

YyonX 実行モデル (1) — 機能分散に基づく設計 —

7 E - 1

井田昌之^{†1}, 古坂孝史^{†2}, 田中啓介^{†1}
 青山学院大学情報科学研究センター 研究教育開発室

1 始めに

YyonX は、Common Lisp 用ポータブルウィンドウツールキット Yy の X-window 上の実現である。その概要は [1],[2] などに発表した。YyonX は、Yy-Protocol により会話する二つのプロセス Yy-server, Yy-client により実現されている。ここでは、このサーバ/クライアントによる実現とそれの上での実行モデル、機能分散について述べる。

2 サーバ/クライアント方式による実現

一般に Lisp 用のツールキットは、ウィンドウシステムそれ自身と同様に、1. カーネル方式: OS 内に基本機能を持つ; Lisp マシンなど、2. ライブラリ方式: 応用プログラムにライブラリとして必要機能が付く; CLIM, Common Windows をはじめ多くのシステム、のいずれかが用いられている。

ライブラリ方式の場合、応用プログラムが実行される機械の CPU 及びメモリに対して更に負荷をかけることになる。ユーザインタフェースアクセサリの実装は、それが増えるにつれ利用者にとっては楽になる一方、プログラムは重くなる。

YyonX は、こうした点を解決できるように、サーバ/クライアント方式により設計した。すなわち、応用プログラムをクライアントとし、基本的なウィンドウ制御機構をサーバとし、これらを分離する。そして、この間のプロトコルを定義する。これにより、異なるベンダによるサーバ、クライアント間での共同動作も可能となる。

図 1 に三台のワークステーション (WS) による共同動作をしめす。YyonX ではこのように、応用が実行される WS、ウィンドウ上のユーザインタフェースが実行される WS、X-server が実行される WS を各々別にする事ができる。これにより、CPU およびメモリの負荷を分散させることができた。

3 Yy プロトコルによる機能分散

Yy プロトコルは、四バイトを一単位とし、それにウィンドウシステム/ユーザインタフェースに必要な指示をコンパクトな形でまとめたプロトコルである。現在の version 1.1 では、48 種ある。これらは、次の三つに大別できる。

- A) コマンド: 同期を要しないサーバへの指令。
- B) 命令: 同期を要し、戻り値を待つ指令。
- C) 通知: サーバからのイベント送信。

この Yy プロトコルの転送速度は一回の転送量によっても異なるが、SUN-4/280 において 0.55msec/byte (40 バイトパケット 100 回連続送信の平均値) である。更に、意味的に高度な機能をコンパクトに転送できる。また、

Design of Two Execution models for YyonX, by Masayuki IDA^{†1}, Takashi KOSAKA^{†2}, Keisuke TANAKA^{†1}

^{†1}Aoyama Gakuin Univ. ^{†2}Aoyama Gakuin Univ./CSK Corp.

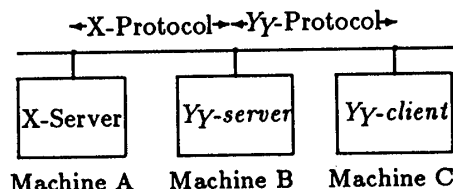


図 1: 三台に分散した Yy 実行環境例

上述のコマンドに対しては、間欠同期 (intermittent synchronization) と呼ぶ機構を持たせ、必要に応じてサーバ/クライアント間の処理の同期をとることが出来る。

4 サーバ/クライアント間の機能分担の課題

「プロトコルの性能」と「Lisp としての特徴を生かす機能分担法」という二つの課題がある。前者は更に次の項目に分けられる。

- (a) Yy-Protocol のフォーマットの最適性
- (b) Yy-Protocol の送受信速度/オーバーヘッド
- (c) X-server と Yy-server との間の送受信速度/オーバーヘッド

全ての入出力及びウィンドウまわりの処理をクライアントで行なう方式が、Lisp としての処理をもっとも柔軟に行なえる。しかし、入力処理の例として、一字一字を完全に同期をとりながら処理することを考えてみると、すべてクライアントで処理する方法では 103msec/byte の平均速度、ほとんど全ての処理をサーバだけで行なう方法 (クライアントには何もウィンドウの状況がわからなくなるのでほとんど意味がない) では、7msec/byte (+ 5msec per 20 pixel actual scroll) の平均速度となる。103msec/byte では、実用的には遅い。実現すべき解はこの中間にある。

これらの問題を解決するには、一般論をたてるのは困難なので、応用のパターン、これを実行モデルと呼んでいる、を設定し、それに対する機能を提供する一方、その実行の動態を指標にしながら全体の性能を評価し、改良を行なうことを考えた。

5 実行モデル

YyonX では、機能として page と viewport とが用意されている。これに対して、それぞれの特性を生かすような実行モデルの設計を行なった。

5.1 Page 実行モデル

次のようなモデルを page に持たせる。

- 「縦方向」に拡張できる巻紙状のウィンドウ。
- インタクションを主体としたウィンドウである。マウス、キーボード割り込みなどの中断機構も含めて入力に対する素早い応答を第一義とする。

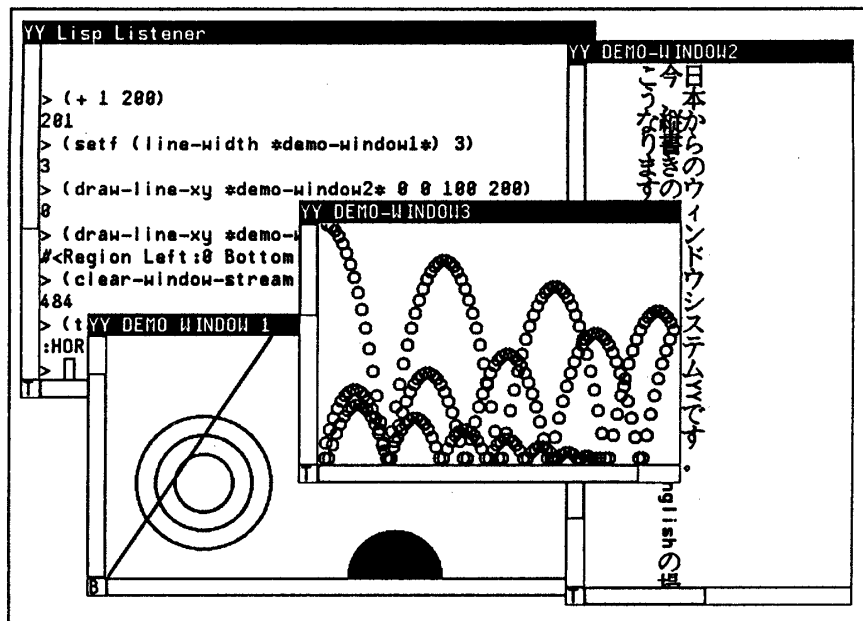


図 2: Yy ウィンドウ表示例

● テキスト出力には方向がある。それに対して最適化した機能を持つ。方向としては、縦及び横とする。縦／横の方向は、選択できる。(縦書きは可、斜め書きは無し。)

- カーソル概念が付随する。
- 画面制御がクライアントからなされることはある。

5.2 Viewport 実行モデル

次のようなモデルを viewport に持たせる。

- 主に Drawing Canvas を想定した二次元の平面である。
- テキスト出力より、高速多量描画に重点をおく。したがって、入力との同期という点では、多少ずれがあっても良い。
- 出力機能の豊富さに重点をおく。テキスト出力では、縦横だけでなく、斜め書き機能も用意する。
- データ入力、マウス指示などは、入力ストリームへのキューで良い。

5.3 各実行モデルの指標

次のような指標を想定して、実行性能のチューニングを行なう。

1. Page については、入力に対する高速応答を指標とし、毎秒4ストロークのキー入力(48語/分)、いかえれば、250msec/字、に対して十分速く応答する時間として、各字のキー入力後15msec以内にその字の表示が完了するものとする。
2. Viewport については、高速多量出力を指標とし、1msec/byteの連続処理能力を持ち、かつ、実際に1000字の表示が1秒以内でできるものとする。

これらの数値は、利用する機械の性能によって達成の困難さが異なるが、手元で利用している SUN-4/110,280 などでの上で計測することとした。

四つのウィンドウを表示している。Lisp リスナ及び縦書きウィンドウは Page モデルで、グラフィック表示は Viewport モデルで各々実行されている。Page では、カーソル表示、入力エコーなどがサーバ側で行なわれる。これにより、クライアントとのトラフィックが減る。Viewport では、表示領域に合わせてワールドが自動的に広げられる。各ウィンドウの左下の T および B のマークは、座標原点がそれぞれ、左上 (T)、左下 (B) にあることを表す、YyonX version 1.0 でのマークである。

5.4 各実行モデルでの検討項目

以下の検討項目に対して解を出して、現在の version 1.0 から、次の version 2.0 への改良点を明確にする。

- 1) スクロール、入力のエコー、カーソルの表示などはサーバ側でできるかどうか、あるいはやることが望ましいのかを検討する。
- 2) もし、ネックがあるとしたらそれは、Yy プロトコルそのものなのか、処理系作成技術なのか、機能分散の設計なのかを明らかにしたい。
- 3) 日本語処理機能を持った Common Lisp 処理系を前提とした時、どのような機能がサーバにあれば良いか?

6 結論

文献 [3] より、Page モデルでの高速な応答のためにはサーバ側でのカーソル表示、入力エコー処理等が必要などがわかった。また、Yy プロトコルの改良点も発見した。文献 [4] より、入力動作に対してはどちらの場合にも協調動作が必要などがわかった。これらの結果、指標データを満足することができ、X 上のポータブルウィンドウツールキットの実現手段としてサーバ/クライアントモデルは有効であることも示した。図 2 に例を示す。

謝辞：ポータブルウィンドウ研究会の諸氏との討論は大変有益であった。

参考文献

- [1] 井田昌之、他：YyonX, 情報処理学会第 40 回全国大会, March 1990. 3J-2, 3J-3, 3J-4
- [2] Masayuki Ida, et.al. : An Overview of Yy and YyonX - A CLOS based Window Tool Kit and its Implementation -, Proc. Europol'90, pp245-252 March 1990.
- [3] 古坂孝史、他：YyonX 実行モデル (2)、第 41 回全国大会, Sept. 1990.
- [4] 田中啓介、他：YyonX 実行モデル (3)、同上