

Common Lisp で実現した X-window 支援ツールキット

2K-6

西村 仁志* 平井 一人* 上田 良寛**

*日本電子計算(株) 大阪支店

**ATR 自動翻訳電話研究所

1. はじめに

現在、X-window を利用するためのプログラムインターフェースがいろいろ開発されているが、C 言語インターフェースのものが多い。Lisp で扱えるもので手軽に使えると言えば、MIT で開発された `clx` (Common Lisp X-window) がある。しかし `clx` は `xlib` と内容的に同等であるため、ベーシックなインターフェースしか持っていない。そのため、ポップアップメニューを表示するだけでも大変な労力を要する。そこで、そのような労力を払わなくても Common Lisp から X-window が利用できる様に、オブジェクト指向的な機能を持つツールキットを作成した。

本稿では、はじめにツールキットの概要について説明する。続いて、本ツールキットの中核を構成するプレゼンテーションシステムについて説明し、その利用例を示している。

2. ツールキットの概要

2.1 ツールキットの位置

図-1 に本ツールキットの位置づけを示す。

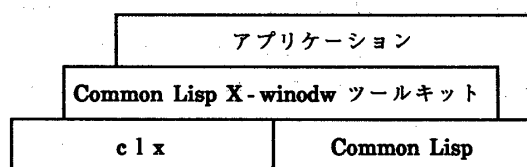


図-1

[注] 本ツールキットの開発は、SUN-4 を利用している。

2.2 開発の目的とツールキットの特徴

Common Lisp から X-window を利用するためのツールキットとして我が国では YYouX が開発されている。YYouX は Common Lisp 処理系での汎用的なウィンドウツールキットの性格であり、CLOS 処理系(当面は PCL)で稼働する。それに対し、本ツールキットは `clx` を利用するための手軽なツールキットを目的として開発した。

本ツールキットを利用すれば、タイトル行、メッセージ行やポップアップメニュー行を持つウィンドウを容易に作るができる。特に、プレゼンテーションタイプという考え方を利用することによりアプリケーションプログラムの開発は容易になる。

本ツールキットの機能には次の5種類がある。

- ① ウィンドウ作成関係
ウィンドウやポップアップメニューなどを表示する。
- ② ウィンドウ操作関係
ウィンドウを削除したり、大きさを変えたりする。
- ③ ウィンドウへの表示関係
ウィンドウにテキストや図形を表示する。
- ④ プレゼンテーションシステム関係
プレゼンテーションを表示したり入力したりする。
- ⑤ プロセス関係
プロセスを作ったり消したりする。

3. プレゼンテーションシステム*とその機能

3.1 プレゼンテーションタイプ

プレゼンテーションシステムにはプレゼンテーションタイプという概念がある。プレゼンテーションタイプとは、画面に表示されているオブジェクトに付けるタイプのことである。このタイプの持つ機能には次の2種類がある。

- 1) プレゼンテーションタイプとして表示されているものを、タイプを指定してそのオブジェクトが持っているデータを取得することができる。
- 2) マウスにセンシティブかどうかや、マウスのジェスチャー(マウスの左ボタンをクリックするなど)に対するアクション(マウスをクリックしたときに実行する処理)を定義できる。つまり、画面に表示されているオブジェクトをマウスでクリックすれば、そのオブジェクトが持っているデータと定義したアクションに基づいてアクションを起こすことができる。

*注: プレゼンテーションシステムは Symbolics Common Lisp でサポートされている機能である。本ツールは、それを X-window 環境でサポートしたものである。

Support Toolkit of X window System implemented by Common Lisp

Hitoshi NISHIMURA¹, Kazuto HIRAI¹, Yoshihiro UEDA²

1. Japan Information Processing Service Co., Ltd. , 2. ATR Interpreting Telephony Research Laboratories

3.2 プレゼンテーションアクションを実現するためのプロセス

プレゼンテーションのアクションを実現するために、マウスの動作を監視しているプロセスがある。

このプロセスは、マウスをオブジェクトの上に持っていたとき、オブジェクトの回りを罫線で囲むようにしたり、オブジェクトの上でマウスをクリックしたときオブジェクトが持っているアクションに起動をかけるようにしたりする。

3.3 プレゼンテーションアクションの考え方を取り入れたメリット

通常 `clx` を使ってアプリケーションの中で「マウスをクリックしてそのアクションとしてプログラムを動かす」プロセスを実現するとき、`xlib` を直接使ってプログラムを書くのと同様になる。そのためにプログラムのロジックとしてイベントを持つ必要があり、プログラム構造が複雑になる(図-2を参照)。

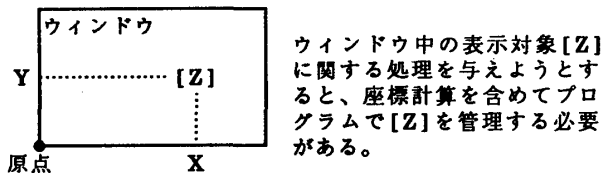


図-2 アプリケーションで直接 `clx` を使うとき

それに対して本ツールキットでは、プレゼンテーションアクションの考え方を利用しているので、アプリケーションの中でウィンドウ中の表示対象のメソッドを定義すればよい。これにより直接 `clx` を使う場合に比べてプログラムのロジックが簡単になり開発量も少なくて済む。

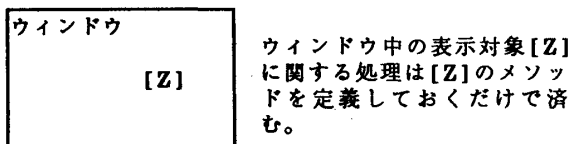


図-3 アプリケーションから本ツールキットを介して `clx` を使うとき

4. プレゼンテーションシステムに関連する関数の使用例

- ① プレゼンテーションタイプを定義する


```
(xl%defclass :edit-item)
```
- ② プレゼンテーションアクションを定義する


```
(xl%define-presentation-action
edit-item
:edit-item
:left-button-down
nil
(window self)
(print window)
(print self))
```
- ③ プレゼンテーションを表示する


```
(xl%present-object
:present "label"
:class :edit-item
:contents 'abc
:stream window)
```
- ④ プレゼンテーションを入力する


```
(xl%accept-object
:edit-item)
```

5. おわりに

Common Lisp で作成した X-window を支援するツールキットの特徴と機能について述べた。現在の版ではプレゼンテーションとして表示できるのはテキストだけである。そこで、今後は図形(直線、円など)も扱えるようにすることを検討している。

謝辞

本ツールキットの開発にあたり、いろいろ御教示いただいた ATR 自動翻訳電話研究所の言語処理研究室およびデータ処理研究室の皆さんに感謝いたします。

参考文献

- [1] 井田昌之、他：YYonX：概要設計、情報処理学会第40回全国大会、1990。
- [2] Oliver Jones：X window ハンドブック [西村 享(監修)], アスキー, 1990。