

木本雅彦<sup>2</sup>

東京工業大学大学院 情報理工学研究所

## 1 はじめに

X-Window System において、ウインドウマネージャは見た目と操作性を決定する大きな要素である。多くのウインドウマネージャは、利用者が設定情報を変更し、各自の趣向に合わせた操作性にできることを特徴としている。しかし異なった操作性の乱立が、初心者間の情報共有を難しくするという問題もある。本稿ではウインドウマネージャに必要な機能について考察し、これに基づいて実装しているウインドウマネージャである prawn について述べる。

## 2 ウインドウマネージャの役割に関する考察

本節ではウインドウマネージャが果たすべきの役割について考察する。X-Window System では、ウインドウマネージャはウインドウの外枠などの外観のみならず、操作性全般に広く影響を及ぼす。昨今の PC-UNIX の普及で、UNIX および X-Window System の利用者が増加したこともあり、さまざまなウインドウマネージャが開発されている。

多くのウインドウマネージャは、その挙動を利用者によって変更(カスタマイズ)できる機能を有する。一部のウインドウマネージャでは、メニュー項目やマウス反応速度などの操作性だけでなく、ウインドウ枠の色や形状、ボタンの形までカスタマイズできるものがある。カスタマイズ機能によって、利用者は自分の趣向にあわせた環境を構築できる。

反面カスタマイズ機能には問題点もある。まず、変更できる項目が多すぎる点である。カスタマイズは、本来システムを設計する際に、利用者の趣向や身体的特徴など個人ごとの差異に対応させるためのものである。例えば目が悪い人のためにカーソルを大きくしたり、マウスの操作が不自由な人のためにボタンを大きくし、ボタンの付近にマウスが近寄ったら自動的に枠内にカーソルを移動させる、といったものである。理想的にはカスタマイズを一切行わなくてもすべての利用者がにと

て使いやすいことが望ましいが、現実には難しい。概してカスタマイズできる項目を増やし、使い勝手は利用者の手による設定に依存しているのが主流である。この結果として、利用者ごとにまったく異なる環境ができあがるという問題がある。

この問題が顕現するのは、操作方法を他人に説明する場合である。特に他者から説明をうけることが多い初心者に対しては、むしろ変更できる項目を制限した方がよい。「説明できること」はこのときの一つの基準になると著者は考える。例えばボタンの配置や大きさの変更を許す場合でも、順番や比率を変更不可にしておけば「~の隣にあるボタン」「~よりも大きなボタン」といった説明が可能になる。

このような利用者ごとに変更を許さない項目や、制限をつけて変更を許す項目の違いをウインドウマネージャのプログラム中に埋め込んで汎用性に欠ける。利用者が変更可能か否かを記述するための設定を記述できる必要がある。

次にカスタマイズする内容の記述形式に言及する。多くのウインドウマネージャでは独自の記述形式を導入している。この形式は、変数の値を指定するか機能の有効/無効を指定するといった簡単な記述しかできない。前述のように、初心者への説明しやすさを考えると、ウインドウの出現位置のように、随時計算する項目の計算アルゴリズムなども変更できたほうが都合がよい。反面複雑な記述形式を導入すると、これを学習するために利用者に新たな負担を強いることになるが、汎用的な言語を用いることで設定情報の記述とアプリケーション開発に用いる言語を統合できる。このように、アプリケーションを記述する言語と、カスタマイズ情報を記述する言語を統合しようという試みは tcl[3] や guile[4] で行なわれている。

## 3 prawn の設計

以上の考察に基づいて著者が開発しているウインドウマネージャが prawn(PICKLES Practical Window Manager)である。

prawn は、著者らがこれまで進めてきた PICKLES プロジェクト [1][2] の一環として開発された。PICKLES

<sup>1</sup> A study of design of window managers for beginners.

<sup>2</sup> Masahiko KIMOTO, Tokyo Institute of Technology.

プロジェクトは、街頭などに誰でも使える情報キオスクを設置することで、必要なときに、必要な場所でインターネットのサービスを利用できる環境を実現しようというものである。PICKLESプロジェクトで開発した情報キオスクのためのOSパッケージを、PICKLES SYSTEMと呼んでいる。PICKLES SYSTEMは現在BSD/OS 3.1を基に改良したものを採用している。

実験としてPICKLES SYSTEMでは、ICカード、PCカード、PDAなどのデバイスを利用者認証と個人情報の蓄積に用いている。prawmにはこのような認証用デバイスの挿抜を監視し、利用者の認証装置が取り外されたら端末の利用を終了させる機能を有する。

カスタマイズ用の設定ファイルとして、端末ごとの設定ファイルと利用者ごとの設定ファイルの2種類を用意する。端末ごとの設定ファイルが最初に参照される。この中で利用者によって変更が許可された項目に関しては、利用者の設定ファイルの内容が反映される。また、カスタマイズ用設定ファイルの記述形式としては、アプリケーション開発にも利用できる汎用言語を採用する。

## 4 prawmの実装と今後の課題

prawmは現在fvwm 2.0.46[5]を基にして実装をすすめている。図1は開発中のparwmの実行画面である。

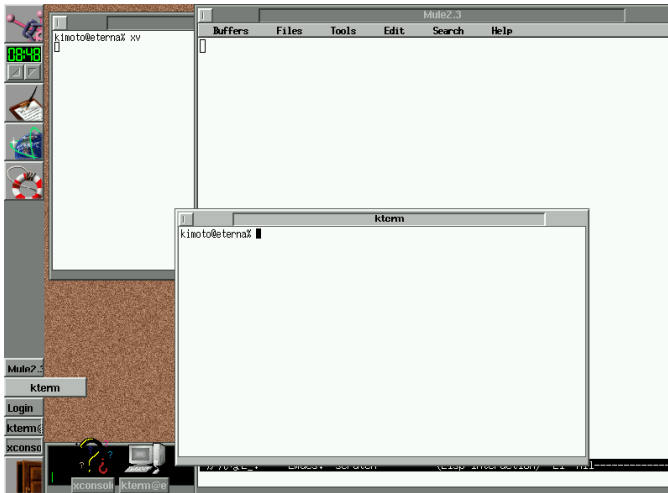


図1: prawm 実行画面

ウィンドウマネージャの挙動を記述するための言語処理系として、ruby[6]を採用した。rubyは手軽なオブジェクト指向プログラミングを目指したスクリプト言語処理系である。prawmはrubyのインタプリタを内蔵

し、rubyの文法に乗っ取って記述された設定ファイルを解釈する。

またfvwmと同様の外部モジュールインタフェースを採用し、fvwmと通信するためのrubyのクラスライブラリを作成した。fvwmの外部モジュールは、ウィンドウマネージャの機能拡張を、外部プログラムとして実現するものである。これによって、設定ファイルに記述するには規模が大きすぎる機能拡張も、外部モジュールとしてrubyで記述できる。rubyにはGUI開発用のライブラリが提供されているため、利用者はrubyの文法を学習するだけで、ウィンドウマネージャの設定と外部モジュールの開発、GUIを含めたプログラムの開発ができるようになる。

prawmは現在開発途中であり、機能強化と実行速度の向上などが今後の課題である。また、prawmを中心としたデスクトップ統合環境の開発も行っていく予定である。この統合環境は前述のPICKLES SYSTEMの標準環境として採用する予定である。

## 5 おわりに

ウィンドウマネージャに必要とされる機能について、特にカスタマイズ機能に注目し考察した。現在の多くのウィンドウマネージャが見た目や豊富なカスタマイズ機能の特徴としているが、初心者向けを念頭に置くとむしろ変更できる機能を制限したほうが良い局面もある。著者が開発中のprawmは、高度な設定の記述が可能であるとともに、設定ファイルの優先度を導入し状況によって設定項目に制限を加えることができる。

## 参考文献

- [1] 木本雅彦 大野浩之, 街角公衆情報端末計画 ~ PICKLESの概要~, Mar. 1996, 第52回全国大会講演番号3Y-2
- [2] 木本雅彦, 大野浩之, 自律型ネットワーク端末 (PICKLES) を用いたシステム運用技法, Feb. 1998, 情報処理学会, DSM シンポジウム
- [3] Tcl/Tk Information, <http://www.tcltk.com>
- [4] Guile Pages, <http://www.gnu.org/software/guile/guile.html>
- [5] The Official FVWM Homepage, <http://www.hpc.uh.edu/fvwm/>
- [6] Ruby Home Page, <http://www.netlab.co.jp>