

## マルチモーダルUIとラピッドプロトタイピング

新田 恒雄 神尾 広幸 雨宮 美香 松浦 博 内山 ありさ 田村 正文

(株) 東芝 マルチメディア技術研究所  
〒 210 川崎市幸区柳町70

**概要** マルチモーダルUI (MUI) の設計-評価を短期間に行うラピッドプロトタイピング開発環境について述べる。開発環境は、MUI設計支援エディタ Muse、UIScript変換、およびマルチモーダル対話プラットホームMultiksDialのツール群からなり、Museで作成したMUIを中間言語(UIScript)を介してcode-dataに変換した後、マルチモーダル対話プラットホームMultiksDialの上で実際に操作・評価することができる。Museでは各UI-object(ボタン/音声認識/センサなどの入力オブジェクト、イメージ部品/テキスト部品/録音合成/規則合成などの出力オブジェクト)のプロパティ設定をdialogue boxを利用して簡単に行えるようになっている。

ラピッドプロトタイピング開発環境の整備により、様々なタスクを対象に短期間にMUIを試作・評価することが可能である。

## Multimodal UI and Its Rapid-prototyping

Tsuneo NITTA, Hiroyuki KAMIO, Mika AMAMIYA, Hiroshi MATSUYAMA,  
Arisa UCHIYAMA, Masafumi TAMURA

Multimedia Eng. Lab., TOSHIBA CORP.  
70 Yanagi-cho, Saiwai-ku, Kawasaki 210

**ABSTRACT:** In recent years, we have developed various types of multimodal dialogue systems, including a ticket vendor, an ATM, and an information kiosk. Because the designing of multimodal user-interface (MUI) is more complicated than that of existing UI based on graphical UI (GUI) and has not obtained its regular method yet, the development of a multimodal dialogue system requires a long span of time. Through the experience on iterative design of the above mentioned applications, we came to think of the importance of the rapid-prototyping of multimodal dialogue systems for collecting data systematically over various types of application areas.

In this paper, we describe a platform of multimodal dialogue systems and rapid-prototyping by using a multimodal UI design support tool with which system developers can design panels, set properties of input/output channels, describe plan-goal scenarios, and evaluate multimodal UI easily.

## 1. はじめに

コンピュータのユーザインタフェース（UI）は、現在、グラフィカルUI（GUI）が中心となり、マウスとキーボード操作による対話が利用されている。しかし、文字・音声・映像といった多様なメディアを介した対話をGUIの枠組みで行うのは適当と言えず、新しい対話の枠組みとしてマルチモーダルUI（MUI）が注目されている。しかしユーザとシステムの双方が、複数のメディアを駆使する対話を円滑に進めることは非常に困難であり、現在はインプリメンテーション技術（実時間・並行処理、時間同期など）、対話技術（対話モデル、応答戦略、ユーザ適応、対話の制御と統合など）、ラピッドプロトタイプング技術、および評価技術といった課題を一つづつ解決するフェーズにある[1]。

我々はこれまでに様々なタスク（券売機、ATM、情報案内システムなど）を対象に、MUIの研究を進めてきた[2],[3]。MUIの設計は既存のGUIベースの設計に比べ複雑なため、開発により多くの時間を要する。このため、様々なタスクに亘るマルチモーダル対話システムを効率良く開発し、対話データを組織的に収集することのできるラピッドプロトタイプング技術が大変重要になる。このような観点から、我々はこれまでMUI開発環境の整備を進めてきた[4],[5]。本文では、最初にマルチモーダル対話プラットフォームを、続いてMUIの設計から評価を短期間に実行できるMUI設計支援ツールを用いたラピッドプロトタイプングを最近の改良点を含めて説明する。

## 2. マルチモーダル対話プラットフォーム

図1はMUI研究を目的に構成したマルチモーダル対話のためのプラットフォーム“MultiksDial (Multimodal Keyword-based Spoken Dialogue System)”の機能構成を示している[2]。最新のシス

テムは入力チャンネルとして音声入力/タッチスクリーン入力/手書き文字入力を、出力チャンネルとして文音声合成出力/録音合成出力/ディスプレイ表示（文字・アニメ（対面エージェントなど）・図形・静止画）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、これらを用いてタイムリなガイダンスを行うなど、ユーザとの対話組み立てに利用している。

音声入力部（“KeySpot (Keyword-spotting unit)”）は、SMQ/HMM（SMQ: Statistical Matrix Quantization）に基づくサブワード単位認識方式を採用しており[6]、孤立単語/連続音声/単語スポッティングの3つのモードを選択使用できる[7]。最新のユニットではタスクあたり、孤立単語モードで1,000語、連続音声およびスポッティングモードで500語を実時間で識別することができる。また音声入力部には、2マイクロホンを用いた適応ノイズキャンセラが装備されており、高騒音下でも利用が可能である[8]。音声入力の大きな利点は気楽に直接、目標にアクセスできることにあるが、ユーザにあまり多くの制約を与えたとこの利点も損なわれてしまう。ユーザの発話行為を束縛しないためには、認識装置が①騒音下においても、②自由発話を、③高速に認識することが必要であるが、KeySpotはこれらの基本性能を満たしている。

次に、手書き文字入力はワークステーション上のソフトウェアとして実装されており、仮名・英数字が入力できる。また、MultiksDialではこの他、ユーザとの対話が円滑に行なえるように光電センサの状態をもとに様々な工夫をしている。例えば、ユーザがハンドセットを取ったことをセンサで検知すると、音声ガイダンスの出力を外部スピーカに対して低く設定しハンドセットのスピーカに切替える。応答チャンネルの切替えは、ユーザがハンドセットを耳に近づけ、ハンドセットを使用することを促すメッセージ、すなわちアフォーダンスとなっていると言える。

マルチモーダル対話マネージャは、音声・手書き文字・タッチボタンおよびセンサ入力情報を受け取ると、後述のUIScriptで記述された対話シナリオに沿って出力チャンネルを起動し、グラフィックス/アニメ/ガイダンス文・音声/応答文・音声を出力する。マルチモーダル対話では、ユーザとシステムの双方が多様なチャンネルを利用し、各々の期待とプラン ( expectation & plan ) を持って対話する。このため、ユーザにシステム側の状態・意図を分かり易く提示することがこれまでのUI以上に要求される。プラットフォームでは、ユーザがその都度入力チャンネルの選択権を持ち、また入力操作手順についてもシナリオ上任意の順序に自由発話で利用できる。この意味でユーザ主導型のUIとなっているが、実ユーザに近い被験者を用いてシステムを総合的に評価すると、かえってシステム主導型にした方が使い易いことが少なくない。MUIにおいては、システムがユーザ側の意図をどれだけ理解でき、またユーザへシステムの意図をどれだけ正確に通知できるかにより、設計方針も大きく変わってくる。このことからラピッドプロトタイプング技術は大変重要であると言える。

### 3. MUI開発支援環境

図2にMUI設計支援エディタ(" Muse (Multimodal User-interface Editor )、UIScript変換、およびマルチモーダル対話プラットフォームMultiksDial からなるMUI開発支援ツールと、これを利用しての作業フローを示す。

#### 3.1 UI記述言語

UIScriptは、MultiksDialの動作を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。図3にUIScriptの例を示した。UIScript(で記述されたファイル)はマルチモーダル対話のシナリオを含む処理のほか、画面、ウィンドウ、カラー、

フォント、変数、マクロ、関数、データベース、評価データなど複数のサブファイルから構成されている[4]。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。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することで、MUIの評価が可能になる。これによって、開発者自身がMUIを実際に体験しながら開発を進めたり、実ユーザに近い人達が試用してMUIの問題点を抽出することができる。

UIScriptは記述ルールの習得が比較的容易で、個々の入出力チャンネルの動作を記述する際に、デフォルト設定や既存のマクロを利用すれば迅速にUIを設計できる。

#### 3.2 MUIエディタ (Muse)

UIの設計、すなわちUIScriptの記述にGUIを導入して開発者の労力を軽減するために、MUI設計支援エディタ Muse を開発した。Museは、エディタ、部品マネージャ、および対話シナリオ・マップビューアからなる。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、画面を示すカードや画面上に配置される部品(UI-object)を独立したオブジェクトとして実装している。画面遷移や合成音出力などのタイミングなど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[5]。

#### 3.3 MUIの設計手順

Muse、UIScript変換、MultiksDialによるMUI開発手順を以下に説明する。

##### 3.3.1 画面の設計:

Muse上でマウス操作により、UI-object(ボタン/音声認識/センサなどの入力オブジェクト、イメージ部品/テキスト部品/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

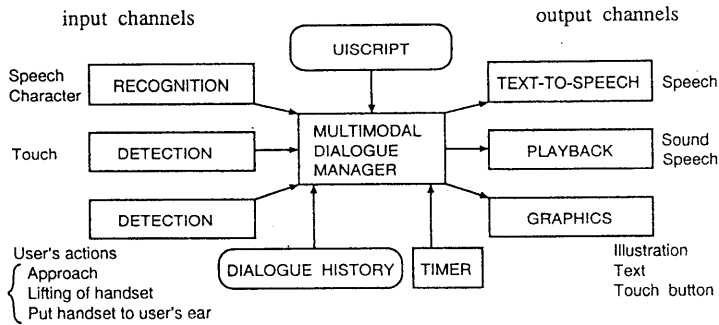
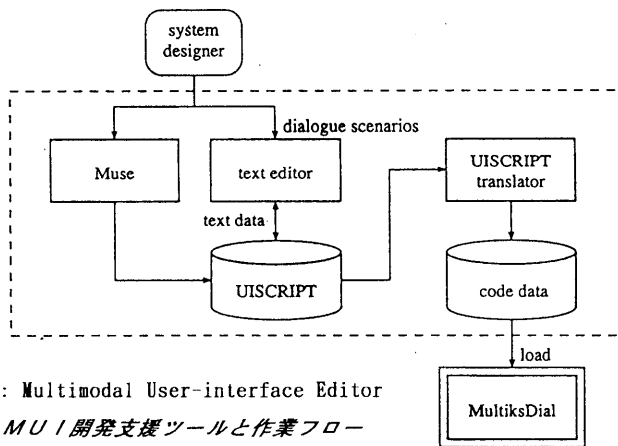


図1 MUIプラットフォームMultiksDialの機能構成



Muse : Multimodal User-interface Editor

図2 MUI開発支援ツールと作業フロー

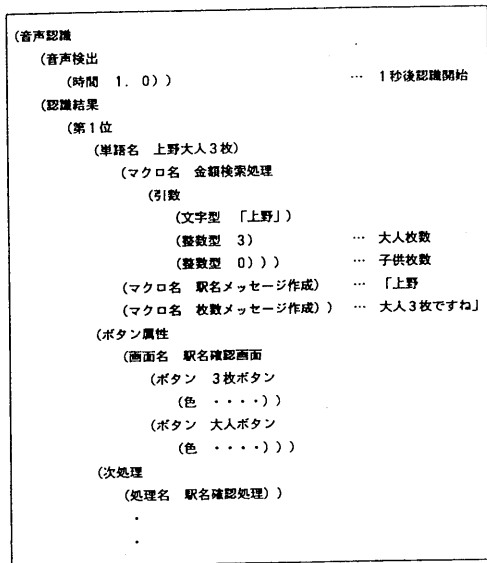


図3 UIScriptの記述例

る。

次に、UI-object の各プロパティをダイアログボックスを表示して設定する。例えば規則合成オブジェクトの場合、音声エディタによるガイダンス文（もしくは応答文）の埋め込み、修正、さらに speaker type / 発声速度 / 音量 / 声の高さ / 出力ポート（ハンドセット or 外部スピーカ）などを TTS-dialogue Box で設定できる（図4参照）。同様に、音声入力の場合には、入力モード（孤

立単語 / 連続音声 / 単語スポッティング）、語彙（セット）とその追加・修正、文法記述、さらにリジェクトの閾値や適応ノイズキャンセラ等のパラメータなどを ASR-dialogue Box で設定することができる。

### 3.3.2 対話シナリオの設定:

UI-object 相互、および画面間にリンクを張る。UI-object はユーザからのイベントを受けるとり

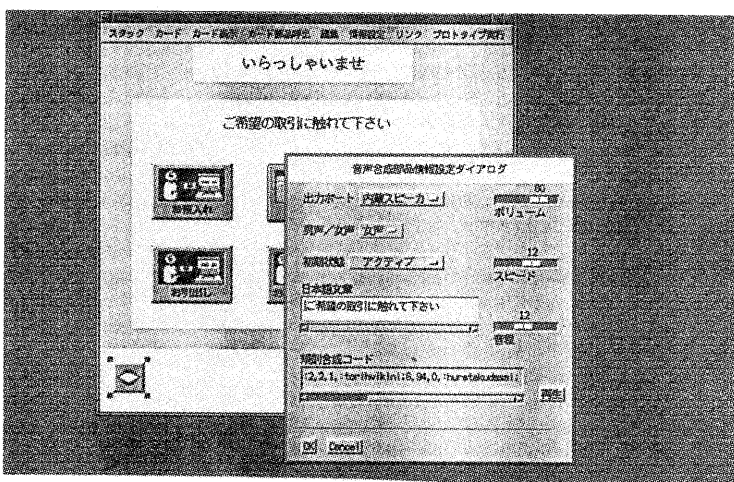


図4 UI-Object に対するダイアログボックスを用いたプロパティ記述 (TTS-dialogue Box の例)

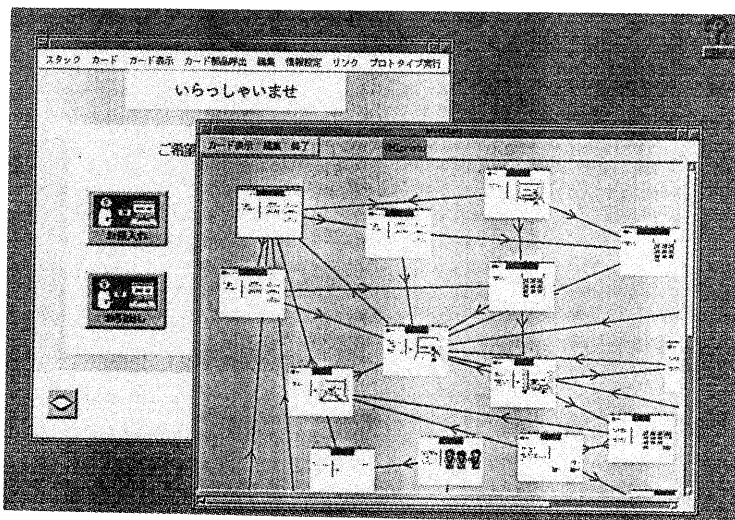


図5 マップビューアによる対話シナリオの鳥瞰

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。最後に、マップビューアにより対話シナリオをチェックする(図5参照)。

### 3.3.3 UIScript への変換:

UIScriptへ変換すると共に、Museで対応できていない機能(現在は例えば文字入力など)をテキストエディタで追加記述する。続いて、プラットフォームMultiksDialで動作するcode dataに変換する。

### 3.3.4 プロトタイプの実行・評価:

プロトタイプの実行は設定された対話シナリオに基づいて行われる。Xウィンドウの入力機能はマウスとキーボードのみのため、マルチモーダル入力を受け付けるデバイスマネージャが用意されている。プロトタイプ実行・評価により問題点が発見されると①～③の該当する作業へ戻り、再設計・テストが行われる。

## 5. む す び

MUIの設計-評価を短期間に行うラピッドプロトタイプ開発環境について述べた。開発環境は、MUI設計支援エディタ Muse、UIScript変換、およびマルチモーダル対話プラットフォームMultiksDialのツール群からなり、Museで作成したMUIを中間言語(UIScript)を介してcode-dataに変換した後、マルチモーダル対話プラットフォームMultiksDialの上で実際に操作・評価することができる。

ラピッドプロトタイプ開発環境の整備によって、様々なタスク(金融システム、情報Kiosk、駅務システムなど)のMUIを短期間に開発評価することが可能になった。また、同一のタスクを客先別・設置箇所別等カスタマイズすることはさらに容易である。

今後の課題としては、各UI-objectのプロパティ設定に使用するdialogue boxの機能拡張、MUI評価機能の組み込みがある。さらに、音声入力・手書き入力などのUI-objectのAPI(Application Program Interface)標準化を通して、MUIのフレームワークを確立することも重要である。

## < 参 考 文 献 >

- [1] 新田: "マルチメディア・サービスとマルチモーダルUI", 情処学研資, SIG-SLP-1-5 (1994-05).
- [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] 松浦, 神尾, 内山, 郡田, 田村, 新田: "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", 情処学研資, SIG-SLP-3-7 (1994-10).
- [5] H. Kamio, M. Ko'orita, H. Matsu'ura, M. Tamura, T. Nitta: "A UI Design Support Tool for Multi-modal Spoken Dialogue System", Proc. ICSP94, S22-11.1, pp. 1283-1286 (1994-9).
- [6] 田中, 正井, 松浦, 新田: "サブワードSMQ/HMMによる不特定話者中語彙音声認識の検討", 音学講論 3-8-1, pp. 109-110(1994-10)
- [7] Y. Masai, J. Iwasaki, S. Tanaka, and T. Nitta: "A Keyword-spotting Unit for Speaker independent Spontaneous Speech Recognition", Proc. ICSP94, pp. 1383-1386(1994-9).
- [8] 新田, 南, 中山, 小野木: "高騒音下における音声認識方式の検討", 信学技報, SP94-20, pp. 45-52 (1994-06).