音声言語情報処理　9－5
（1995. 12. 14）

社会情報システムのための ラピッドプロトタイピングツールMuseの開発

神尾広幸、雨宮美香、内山ありさ、松浦博、新田恒雄
（株）東芝 マルチメディア技術研究所

〒210 川崎市幸区柳町70
044-548-5352
kamio@sp.mmlab.toshiba.co.jp

あらまし　本稿では、マルチモーダルインタフェースの作成と評価を容易に行うツールMuse (Multimodal User-interface-design Support Editor) について述べる。MuseではUI-objectを画面上に配置することでUIの外観を設計し、UI-objectの機能・情報設定やリンクの設定を行うことで動作の設計を行う。Museでは、入力手段としてタッチパネルによるポインティング入力、音声入力、画面上に描かれた文字を認識する文字入力を、また出力手段として静止画とアニメーションの表示、録音音声の出力、規則合成の出力という、マルチモーダル入出力機能を備えている。Museでは、これらの機能すべてをGUIによって取り扱うことができる。

キーワード　ユーザインタフェース、マルチモーダルインタフェース、GUI、オブジェクト指向

The Development of Rapid Prototyping Tool "Muse" for Social Information Systems

Hiroyuki KAMIO, Mika AMAMIYA, Arisa UCHIYAMA,
Hiroshi MATSU'URA and Tsuneo NITTA
Multimedia Engineering Laboratory, Toshiba Corporation

Yanagi-cho 70, Saiwai-ku, Kawasaki-si, Kanagawa, 210 Japan.
044-548-5352
kamio@sp.mmlab.toshiba.co.jp

Abstract

This paper describes a Multimodal User-interface-design Support Editor (Muse) that reduces the designing and evaluating hours of multimodal user intarface. A developer can design a card's appearance by putting UI-objects on a screen, and construct scenarios by setting functions and/or messages and linking UI-objects to other UI-objects or screen. Muse equips multiple input channels of speech, pointing, and hand writing, as well as multiple output channels of picture, animation, audio, and text-to-speech. These multimodal functionalities can be implemented by using GUI on Muse.

key words　User Interface, Multimodal Interface, GUI, Object Oriented

1 まえがき

銀行の自動取引装置(ATM)、駅の自動券売機など、社会情報システムの高機能化が年々進行し、それに伴いユーザに要求される操作もますます複雑になってきている。このことから、より使いやすいユーザインタフェースの検討が盛んに行われている。従来のGUI(Graphical User Interface)に音声認識、音声合成などを取り入れたマルチモーダルユーザインタフェース(MUI)も、システムのユーザビリティを向上させるものとして期待されている。

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を開発し、地理案内[1]、音声券売機[2]など多様な分野への応用を試みた。このような不特定のユーザが利用するシステムの場合、実際にユーザにシステムを操作させて使い勝手を評価し、問題点を改良するという、評価・改良のサイクルを繰り返し行うことが、よりよいユーザインタフェースを構築する近道となる[3]。しかしMUIでは、対話のシナリオが複雑になるため、評価・改良のサイクルを数多く繰り返す必要があり、アプリケーション開発に要する期間が長期化する傾向にあった。

そこで、今回我々はMUIの試作・評価を迅速に行うラピッドプロトタイピングツールMuse(Multimodal User-interface-design Support Editor)を開発した。Museでは画面レイアウトや画面遷移による対話シナリオの設計、マルチモーダル入出力機能の付加などを全てGUIを用いて行うことができ、また設計したユーザインタフェースを実際に動作させて使い勝手の評価を行うことができる。

ラピッドプロトタイピングツールMuseを用いたMUI開発については既に報告しているが[4]、本稿ではMuseの構成、Museにおけるマルチモーダル入出力の取り扱い方法等について詳しく述べる。

2 Museの構成と開発過程

Museは、オブジェクト指向方法論OMT(Object Modeling Technique)[5]を適用して開発された。ソフトウェア開発におけるオブジェクト指向アプローチでは、ソフトウェアを実世界に対応したオブジェクトでモデル化し、そのモデルを利用して実装言語に独立に設計を行っていく手法が採られる。具体的には個々の機能モジュールをオブジェクトとして抽出し、オブジェクト相互のメッセージ交換によってMuse全体を構成する。OMT法はオブジェクト指向分析(OOA:Object Oriented Analysis)、オブジェクト指向設計(OOD:Object Oriented Design)を行う際に、モデルを効率よく記述する図式記法である。

OOAの初期段階では、Museを構成するオブジェクトの抽出が行われた。この結果、以下に示す2種類のデータと5つのマネジャが抽出され、各々独立したオブジェクトとして分析が行われた。それぞれのオブジェクトの役割を以下に示す。

1. データ
   - カード：ユーザインタフェースの画面を表すオブジェクト
   - UI-object：カード上に配置される部品オブジェクト
2. マネジャ
   - カードレイアウトマネジャ：カードの編集を行うマネジャ
   - マップビューア：作成したカード群の鳥瞰を行うマネジャ
   - パーツマネジャ：UI-objectを発給するマネジャ
   - ログマネジャ：作成したユーザインタフェースを動作させ、操作状況の記録を取るマネジャ
   - スタックマネジャ：作成したユーザインタフェースを管理保存するマネジャ

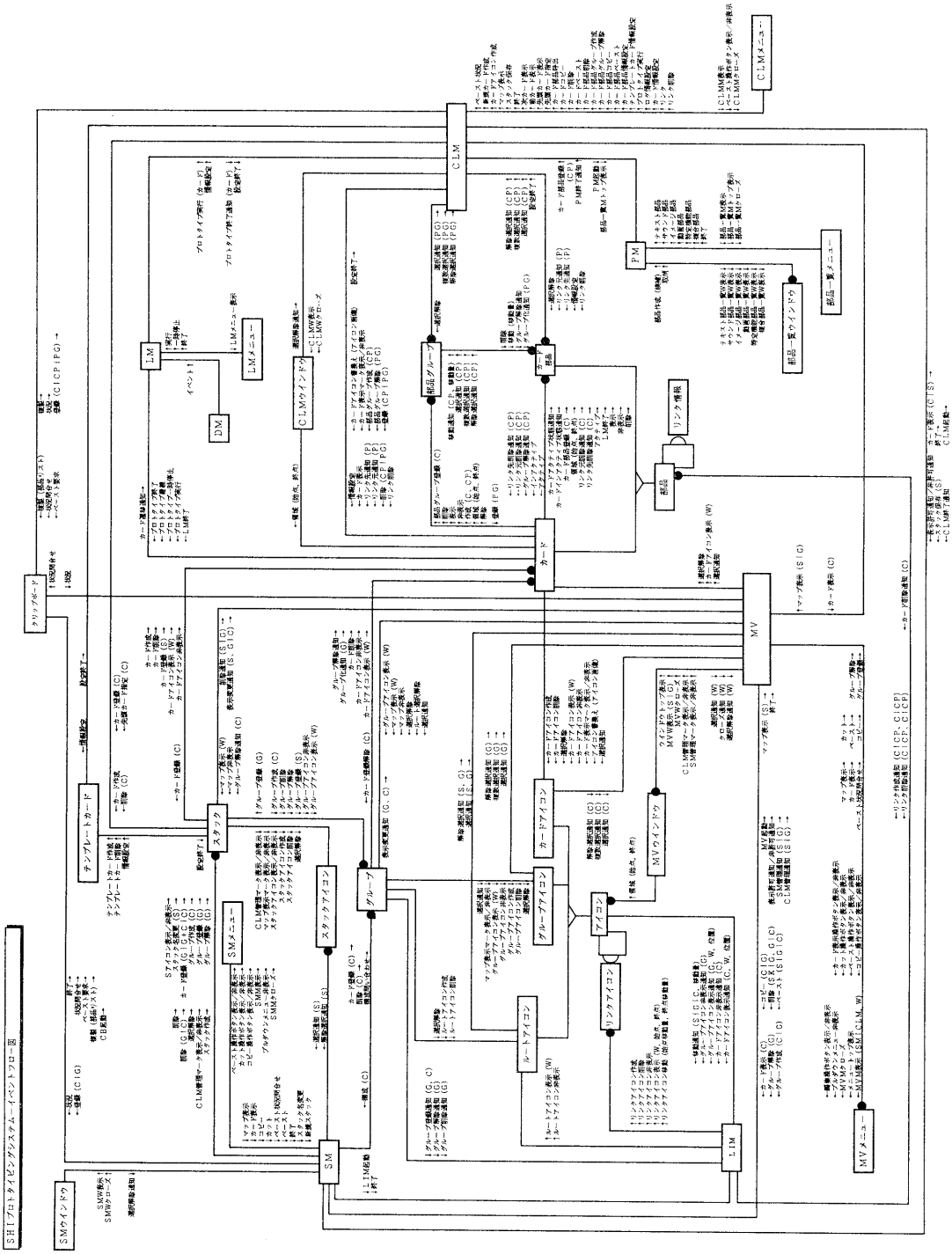

図 1: Muse の OOA におけるイベントフロー図

次にMuseの動作仕様を記述したシナリオが作成され、このシナリオを実現するためにオブジェクト間で交わされるメッセージの設定が行われた。また、シナリオ実現のため、新たなオブジェクトの導出も行われた。これらのオブジェクトとメッセージの流れをまとめたものが、図1に示すイベントフロー図である。イベントフロー図は、抽出されたオブジェクト相互の関連と、オブジェクト間に流れるメッセージを記述したものである。この図によって各オブジェクトの外部仕様を明確にすることができる。

OODでは、OOAで得られたオブジェクト構造を基に実装を考慮しながら詳細設計を行う。今回はMVCモデル[6]に基づき、画面上に表示されてユーザからの入力を直接受けるView、カードやUI-objectのデータ部分でありViewを所有するModel、Modelを管理するマネジャであるControlに分類し、それぞれに対応するオブジェクトを割り当てた。

図2は今回開発したMuseの全景図である。Museでは、カード上にテキスト、イメージなどのUI-objectを配置することでカードの外見を設計し、UI-objectから他のカードやUI-objectにリンクを設定することでユーザインタフェースの動作を設計する。UI-objectにはイメージ、テキスト、サウンド、音声合成、アニメーション表示という出力部品オブジェクトと、音声認識、文字認識という入力部品オブジェクトがある。また条件分岐やタイマなどの特殊な部品オブジェクトも用意されている。これらの部品オブジェクトは全て抽象クラスであるUI-objectを継承して作成されている。このためMuseではUI-objectの配置やリンク設定などを、UI-objectの種類を意識することなく共通に行うことができる。

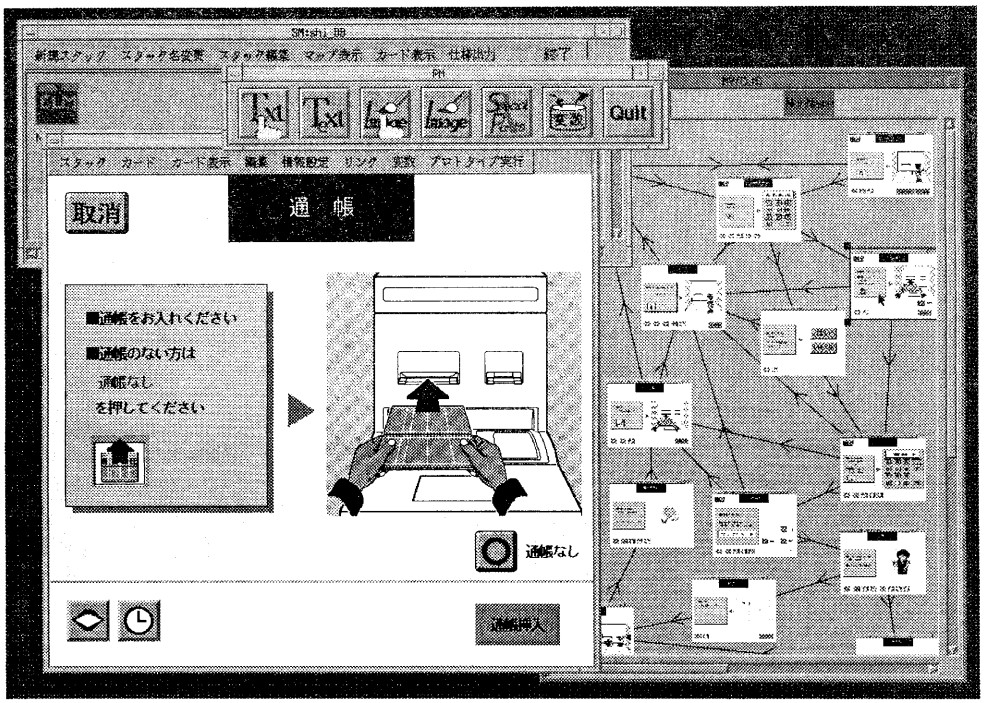

図2: ラピッドプロトタイピングツールMuseの全景

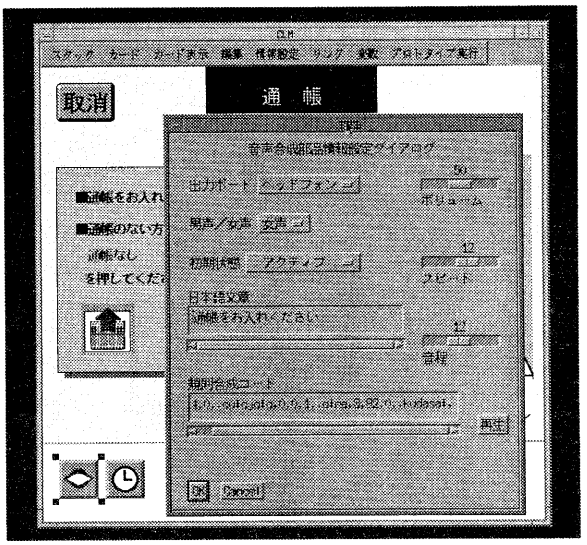

図 3: 音声合成部品と情報設定ダイアログ

3 マルチモーダル入出力の取り扱い

Muse の出力機能は、画面表示を行うテキスト部品とイメージ部品、音声出力を行うサウンド部品と規則合成部品から成る UI-object を組み合わせて実現される。出力機能の中で、音声出力を行う UI-object は本来実行画面上には表示されないものである。このため、カード上の実行画面下部にこれら UI-object のアイコンを配置する領域を設けた。これにより、リンクの設定や機能・情報設定ダイアログボックスの表示を他の UI-object 同様、カード上で行いながら設計することができる（図3参照）。

一方、Muse の入力機能は、マウスによる入力の他に、タッチパネルからの入力、音声入力、タッチパネル上に描かれた文字を認識する文字入力が用意されている。タッチパネル入力は、タッチパネルで検出される座標データを X-Window のマウスイベントに変換して Muse に送信している。音声入力機能は、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[7] を Muse に接続して実現している。KeySpot は認識単語をカナ文字で登録することで、任意の語いを認識でき、語いの変更などを行うラピッドプロトタイピングに適している。

文字認識機能では、指でタッチパネル上に描いた平仮名・片仮名・英数字を認識することができ、ソフトウェアで実装されている [8]。音声入力機能、文字認識機能は、それぞれ専用の音声認識部品、文字認識部品を音声出力部品と同様カード上に配置して、リンク設定やダイアログボックスによる機能・情報設定を行うことができる。

Muse では、これらの機能の他に変数機能を備えている。これは音声認識や文字認識の結果を格納したり、変数の内容によってリンク先を切替える条件分岐を行うものである。変数機能は、全ての UI-object から参照できる変数オブジェクト、変数オブジェクトに記憶されている値によってリンク先を変更する分岐オブジェクトから成り立っている。例えば、図 4のように Card1 において音声認識装置 KeySpot から結果が得られた時、Card1 上の音声認識部品は結果を変数オブジェクトに保存して、リンク先の分岐部品にメッセージ “Active” を送信する。分岐部品では変数オブジェクトに保存されている結果によってリンク先を切替え、メッセージを送信する。このように、条件分岐のような複雑な処理も Muse 上で記述できるようになっている。

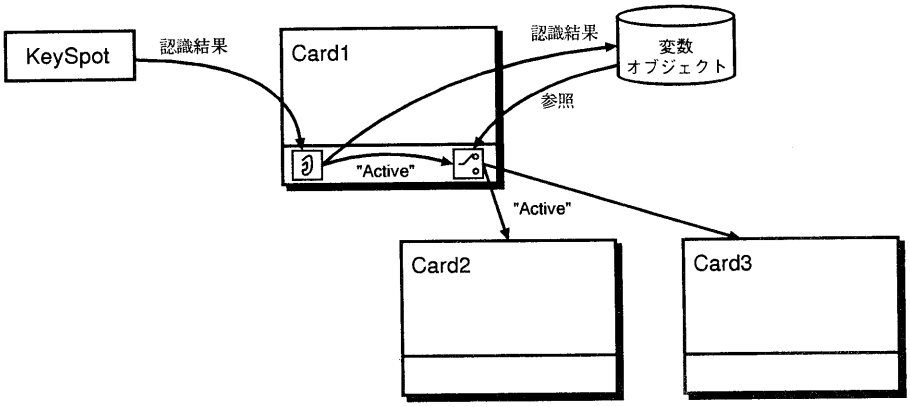

図 4: 変数オブジェクトと分岐部品による条件分岐

4 まとめ

社会情報システムのユーザインタフェース向上を目的としたラピッドプロトタイピングツール Muse を開発した。Muse は音声を含むマルチモーダル入出力を取り扱う UI-object をアイコンという形で可視化することにより、イメージやテキストと同様に画面上で取り扱うことが可能である。これによって、複雑になりがちなマルチモーダル対話システムのラピッドプロトタイピングを容易に行うことができた。

参考文献

[1] 神尾, 松浦, 正井, 新田: “マルチモーダル対話システム MultiksDial”, 信学論, D-II, Vol.J77-D-II, No.8, pp.1429-1437(1993-11).

[2] 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).

[3] 日本認知科学会編: “認知科学の発展 第 5 巻”, 講談社 (1992).

[4] 新田, 神尾, 雨宮, 内山, 田村: “マルチモーダル UI とラピッドプロトタイピング”, 情処研報, Vol.95, No.73, pp.29-34(1995-7).

[5] J. ランボー,M. プラハ,W. プレメラニ,F. エディ,W. ローレンセン:“オブジェクト指向方法論 OMT”, トッパン (1992).

[6] 青木淳: “例題による !! オブジェクト指向分析設計テクニック”, pp.65-103, SRC(1994).

[7] 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 ICSLP94, S23-9.1, pp.1383-1386(1994-9).

[8] 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).