

マルチモーダル対話システムのための UI 設計支援ツール

Multimodal User-interface Editor
for Multimodal Dialogue System

松浦 博
Hiroshi Matsuura

神尾広幸
Hiroyuki Kamio

内山ありさ
Arisa Uchiyama

郡田美香
Mika Ko'orita

田村正文
Masafumi Tamura

新田恒雄
Tsuneo Nitta

株式会社東芝 マルチメディア技術研究所
Multimedia Engineering Laboratory, Toshiba Corporation

Abstract: This paper describes a user interface design-support tool that simplifies the design and evaluation of multimodal user interfaces (MMUIs). With this tool, a designer first prepares screens by dragging and dropping UI-objects onto a card using a mouse. These UI-objects consist of both visible objects, such as button, character recognition, text, and image objects, and invisible objects, such as speech recognition, timer, sensor, sound and text-to-speech objects. Next, the designer describes scenarios for multimodal dialogue by linking between UI-objects and other UI-objects or cards. Screen data and scenarios are then converted into UIScript, which describes multimodal dialogue through frame-like expression. At this point, the UIScript can be reviewed and revised as desired using an editor. The MMUI based on this UIScript can then be executed using MultiksDial, a multimodal dialogue system. This enables the designer to quickly evaluate and modify the MMUI.

1. まえがき

様々なシステムの高機能化が進む中で、誰でも短時間で使い方が習得でき、しかも簡単に使えるユーザインタフェース (UI) の開発が望まれている。我々は社会自動化情報システムへのマルチモーダルUI (MMUI) 適用を中心にUIの改良を試みてきた。MMUIはユーザが各自に合った使い方を選択したり、システム側からも場面場面に応じて最適な使い方を提供できるなどの特長がある。我々は、これまでに入出力の双方をマルチモーダル化した対話システムMultiksDialを開発すると共に、情報案内 [1, 2], 券売機 [3] など多様な分野への応用を試みた。しかし、MMUIでは対話のシナリオが複雑になるため、優れたUIを開発するに際して評価と改良のプロセスを繰り返し行う必要がある。このためアプリケーション開発の期間が長期化する傾向があった。今回、我々はMMUIの開発を迅速に行うラピッドプロトタイプ環境を整備した。この環境では、UI記述言語UISCRIPTを定義して

使用している。一方、記述言語に基づくテキストデータから設計を進める手法は、すべての開発者にとって、必ずしも使い勝手が良いとは言えない。例えばハードウェア設計の分野では、テキストデータに比べ直観的に把握しやすい図面による設計を行った後、テキストデータに変換する開発ツールが多く利用されている。同様、MMUI開発においても、画面構成と対話のシナリオ設定をGUI(Graphical User Interface)の手法を導入して進めることが望ましい。そこで、今回開発者にとっての使い勝手向上を目的に、UI設計支援ツール Muse(Multimodal User-interface Editor)を開発した。

2. マルチモーダル対話システム MultiksDial

最初に、MMUIを実行するマルチモーダル対話システム MultiksDial(Multimodal Keyword-based Spoken Dialogue System)について説明する。MultiksDial は図1に示すように、入力チャンネルとして音声認識・タッチスクリーンからの入力・文字認識を、出力チャンネルとして文音声合成・録音合成・ディスプレイ表示を備えている。また、ユーザの接近・ハンドセットの取上げ・ハンドセットの耳あてを光電センサによって検知する機能を持ち、対話の組立てに利用している。

音声認識部はキーワードスポッティングに基づく不特定話者単語認識を実時間処理する専用ハードウェア KeySpot(Keyword Spotting Unit)で構成されている [4]。文字認識部には指でタッチスクリーン上に書いた平仮名・片仮名・数字を認識するソフトウェアを使用した [5]。文音声合成部は漢字仮名コードで入力した文をケプストラム方式によって音声に変換し出力する専用ハードウェアである [6]。録音合成部はワークステーション上の録音合成ソフトを用いた。ディスプレイ表示部は静止画像・イラスト・文字・ボタンなどを表示する。センサ情報はマルチモーダル制御ボードを経由して、ワークステーションに送られる。ワークステーション上のマルチモーダル対話マネージャは音声認識結果、文字認識結果、タッチスクリーンデータ、センサ情報を受取り、UISCRIPTで記述された対話シナリオを参照しながら文音声合成などの出力チャンネルを起動する。

MultiksDial においては音声認識部など処理量の多い機能を専用ハードウェア化し、ユーザの動作に対する素早い応答を確保している。アプリケーションシステム構築の際は、コンピュータと対話することへのユーザの抵抗感を弱めるられるよう、多様な入出力チャンネルをその特性に応じて利用することが重要になる。

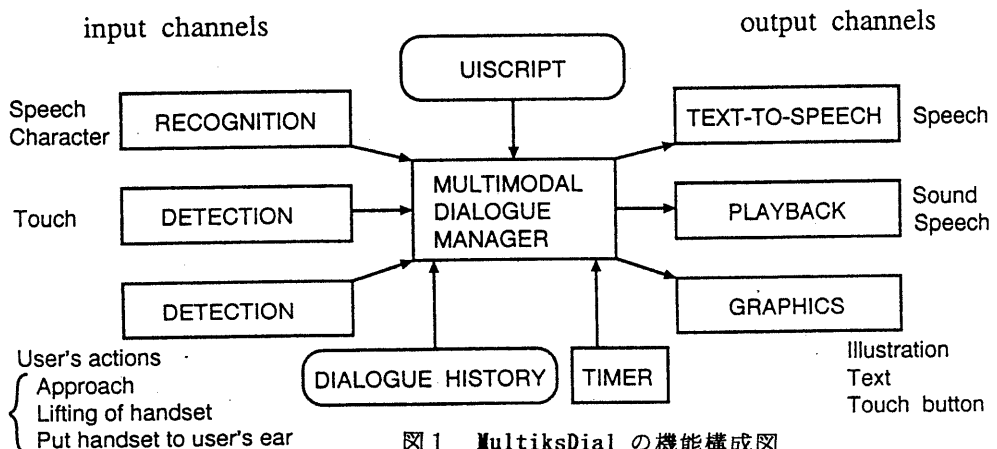


図1 MultiksDial の機能構成図

3. UI記述言語 UISCRIPT

UISCRIPTはMultiksDial のすべてのマルチモーダル動作を図2に示すフレーム表現に類似の形式を用いて記述する言語である。UISCRIPTはカラー、フォント、変数、マクロ、関数、処理、画面、ウィンドウ、データベース、文字認識、評価データについてそれぞれ記述する複数のサブファイルから構成される。以下に各サブファイルの内容を説明する。

- a. カラーファイルには使用する色を、イラストなどを保存するラスターファイルで用いる色と合わせて 256個以内で指定する。
- b. フォントファイルには文字フォントを定義する。
- c. 変数ファイルには演算処理を記述するマクロファイルで用いられる変数や対話シナリオを記述する処理ファイルで用いられる変数を記述する。
- d. マクロファイルにはマクロ処理を記述する。マクロ処理とはテンキー入力による金額計算・表示処理などの汎用的な処理および、アプリケーションの中で再三、出現するまとまった処理である。マクロ処理で使われる演算には、条件処理・四則演算、文字列の複写・結合、データベースの追加・削除・検索・検索条件設定などがある。
- e. 関数ファイルは処理の内のまとまりのある部分を関数として登録する。
- f. 処理ファイルには音声認識・文字認識・文音声合成・録音合成・センサ情報などのマルチモーダルチャンネルの動作が対話シナリオに沿って記述される。
- g. 画面ファイルにはディスプレイ表示画面を記述する。具体的には、背景を示すバックグラウンドや画面の区分けを示すボードの位置・大きさ・色を指定することによって画面の概略構成を作成する。ボードの中には文字やイラスト・静止画などのラスターファイルを表示させる。また、タッチ入力するボタンをあらかじめ用意した数種のボタンの中から選んで位置・大きさ・色・名称を指定する。なお、イラストなどを表示した領域をタッチ入力領域として指定する機能がある。
- h. ウィンドウファイルにはディスプレイ画面の一部をコイン投入などの仮想デバイスとして使用するための記述を行う。各ウィンドウはそれぞれ独立したプロセスによって制御され、各状態を送受信する機能を有する。

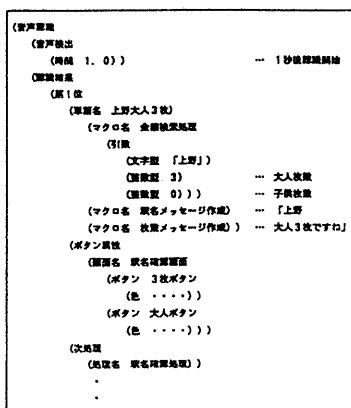


図2 UISCRIPTによる音声認識チャンネルの記述の一例

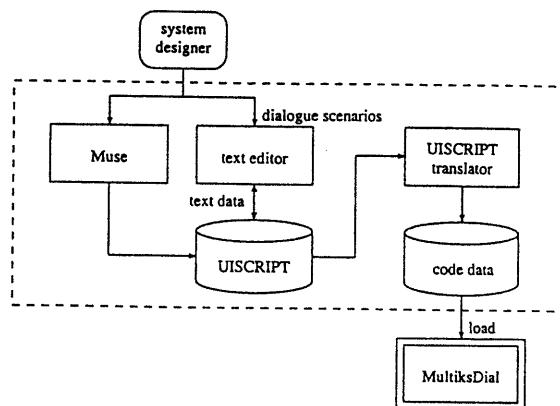


図3 UI設計支援ツールの機能構成図

- i. データベースファイルには例えば駅の券売機に応用する場合、駅名や運賃体系などのデータが格納される。
- j. 文字認識ファイルには認識対象、入力データ精度などの文字認識の条件設定を行う。
- k. 評価データファイルにはユーザによる音声入力、タッチ入力やセンサ検知情報の行われた時間を収集・記録する。

図3に示すように、UISCRIPTの各サブファイルは解析プログラムにより字句解析・構文解析・リンクされ、コードデータである中間ファイルに変換される。ここで、文字認識と評価データ以外のサブファイルについては複数個、リンクすることも可能である。次に、中間ファイルを参照して、MultiksDialの実行プログラムを起動し、MMUIの評価を行うことになる。すなわち、開発者自身がMMUIを実際に体験しながら開発を進めることができる。また、実ユーザに近い人に使用してもらい、作成されたMMUIの問題点を抽出することも可能である。

UISCRIPTの特徴としては、UI記述のルール習得がC言語などに比較して容易であること、作成も迅速に行えることがあげられる。すなわち、各入出力チャンネルの動きを記述する時に、デフォルト設定や既存のマクロ処理を利用することができる。また、解析エラーを無くせば少なくとも動作はするため、UI設計上の間違いや改良点を発見しやすく、変更も容易である。さらに、日本語で記述されているので、時間が経過してから見直したり、他の設計者が修正する際にも支障が少ない。また、今まで仕様検討・UI作成・評価など別の部門あるいはメンバーが行っていたところを、一人で行うことも可能になり、開発のスピードアップに大きく貢献する。

4. UI設計支援ツール Muse

Museでは、MMUIの設計をマウスボタンのドラッグ・アンド・ドロップなどの操作とアイコンの表示によるGUI環境で行うことで、UISCRIPTの作成をさらに容易にする[7]。Museの主な機能は編集機能、カード部品管理機能、プロトタイプ実行機能、UISCRIPTへの変換機能、対話シナリオマップ表示機能からなる。また、Museはオブジェクト指向に基づくXウィンドウ上で動作するアプリケーションであり、画面を示すカードや画面上に配置される部品(UI-object)はすべて独立したオブジェクトとして実装される。画面遷移や合成音の出力などのタイミングはオブジェクト間のメッセージ通信によって与えられる。以下に各機能について説明する。

- a. 編集機能を用いてカードにUI-objectをレイアウトし、一画面ずつデザインする。次に、カードやUI-object間にリンクを張ることによって、各UI-objectがイベントを受けた時の遷移先を決定する。
- b. カード部品管理機能は種々のUI-objectを管理し、編集機能に提供する。
- c. プロトタイプ実行機能は編集機能によって設定したリンク情報をもとにMultiksDialを動作させる。
- d. UISCRIPTへの変換機能は、作成した画面情報や対話シナリオをUISCRIPT形式に出力する。
- e. 対話シナリオマップ表示機能は設計者が対話シナリオ全体を容易に把握できるようにするために、画面遷移の全体像を表示する。

次に、Museによる開発手順を述べる。

①画面の設計

マウスの操作によって、画面上にUI-objectを配置し、画面を設計する(図4)。Museでは、画面を構成するUI-objectとして、イメージ部品とテキスト部品がある。また、ユーザとの対話を行うUI-objectであるボタン、音声認識、タイマ、センサなどの入力オブジェクトや録音合成、規則合成などの出力オブジェクトをカード上に配置する。

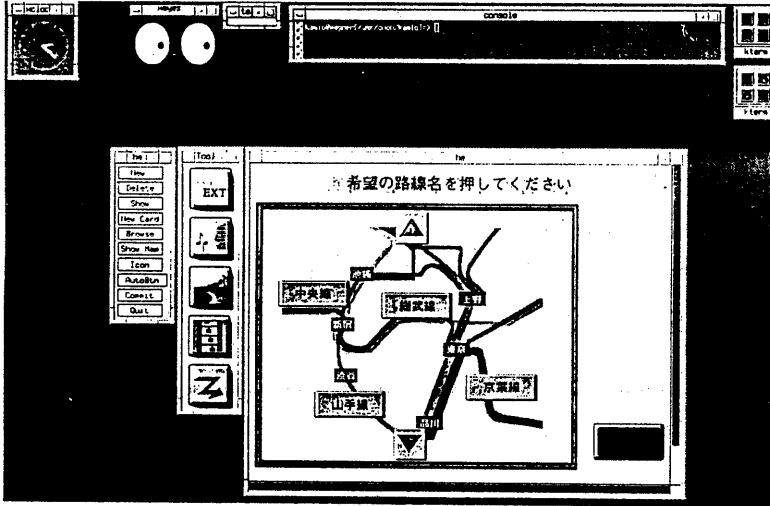


図4
UI設計支援ツール
による画面設計

②対話シナリオの設定

画面の設計の次に、システムの対話シナリオを設定する。対話シナリオの設定はUI-objectと他のUI-objectや画面との間にリンクを張ることによって行う(図5)。UI-objectはユーザからのイベントを受けるとリンク先にメッセージを送る。このメッセージによってリンク先のオブジェクトが活性化し、画面遷移や一定の動作が引き起こされる。

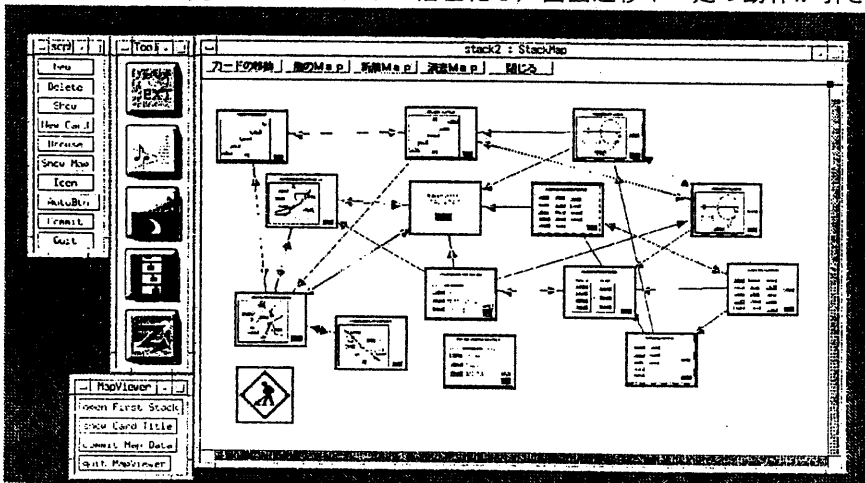


図5 リンク付けによる対話シナリオ作成

③プロトタイプ実行

MMUIのプロトタイプ実行は設定された対話シナリオに基づいて行われる。Xウィンドウ

では入力機能としてマウスとキーボード以外は用意されていないため、Museではマルチモーダル入力を受け付けるデバイスマネージャを用意した。デバイスマネージャはユーザの様々な入力をXウィンドウのマウスイベントに変換し、それぞれのUI-objectへ送信する。例えば、音声認識された場合(図6)、デバイスマネージャは認識結果を受けとり、音声認識部品にイベントとともに3位までの認識結果を送信する。イベントを受けた音声認識部品は、認識結果に対応したリンク先にメッセージを送り、例えば画面の遷移や規則合成の出力を行う。プロトタイプ実行によって、問題点が発見されると②③へ戻る。

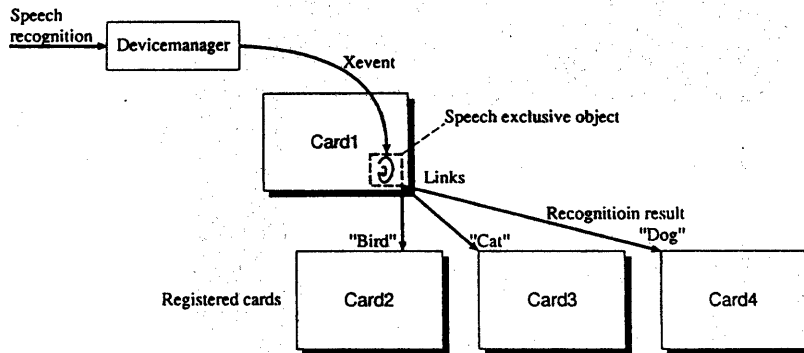


図6 音声認識チャンネルからのメッセージの流れ

④UISCRIPTへの変換

最後に、UISCRIPTへ変換し文字認識などMuseで現在対応していない機能をエディタで記述する。

5. まとめ

マルチモーダル対話システムMultiksDialのためのUIラピッドプロトタイピングを目標にするUI記述言語UISCRIPTを開発した。さらに、画面構成と対話のシナリオ設定をGUIの手法によって行うUI設計支援ツールMuseを構築した。Museで作成されたデータはUISCRIPTに変換され、UIの細部について記述することができる。この結果、個々のアプリケーションを開発する際、使い勝手の良いシステムの構築とUI評価に基づく改良が容易に行える環境が整った。

文献

- [1] H. Matsuura, Y. Masai, J. Iwasaki, S. Tanaka, H. Kamio, T. Nitta: "Applying a spontaneous speech recognizer, a touch-screen, a rule-based speech synthesizer, and photoelectric sensors to a multimodal dialogue system", proc. of int. symp. on spoken dialogue, 2-1-4(1993-11).
- [2] 神尾, 松浦, 正井, 新田: "マルチモーダル対話システムMultikDial", 信学論, D-II, Vol. J77-D-II, No. 8, pp. 1429-1437(1994-08).
- [3] 松浦, 新田, 神尾: "マルチモーダル対話の社会情報システムへの応用", 東芝レビュー, pp. 16-19(1994-1).
- [4] Y. Masai, J. Iwasaki, S. Tanaka, T. Nitta, M. Yao, T. Onogi, A. Nakayama: "A Keyword-Spotting Unit for Speaker-Independent Spontaneous Speech Recognition", Proc. of ICSP 94, S23-9.1, pp. 1383-1386(1994-9).
- [5] 河村, 由良, 比田井, 田中, 南川: "方向成分密度法によるオンライン手書き漢字認識", 信学全大, D-528(1991-3).
- [6] 原, 江川, 新田, 小林, 斉藤: "文音声合成ボードの試作", 信学全大, A-6(1992-3).
- [7] H. Kamio, M. Koarita, H. Matsuura, M. Tamura, T. Nitta, "A UI DESIGN SUPPORT TOOL FOR MULTIMODAL SPOKEN DIALOGUE SYSTEM", Proc. of ICSP 94, S22-11.1 pp. 1283-1286(1994-9).