

操作環境と機能の接続方式に関する考察

森 岳志†

2J-6

FRIEND21 研究センター

1 はじめに

計算機に保存された情報をユーザに効率良く伝えるために、情報の表示形態は重要な意味をもつ。多様で複雑に絡みあった情報を効果的に表示することは、使い易いインタフェースを提供するための要件の一つである。しかし、情報の最も効果的な表示形態は、ユーザの視点や好み、利用目的、作業経過等に依存する。そのため、一つの情報に対し、ソフトウェア開発者が全てのユーザに最適な単一の表示形態を提供することは困難である。

本稿では、一つの情報を様々な視点から表示可能で、ユーザが表示形態を動的に選択でき、新しい種類の情報を慣れ親しんだ表示形態で表示したり、既存の情報を新しい種類の表示形態によって表示できる環境をユーザに提供するための一構成方法を提案する。以下では情報ソースを単に AP、表示形態を単に Viewer と略す。

2 システム構成法

2.1 Viewpoint

本研究は、アプリケーション固有部と HI 処理部を分離した Smalltalk-80 の MVC(Model-View-Controller) 等の枠組 [1][2][3] に深く関連する。MVC では、Model を様々な視点から表示できるマルチビュー機構を提供している。しかし、Model のどの部分のどのような特性を抽出して視覚化するのは Model 内に存在する手続きが決定している(本稿ではこの手続きを観点情報と呼ぶ)。そのため、全く新しい種類の View によって既存の Model を視覚化しようと考えた時、図 2.1 のように既存の Model に変更を加える必要がある。

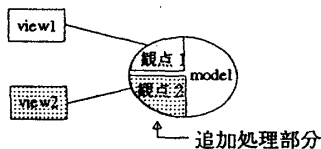


図 2.1: マルチビュー

観点情報を Model から独立させることができれば、Model に変更を加えないで新規 View との接続が可能となると考え、この観点情報を保持するオブジェクトを図 2.2 のように Viewpoint(VP) として独立させる。

An Approach for Mapping of Multiple Appearances and Functions

Takeshi Mori

FRIEND21 RESEARCH CENTER

†現日本電気(株)

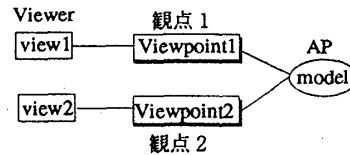


図 2.2: Viewpoint

2.2 Viewpoint によるシステム構成法

本研究では第1章で述べた要求を満足させるため、図 2.3 のシステム構成を採用する。

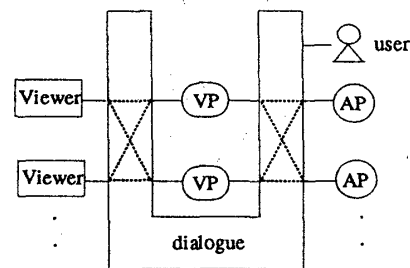


図 2.3: システム構成

ユーザ、Viewer、VP、AP が Dialogue を経由してコミュニケーションを図る構造を採用する。Dialogue は、コミュニケーションの場の役割と、オブジェクトの管理を行なう統括的なオブジェクトとして位置付けられる。

本稿では、ユーザが AP、VP、Viewer を指定し、接続操作を行なうことによって AP が視覚化されることを仮定する。図 2.3 のような接続関係を可能にすることによって、より自由度の高い情報の表示環境をユーザに提供することが可能となる。

(1) データ交換

AP、Viewer、VP の各オブジェクトの独立性を保持し、プラグブルな環境を提供するためには、オブジェクト間のコミュニケーション手法が問題となる。本稿では研究の第一歩として、AP を属性付けられたタプルの集合で記述可能な領域に限定する。同様に、各オブジェクト間のデータ交換(中間表現)にもタプルの集合を用いる。

(2) VP の役割

AP、Viewer、VP の独立性と再利用性を高めるために、AP と Viewer から一時的な局面情報や接続情報を排除する。VP は、(a) VP の仕様に従った特性を AP から抽出し、(b) 接続に必要な一時的変換情報と、AP と Viewer の要素間の対応関係を保持する。

3 具体例

3.1 概要

第2章で述べたシステム構成法に従い、簡単な実験システムを作成した。本実験システムでは、[AP] 電子メール AP, ファイル AP, 基本テキスト AP、[VP] $1 \times n$ の数値要素を生成する vp_num, $n \times n$ のテキスト要素を生成する vp_table, 1×1 のテキスト要素を生成する vp_text、[Viewer] 棒グラフ Viewer, カレンダー Viewer, 表 Viewer, テキスト Viewer を提供している。

実験システムの初期画面(図 3.1 参照)は、システム中の AP, VP Class, Viewer を list 形式で表示する。ユーザは、Plug-in ボタンを押すことにより、オブジェクトを動的に Plug-in することが可能である。ユーザが AP と Viewer と VP を list から選択し、Attach ボタンを押すことによって AP と Viewer の接続を行う。

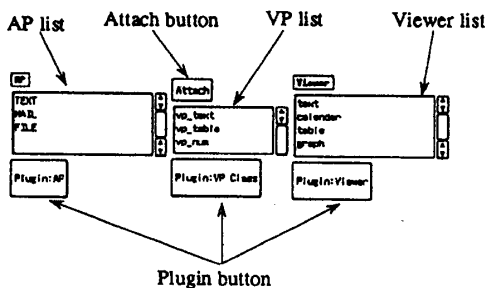


図 3.1: 実験システムの初期画面

3.2 各オブジェクトの詳細

Dialogue システム全体の管理・支援を行なうオブジェクトである。本オブジェクトは、初期画面の生成と表示、オブジェクトとリソースの管理、ユーザへのメッセージ出力と入力の受け付け、VP の生成、オブジェクト内の各種 Method の dispatch、Plug-in のための処理を行なう。

AP(情報ソースオブジェクト) ファイルシステムから必要な情報を抽出した、仮想的な表形式のオブジェクトである。AP の開発者は、キー項目に基づいて整理した表、各カラムに対応するファイルシステム上での位置、VP との接続時に呼び出される Attach Method を記述する。

VP(Viewpoint) VP Class と VP の構成を図 4.1 に示す。VP の変換情報は AP の部分情報を抽出するために必要となる情報で、接続に先立ち、ユーザから取得する情報である。対応表は、AP の一要素と中間表現の一要素との対応情報である。

Viewer(表示形態オブジェクト) Viewer の構成を図 4.2 に示す。対応表は、中間表現の一要素と表示のための一要素との対応情報である。calendar と graph Viewer の画面例を図 4.3 に示す。

4 おわりに

ユーザの好みや作業目的に従い一つの情報を様々な視点から表示可能で、Plug-in によって拡張可能な環境を構築することを目的としたシステムのオブジェクト構成法について提案し、簡単な実験システムを作成した。ユーザのカスタマイズ環境の整備と AP のデータモデルに関する検討が今後の課題である。

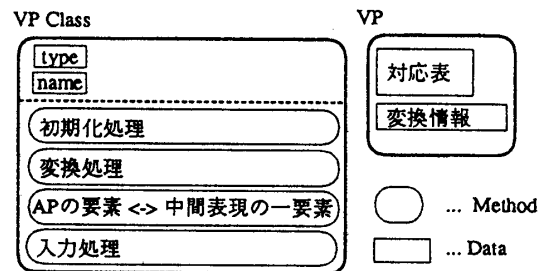


図 4.1: VP Class と VP オブジェクト

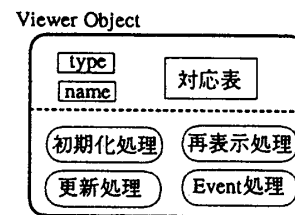


図 4.2: Viewer オブジェクト

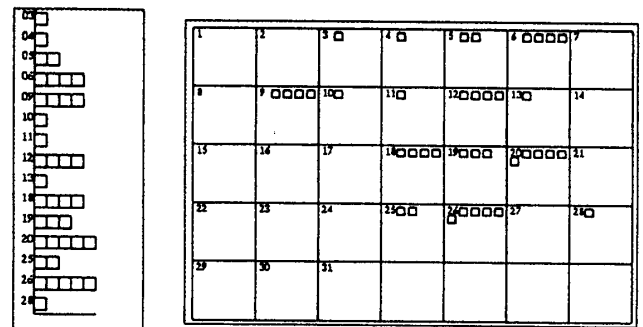


図 4.3: graph と calendar Viewer

参考文献

- [1] Adele Goldberg. Smalltalk-80: 対話的プログラミング環境. オーム社, 1986.
- [2] J.Coutaz. Pac, an object oriented model for dialog design. In INTERACT'87, pp. 431-436, 1987.
- [3] Ralph D.Hill. The abstraction-link-view paradigm. In CHI'92, pp. 335-342, 1992.