

# オブジェクトの堆積・連結モデルに基づく ウィンドウ・システム

2K-8

新城 靖、武川 努、翁長 健治  
(琉球大学)

## 1 はじめに

オブジェクトの堆積 (object-stacking) とは、object-basedシステムを構造化するためのモデルである [1]。その特徴は、一様なインタフェースを持つオブジェクトを結合することで、それらが持つ機能を統合して利用する点にある。そのモデルの有効性は、すでに分散型オペレーティング・システムにおける高機能ファイル・システムを構築する場面において確認している [2]。この論文では、オブジェクトの堆積の考え方をウィンドウ・システムに適用する方法について述べる。

ウィンドウ・システムは、高度な利用者インタフェースを実現する上で不可欠なものである。本研究の目的は、オブジェクトの堆積の考え方をウィンドウ・システムに適用することで、その機能を効率的に高度化することである。さらに、結果としてウィンドウ・システム全体の実現を容易にする。

この論文では、ウィンドウ・アプリケーションとウィンドウ・カーネルの間でプロセス間通信を行うようなウィンドウ・システムを考える。例として、Xウィンドウ・システムがあげられる。ウィンドウ・システムにおけるプロセス間通信の特徴として、ウィンドウ・アプリケーションとウィンドウ・カーネルの双方が、能動的にメッセージを送信することがあげられる。この性質を活用するために、オブジェクトの堆積モデルを拡張し、オブジェクトの連結モデルを提案する。

この論文では、次のような言葉を用いる。

(1) クライアントとサーバ：これらの言葉は、プロセス間通信におけるクライアント・サーバ・モデルに基づくものとする。すなわち、プロセス間通信のパターンにより、最初にメッセージを送る方をクライアント、受ける方をサーバと呼ぶ。クライアント・サーバ・モデルは、本来対称的なプロセスを最初にメッセージを送る方と受ける方に分類することで、

"A Window System based on the Object Stacking and Cascading Models"

Yasushi Shinjo, Tsutomu Takekawa, and Kenji Onaga  
University of the Ryukyus  
E-Mail: yas@ie.u-ryukyu.ac.jp

プロセス間通信のプログラムを記述しやすくするものである。(この論文では、サーバという言葉を提供するものという意味では用いない。)

(2) ウィンドウ・カーネルとウィンドウ・アプリケーション：前者は、ディスプレイやキーボードを管理するもの、後者は、前者に働きかけてウィンドウを表示させるものである。(Xウィンドウ・システムでは、それぞれサーバとクライアントと呼ばれている。)

## 2 オブジェクトの堆積

オブジェクトの堆積の基本的な考え方は、一様なインタフェースを持つオブジェクトを積み重ねることである。オブジェクトとは、手続きとデータをカプセル化したものであり、公開された手続きによってのみ内部のデータの操作が許される。インタフェースとは、オブジェクトが受け付けることができる手続きの集合である。積み重ねる (stack) とは、上位層のオブジェクトが下位層のオブジェクトの識別子を保持し、かつ、上位層のオブジェクトが自分自身の機能の実現するために下位層のオブジェクトを呼び出すことである。

図1に、3つのオブジェクトが積み重ねられている様子を示す。上の2つのオブジェクトは、堆積可能オブジェクト (stackable objects) であり、それぞれ下位層のオブジェクトの識別子を保持している。最下位層のオブジェクトは、基底オブジェクトであり、他のオブジェクトの識別子を保持していない。

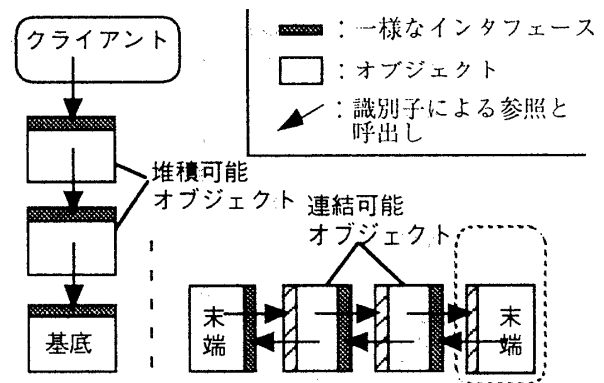


図1 堆積

図2 連結

オブジェクトの堆積に基づくファイル・システムでは、各ファイル・サーバは、フィルタ操作、キャッシング、複製、移動といった高度な機能の中から1つの機能だけを持つオブジェクトを管理する。それらのオブジェクトを積み重ねることで、それらが持つ機能を統合的に利用することが可能となる。

### 3 オブジェクトの連結

ファイル・システムでは、サーバは、最初にクライアントからの要求メッセージを待っている。一方、ウィンドウ・システムでは、ウィンドウ・カーネルは、ウィンドウ・アプリケーションからの要求メッセージを受け取り描画を行うこともあるが、逆に、ウィンドウ・アプリケーションに対して、能動的にメッセージを送ることがある。たとえば、Xウィンドウ・システムでは、カーネルは、マウスの動きやキーが押されたことをイベントとしてアプリケーションに通告する。このことは、プロセス間通信におけるクライアント・サーバ・モデルに反している。カーネルもアプリケーションも能動的にメッセージを送信しているからである。

プロセス間通信におけるクライアント・サーバ・モデルが成り立っていないことは、オブジェクトの堆積モデルを適用することができないことを意味する。オブジェクトの堆積では、上位層のオブジェクトが下位層のオブジェクトを呼び出すことがあっても、逆に下位層のオブジェクトが上位層のオブジェクトを呼び出すことはないからである。

この問題点を解決する方法としては、次の2つの方法が考えられる。

(1) クライアント・サーバ・モデルが成り立つように変換する。カーネルからの能動的なメッセージ送信を、アプリケーションからのメッセージ受信要求に対する応答として実現する。

(2) オブジェクトの堆積モデルを拡張し、上位層から下位層への参照・呼び出しに加えて、下位層から上位層への参照・呼び出しを可能にする。

ここで、(2)のモデルをオブジェクトの連結(object-cascading)と呼ぶことにする(図2)。オブジェクトの連結では、左右(あるいは上下)に、2種類のインタフェースがある。中間のオブジェクトは、左右(上下)に2つのインタフェースを持つ。これを、連結可能オブジェクト(cascadable object)と呼ぶ。両端のオブジェクトは、一方のインタフェースしか持たない。これを末端オブジェクト(end object)と呼ぶことにする。

### 4 関連した研究

文献[3]では、拡張可能ファイル・システムを実現するために2種類のオブジェクトを用いる方法が提案された。その文献で用いられている方法は、我々の言葉では、オブジェクトの連結モデルに基づいていると言える。そのシステムでは、我々のシステム[1][2]と同様に、高度な機能を実現するためにキャッシュを用いている。元のファイルが更新されたことをキャッシュに連絡するために、それらの間に結合が形成される。

### 5 おわりに

現在、X11システム上に連結可能なウィンドウ・オブジェクトの実現を進めている。X11の配布に含まれているXTVというアプリケーションを参考にしている。XTVでは、一つのウィンドウ・アプリケーションの画面を複数の利用者の表示装置に表示することが可能になっている。

XTVの機能を利用しつつ、さらに高度な機能を提供する連結可能なウィンドウ・オブジェクトの実現を進めていく。たとえば、多重表示、表示装置の切替え[4]、永続的なウィンドウ、フォントやグラフィック・コンテキストの変更、スケールの変更、イベントのフィルタリングなどが考えられる。

現在、Xプロトコルのレベルにおいて連結可能オブジェクトのインタフェースを規定している。しかしながら、このレベルでは、オブジェクトに対する操作のレベルが低く、部品(widget)を使う方法と比較して、プログラミングが困難である。今後は、連結可能なウィンドウ・オブジェクトに適したインタフェースを考案したい。

### 参考文献

- [1] Y.Shinjo and Y.Kiyoki: "The Object-Stacking Model for Structuring Object-Based Systems", IEEE Proc. of 2nd International Workshop on Object Orientation in Operating Systems (I-WOOS'92), pp.328-340 (1992).
- [2] 苅部, 新城, 清木: "オブジェクト堆積モデルに基づくファイル・サーバの実現", 情報処理学会研究会報告, 93-OS-58-2, pp.9-16 (1993).
- [3] Y.Khalidi and M.Nelson: "Extensible File Systems in Spring", 13th SOSP, ACM Operating System Review, Vol.27, No.5, pp.1-14 (1993).
- [4] 一柳, 北川: "Xウィンドウ・マイグレーション機構とその一実現例", 情報処理学会第47回全国大会講演論文集(5), 2D-2, pp.145-146 (1993).