や、コンポーネントに固有なパラメータなどの設定事項が用意されており、開発者が設定ウィンドウによってカスタマイズすることが

可能である。DTMF ブロックの設定ウィンドウを図 2 に示す。設定ウィンドウにはいくつかの基本的な設定項目が用意されており、コンポーネントの処理に関する知識がなくてもブロックを編集できる。

なお、特別なブロックとして開始ブロックがある。開始ブロックは、対話システムの処理の開始点を表しており、システム変数の初期化などを行う。

3.3. 内部構造とオブジェクト指向に基づく拡張機能

本開発ツールでは、コンポーネントはオブジェクト指向におけるクラスとして表現しており、コンポーネントのインスタンスであるブロックはオブジェクトとして実現される。それぞれのブロックは、以下の3つの機能を持つオブジェクトから構成されている。

- **情報オブジェクト**：ブロックの名称、ブロックのタイプ、コンポーネントごとの設定（音声合成で読み上げるテキストや、音声認識のパラメータなど）、次ブロックと遷移条件の対応リストなどを保持する。
- **GUI 情報オブジェクト**：GUI キャンバス上での表示位置や表示色などの情報を保持し、接続処理などのためのメソッドを持つ。
- **動作定義オブジェクト**：ブロックの動作定義をメソッドとして保持する。

このうち、GUI 情報オブジェクトは、すべてのコンポーネントに共通するクラスのインスタンスであり、他の2つのオブジェクトは、共通の機能を定義したクラスを継承し、コンポーネント固有の機能を追加したクラスのインスタンスである。図 3 にクラスの関係を示す。

実行の際には、処理スレッドがブロックの動作定義オブジェクトの実行メソッドを実行し、情報オブジェクトにある次ブロックへの分岐条件に従い、処理ブロックを遷移することより全体の処理を実行する。この様子を図 4 に示す。

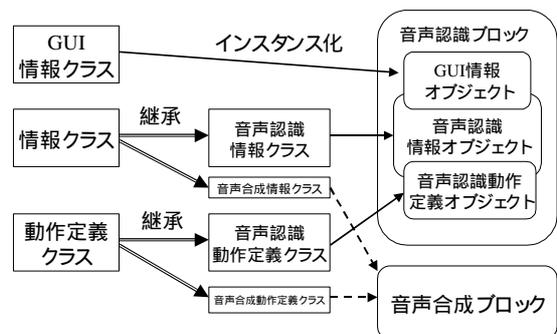


図 3 . クラスの継承関係

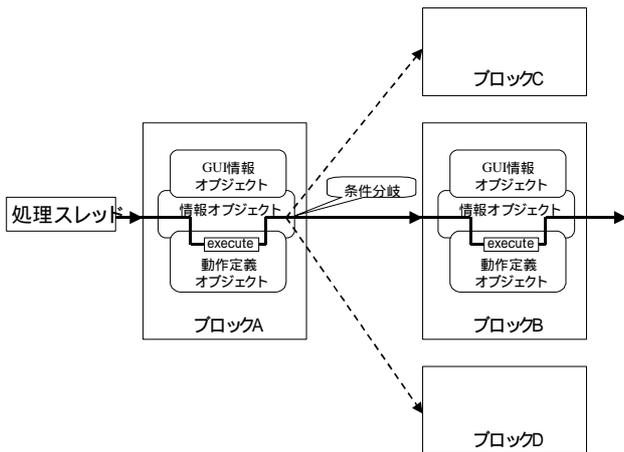


図4 . ツール内部での動作の実現

TEDDI では、前述した基本コンポーネント以外に、機能拡張のための機構として以下のコンポーネントを用意している。

- **スクリプト記述**：システム開発者が Ruby スクリプトを記述することにより、基本コンポーネントで記述されていない処理を追加できる。
- **クラス定義**：動作定義オブジェクトのクラスを新たに定義することにより、新たなコンポーネントを作成することが可能。

これらの機能拡張コンポーネントのブロックは、基本コンポーネントのブロックと同様に、情報オブジェクト、動作定義オブジェクト、GUI 情報オブジェクトの3つから構成されている。そのため、そのほかのブロックと区別することなく実行することができる。

スクリプト記述ブロックでは、対話システム設計者は Ruby のスクリプトによって対話システム固有の処理を記述することができる。記述されたスクリプトは、情報オブジェクト内に文字列情報として格納され、実行の際には動作定義ブロック内の実行メソッドでこの文字列が評価されることによって、設計者が記述した処理が実現される。このような実現方式を採用したことにより、Ruby スクリプトで記述できる範囲の機能を、基本コンポーネントと同じ構成で対話システムに組み込むことができる。

クラス定義コンポーネントでは、対話システム設計者が独自の動作定義オブジェクトを指定することができる。この際に、既存のコンポーネントの動作定義クラスを継承するクラスを作成することにより、効率よくコンポーネント定義が可能である。既存のコンポーネントを拡張した場合のクラス関係を図5に示す。この例では、音声認識コンポーネントの動作定義クラス

を継承し、ユーザ定義の音声認識クラスを作成している。既存のクラスを継承することで、音声認識サーバとの通信処理などをそのまま利用し、複数の音声認識サーバと通信して最もよい認識結果を選択するという機能の拡張が実現できるものと考えている。

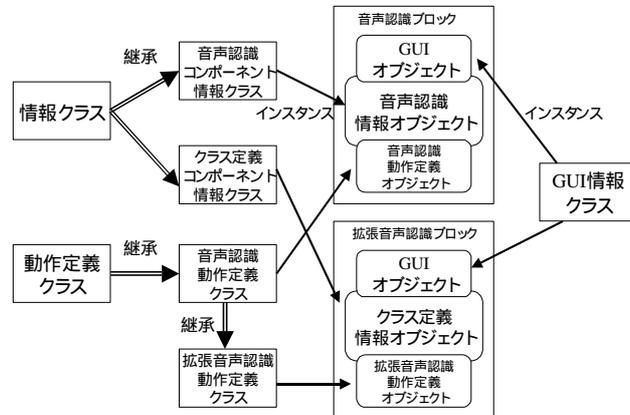


図5 . 継承を用いたクラス定義

4. TEDDI を用いた開発事例

対話システム開発支援ツール TEDDI の有用性を確認するため、実際に本ツールを用いて対話システムの開発を行った。その開発事例として、被験者実験での電話音声収録システムと、ネットワーク状況自動応答システムの開発について述べる。

4.1. 電話音声収録システム

被験者による電話音声収集実験において、音声収録するためのシステムの開発を開発支援ツール TEDDI 上で行った。実験は、被験者が収録システムに電話をかけ、システムからの音声プロンプトを聞いてレストラン検索と天気予報問い合わせの2つのタスクについて発話をするものである。

この実験で使用した収録システムは、基本コンポーネントだけで実現でき、11個のブロックから構成される簡単な対話システムである。その収録システムの開発キャンバスを図6に示し、使用したブロックの種類を表1に示す。システムの音声プロンプトは音声合成ブロックにより実現し、音声の録音には音声認識ブロックを用いた。また、2つの発話タスクの選択にはDTMFブロックを使用した。本システムの開発は、開発支援ツールを用いることにより数分で構築することができ、実験方法の調整による収録方法の変更にも柔軟に対応することができた。実験では本システムを利用して89名の音声を収録し、開発した対話システムが適切に動作することを確認した。

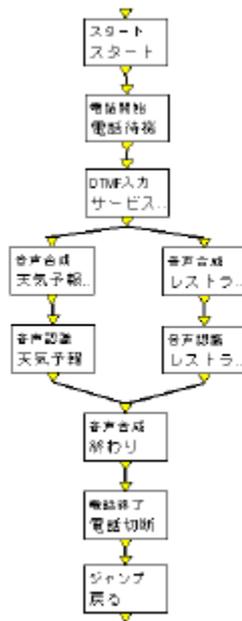


図 6 . 電話音声収録システムの開発キャンパス

4.2. ネットワーク状況自動応答システム

名古屋大学のネットワーク状況を電話で応答する対話システムの開発を、TEDDI を用いて行った[8]。

本ネットワーク状況応答システムは以下の 3 つのサービスを提供する。

- ネットワーク利用者が IP アドレスを指定し、そのアドレスまでのネットワーク状況を調査し報告する。
- ネットワーク工事などの情報を通知する
- ネットワーク管理者への伝言を記録する

応答システムの利用者は、ネットワーク利用に何らかの問題が発生したときに、応答システムが設置されている電話番号に電話をかける。その後、合成音声による案内に従い、DTMF 入力によってサービス内容の選択や IP アドレスの入力を行う。

ネットワーク状況応答システムは、全 31 個のブロックで構成される。そのブロック構成を表 1 に示す。本応答システムでは、利用者からの入力はすべて DTMF 入力によって行っている。また、利用者からの伝言の記録には音声認識ブロックを利用している。さらに、基本コンポーネントでは実現できない以下の 4 つの機能を、スクリプト記述ブロックによって実現した。

- 入力された IP アドレスの検証
- 指定された IP アドレスおよび大学外部のサイトに対して ping によってネットワーク状況を調査する処理
- 伝言が残されたことを伝える電子メールの送

信

- 利用後のアンケートの集計

開発したネットワーク状況自動応答システムは、名古屋大学情報連携基盤センターにて運用中である。

表 1 . システムのブロック構成

ブロックの種類	音声収録	ネットワーク
開始	1	1
音声合成	4	17
音声認識	2	1
DTMF 入力	1	5
ジャンプ制御	1	1
電話開始	1	1
電話切断	1	1
スクリプト記述	0	4
合計	11	31

5. おわりに

本稿では、GUI 上で電話を用いた対話システムを設計できる、対話システム開発支援ツール TEDDI について述べた。本ツールでは、オブジェクト指向に基づいてコンポーネントを実現した。また、2 つの拡張用コンポーネントを利用することによって、開発者が必要な機能を効率よく柔軟に拡張することが可能である。

また、本開発支援ツールを用いた開発事例として、電話音声収録システムとネットワーク状況応答システムの開発について述べた。

今後、本開発支援ツールを用いて、電話を用いた対話システムの評価実験や収録実験で得た電話音声の認識実験などに取り組む予定である。

参考文献

- [1] T. Nouza, J. Nouza, “Graphic platform for designing and developing practical voice interaction systems”, Eurospeech 2001, pp.1287-1290, 2001.
- [2] S.Sutton, D.G.Novick, R.Cole, P.Vermeulen, J.de Villiers, J.Schalkwyk and M.Fanty, “Building 10000 Spoken Dialogue Systems”, Proc. of ICSLP’ 96, pp.709-712, 1996.
- [3] <http://cslu.cse.ogi.edu/toolkit/>
- [4] <http://www.w3.org/TR/voicexml20/>
- [5] <http://www.ruby-lang.org/>
- [6] <http://www.createsystem.co.jp/>
- [7] 河原達也, 住吉貴志, 李晃伸, 武田一哉, 三村正人, 伊藤彰則, 伊藤克巨, 鹿野清宏, 連続音声認識コンソーシアム 2000 年度版ソフトウェアの概要と評価, 情処研報, 2001-SLP-38-6, pp.37-42, 2001.
- [8] Y. Yamaguchi, K. Ito, N. Kawaguchi, S. Matsubara and Y. Inagaki, “Design and Development tool for Telephone-based Network Information System”, Proc. of SCI2002, vol. III, pp.285-290, 2002