

IntelligentPadの機構

7S-1

長崎 祥 今滝 隆元 田中 譲

北海道大学 工学部

1.はじめに

我々は、IntelligentPad^[1]の開発を行ってきた。Padは、全てのものを紙(pad)とみなし、それらを貼り合わせるにより、互いの機能を組み合わせ、より複雑なものを作っていくことができるシステムである。ここでは、Padの機構について報告する。

2.Padの機構

2.1 Padの内部機構

Padは、Smalltalk-80^[2]のウィンドウに利用されているMVCモデルの拡張である。その内部はModel,View,Controllerの3つのオブジェクトの組み合わせにより構成される。これら3つのオブジェクトの組み合わせを変えることにより、Padの表示形態(View)を変更したり、機能(Model)を変更した別のPadを作ることが可能である。これにより、少ない種類のクラス定義により様々なPadを作ることができる。この3つのオブジェクトの主な機能は、

- ・Model 他Padから送られてきたデータの処理を行う。
- ・View Pad間の通信を行う。Padの表示形態を決める。
- ・Controller マウス、キーボードからの入力に対する反応を決める。

に分けられる(図1)。機能分担に伴って、3つのオブジェクトの間でのメッセージのやり取りが必要となるが、Padではこれらのオブジェクトの間のメッセージ交換の構造を図2のように制限している。Controllerは直接Modelに対してメッ

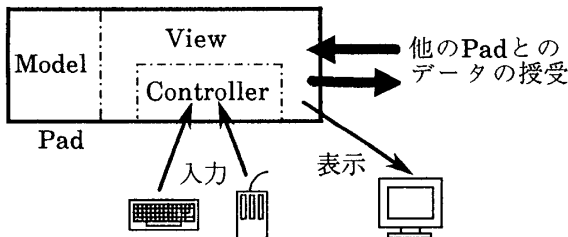


図1 Pad内部の機能分担

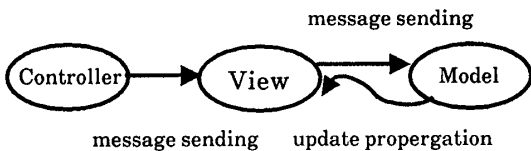


図2 Pad内部のメッセージの流れ

セージを送ることはなく、ViewからControllerへメッセージが送られることもない。特にModelとViewとの間でのメッセージは標準化された幾つかのメッセージに限定されている(表1)。これらの標準メッセージはすべてのPad用のModelとViewに共通なもので、ModelからViewへは、Modelの更新要求をViewへ知らせるメッセージのみ、ViewからModelへのメッセージも数種類である。これは、ModelとViewを組み合わせるPadを構成する場合、どのように組み合わせてもエラーが発生しないようにするためである。Controllerは種類が少なく、Viewとの係わりが深いため、あらかじめ定められた標準メッセージはない。

ModelからViewへ	ViewからModelへ
update: aSymbol	inputData: dataPair getData: dataName setData: dataName keyboardPressed: code buttonPressed: code 等

表1 ModelとViewの間のメッセージ

2.2 Pad間の機構

IntelligentPadでは、Pad同士を貼り合わせていくことにより、互いの機能を組み合わせ、より複雑な機能を持つPadを作り上げることができる。このような機能は、貼り合わせたPadの間(各機能間)のデータ授受によって実現されている。Padの間のデータ授受は図3にある3種類の標準メッセージによって行われる。Pad間のデータ

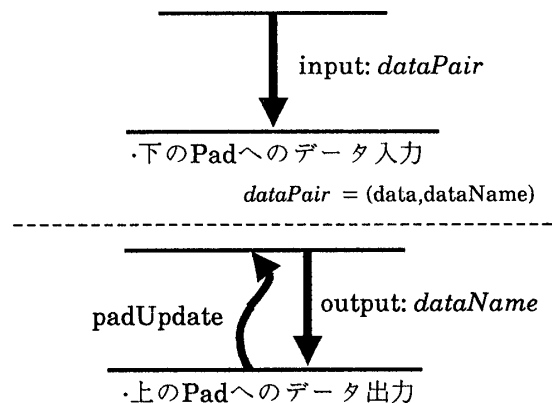


図3 Pad間のメッセージ

Mechanisms of IntelligentPad System

Akira NAGASAKI, Takamoto IMATAKI, Yuzuru TANAKA
Hokkaido Univ.

授受は、自分の下にあるPadに対してデータを送りたい場合と、自分の上にあるPadへデータを送りたい場合とに分けられる。前者をデータ入力、後者をデータ出力と呼ぶ。それぞれの場合について説明する。

・データ入力の場合

自分の下にあるPadに対してデータを出したい場合には、転送先のPadにinput: dataPairを送ることによりデータ転送を行う。dataPairは、Arrayであり、第一要素に送りたいデータ、第二要素に転送先Padの中でのデータの格納場所の名前が入っている。

・データ出力の場合

自分の上に貼られているPadにデータを送りたい場合には、自分の上に貼られているすべてのPadにpadUpdateを送り、状態の更新を要求する。padUpdateを受けたPadは、自分が更新要求をしてきたPadの出力装置となっていれば、そのPadにoutput: dataNameを送って対応するデータを読んでくる。dataNameは更新要求をしてきたPadが持っているいくつかのデータの内、自分が出力としてもらうべきデータの名前である。

このようにメッセージを制限するのは、どのようなPadを貼り合わせても、Pad間のデータの授受によるエラーが発生しないようにするためである。

2.3 Padを構成するクラス

IntelligentPadは、すべてのPadに共通な機能がいくつかある。例えば、自分の移動やコピー、Padを構成する3つのオブジェクトの間やPad間のデータの授受などである。このような共通機能は、新規にPadを定義する場合、新しく定義するModelやViewでも当然必要な機能である。新しいModelやViewを定義する度に、新規に追加される機能とともに共通機能も定義しなければならないとすると煩雑である。そのためIntelligentPadでは、Smalltalk-80の持つクラス間の機能継承機構を利用して、共通な機能は、全てPadの基本クラス(PadModel, PadView,

PadController)に定義されている。そのサブクラスとして様々な機能を持つModel, View, Controllerが定義されている。新たなPadを作りたい場合は、そのPadが特別な表示形態を必要とするならばViewを、新たな機能を必要とするならばModelを、新たな入力デバイス処理を必要とするならばControllerをそれぞれ基本クラスのサブクラスとして定義する。すべてのPadに共通な機能は、基本クラスにすべて定義されているので、それらのサブクラスとして作られるクラスは共通機能を継承する。そのためサブクラスでは、そのクラスで必要な部分のみを作ればよく、最低限のプログラミングで新しいPadのためのクラスを作ることが可能である。

3. まとめ

IntelligentPadの機構について述べた。IntelligentPadは、内部オブジェクト(MVC)間、Pad間のそれぞれでのメッセージの流れを制限することにより、メッセージの標準化が可能となった。これにより、Pad内部ではMVCの組み合わせが自由になり、様々なPadを容易に作る事が可能となった。Padの間ではPad同士の貼り合わせが自由になり、どのようなPadの組み合わせも可能となった。今後は、基本的な機能を持つPadの為のModelやViewを数多く定義していくとともに、作られたPadを管理、検索する機構についても考えていきたい。

【参考文献】

- [1] Y.TANAKA and T.IMATAKI: IntelligentPad: A HYPERMEDIA SYSTEM ALLOWING FUNCTIONAL COMPOSITIONS OF ACTIVE MEDIA OBJECTS THROUGH DIRECT MANIPURATIONS, Proc. of the IFIP 11th World Computer Congress, pp.541-546, San Francisco (1989)
- [2] Adele Goldberg and Davis Robson: "Smalltalk-80: The Language and its Implementation" (1983)

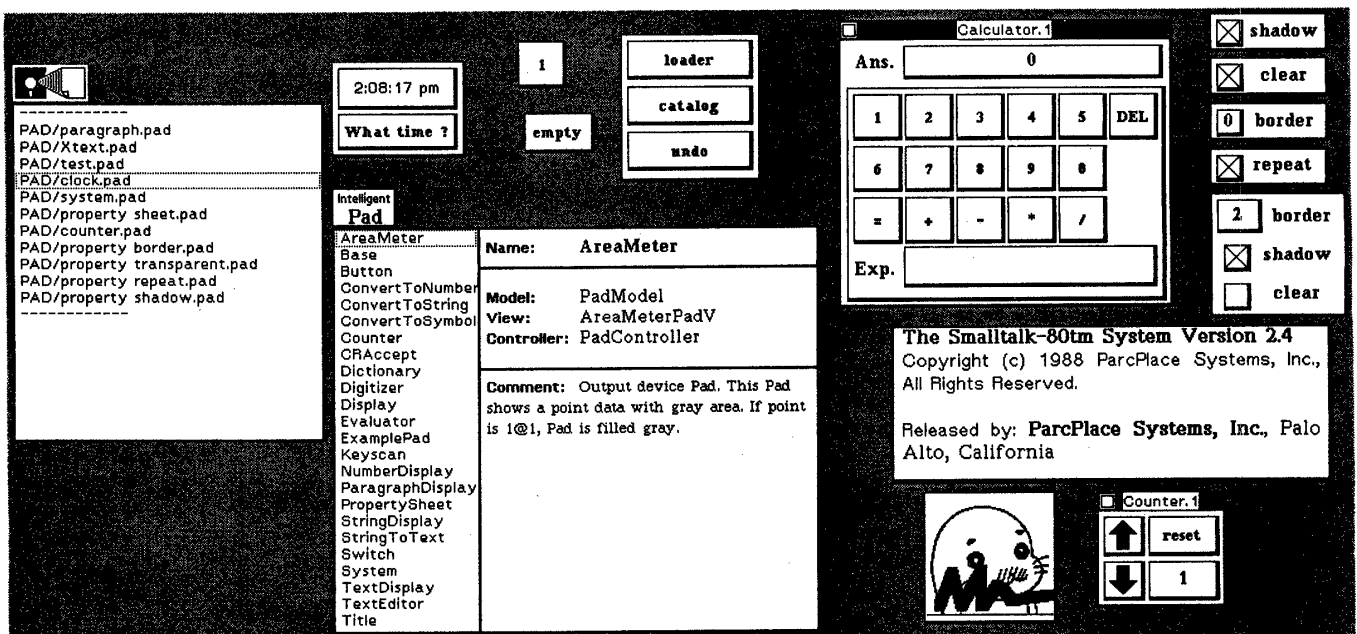


図4 実際のPadの例