

# Smalltalk によるソフトウェア設計支援環境 「電脳机」の実現

3M-1

森田孝司<sup>1</sup>, 高橋哲也<sup>1</sup>, 本田克己<sup>2</sup>, 勝間田敏秀<sup>2</sup>, 飯塚正樹<sup>2</sup>

<sup>1</sup> ㈱神戸製鋼所, <sup>2</sup> 横河・ヒューレット・パッカード㈱

## 1. はじめに

オブジェクト指向言語である Smalltalk を用いて、ソフトウェアの設計段階で発生する多種多様のドキュメントの作成および管理を支援する環境を実現した。このシステムの特徴は、ハイパーテキストの概念に基づいて、関連のある情報どうしを結びつけることにより、ドキュメントを参照する機能をパソコン上で実現したことである。そのため使用した Smalltalk 言語は MS-DOS 上で作動するサブセット版の Smalltalk/V であり、マルチウィンドウ、マウスなどによる優れたマンマシンインタフェイスや Smalltalk/V 自身の記述のために用意された豊富なプログラム部品が再利用できる反面、応答性の悪さやメモリの制限を克服するためのプログラミング技法が必要であった。

本稿では、実現したシステムの特徴を、Smalltalk/V 上でのインプリメントの方法とともに紹介する。

## 2. ソフトウェア構成

本システムのソフトウェア構成は図1のようになる。Smalltalk/V には、イメージと呼ばれるオブジェクトの集まりがあり中間言語で書かれているが、それを実行するインタプリタとプリミティブの部分がある。そしてそれらは MS-DOS の上で実行される。

しかし、Smalltalk/V においてプログラムの実行はオブジェクトにメッセージを送ることの繰り返しで、中間言語インタプリタであるため実行速度が遅い。このため本システムでは、専用のメソッドの追加・修正によるイメージの変更だけでなく、アセンブラによるユーザ定義

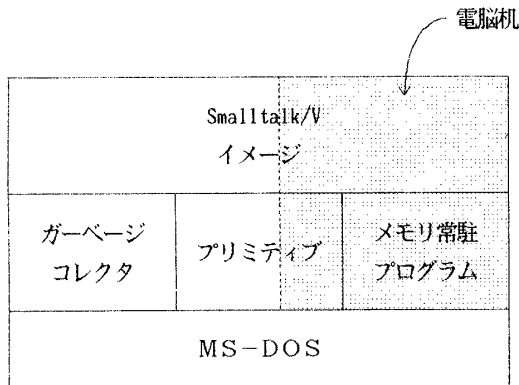


図1 ソフトウェア構成

のプリミティブメソッドを追加したり、C言語でメモリ常駐プログラムを書いたりして、実行速度の高速化をはかっている。

## 3. システムの機能

### (1) マルチメディア文書処理機能<sup>2)</sup>

テキスト、図形、画像など文書に含まれるそれぞれの要素はすべてオブジェクトとして扱うことができる。図2に示すようなマルチメディア文書において、(a)の文章、(b)の長方形、(c)の画像はそれぞれクラス "Text", "Rectangle", "Image" に属する1つのオブジェクト、すなわちインスタンスである。そして各クラス毎にオブジェクトに対するメソッドが定義されていて、たとえば "draw" という同じ名前のメッセージを(a), (b), (c)の各オブジェクトに送ったとしても、クラスが異なればそれぞれ異なるメソッドが実行される。そのため、文書の作成や編集などの主要機能は、オブジェクトのデータ型を意識することなくプログラムでき、他の言語で開発する場合と比較して大幅にプログラム作成量が少なく済む。

また、Smalltalk には上位のクラスのメソッドを継承するインヘリタンス機能があり、プログラム部品の再利

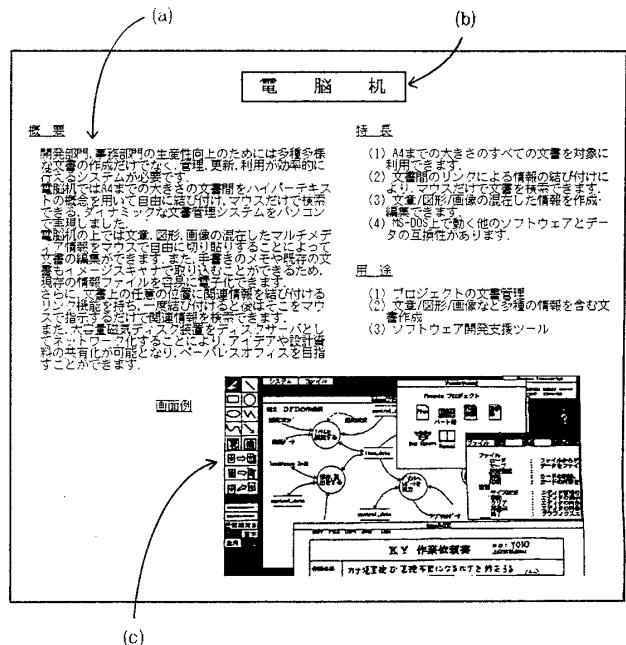


図2 マルチメディア文書

Implementation of Software Development Environment with Smalltalk

Takashi MORITA<sup>1</sup>, Tetsuya TAKAHASHI<sup>1</sup>, Katsumi HONDA<sup>2</sup>, Toshihide KATSUMATA<sup>2</sup>, Masaki IIZUKA<sup>2</sup>

<sup>1</sup> KOBE STEEL, LTD., <sup>2</sup> YOKOGAWA · HEWLETT · PACKARD, LTD.

用がはかれるが、テキスト・エディタを例にとってその再利用性を調べてみる。Smalltalk/V は本来、日本語に対応していないため1バイトコードのエディタになっているが、日本語にも対応させるため2バイトコードのエディタに改良した。このとき、従来のエディタがもっているクラスの下に、日本語エディタ用に表1に示す2つのサブクラスを新規に作成し、一部メソッドの追加を行ったが、大部分は上位のクラスのメソッドを継承している。表中には、それぞれのサブクラスについて、その上位のクラス名、新たに追加したメソッドの数、上位のクラスから継承したメソッドの数、上位のクラスにもともとあったメソッドの数を示している。これからもわかるように、もともとあったメソッドの約80%のメソッドが下位のクラスに継承され、新規に追加したメソッドの数は全体の約40%である。

表1 日本語エディタ用のサブクラス

サブクラス	上位のクラス	新たに追加したメソッド数	継承したメソッド数
KanjiEditor	TextEditor	31	27 (37)
KanjiPane	TextPane	25	57 (67)
合計		56	84 (104)

( ) 内は上位のクラスにもともとあったメソッド数

## (2) 情報リンク機能

ドキュメントを作成する過程では、単に文章や図形などを入力し編集するだけでなく、それ以外にも、他の資料を参照、引用したり、脚注を付けたり、より詳細化するなどの作業を頻繁に行う。本システムでは、このような作業を、関連のあるドキュメント間にリンクを付けることによって行っている。すなわち、参照、引用、脚注、詳細などの必要な箇所に、その対象となるドキュメントをリンク付けして保管しておくことにより、必要な時にそれらのドキュメントを容易に取り出してくることができるのである。

図3には、リンクを辿って次々に画面上に表示されたドキュメントを示すが、Smalltalk/V のマルチウィンドウとマウスによる優れたマンマシンインタフェースを利用すれば、ドキュメント内のリンク箇所をマウスでクリックするだけでそこにリンクされているドキュメントを引き出せる環境を、容易に実現できるのである。

## (3) A4版ドキュメント表示機能

日常我々が使っているドキュメントのほとんどはA4版が基本であるが、パソコンのディスプレイでは解像度の点でA4版の大きさを実用的に表示できるものが少ない。多種多様のドキュメントの作成および管理を支援する環境として本システムを実用的に使うためには、必然的に、高解像度ディスプレイを用いて実物大の大きさにドキュ

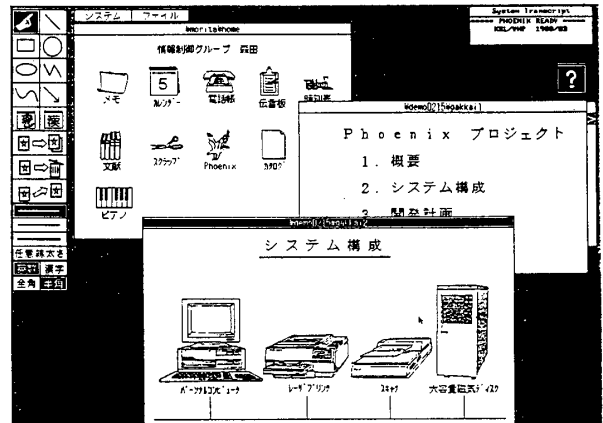


図3 情報のリンク

メントを表示できることが必要になってくる。しかし、Smalltalk/V ではこのような高解像度ディスプレイに対してA4版の大きさのドキュメントを取り扱うことはできない。

そこで本システムでは、大量のメモリを消費するビットマップ（ディスプレイ画面の画素ビットの配列でグラフィックス表示の基本となる）をメモリ上ではなくファイルとしてもち、他のデータ領域としてのメモリ領域を確保するとともに、ビットパターン移動、複写などのビットマップ操作をつかさどるクラス "BitBlit" のメソッドの一部をC言語で書き換え、大きなサイズのドキュメントにも対応できるようにした。また、それと同時に、従来メモリの制限のため時間のかかっていたドキュメントのウィンドウ表示の速度を1秒以下にすることができ、高速化もはかることができた。

## 4. おわりに

Smalltalk/V のオブジェクト指向、インヘリタンス機能によるプログラム部品の再利用性や、マルチウィンドウ、マウスによる高度なマンマシンインタフェースにより、比較的容易に本システムを構築することができた。Smalltalk/V が中間言語インタプリタであることや、MS-DOS 上のシステムであることから生じる、実行速度や扱えるデータ容量の問題は、アセンブラやC言語によって、実行速度のボトルネックとなっている部分や大量のデータを扱う部分を書き換えることによって解決し、本システムを実用的なものとすることができた。

### 参考文献

- 1) 本田 他：ハイパーテキスト概念に基づくソフトウェア設計支援環境「電脳机」の開発、情報処理学会第37回全国大会論文集
- 2) 高橋 他：オブジェクト指向型プログラミングによるマルチメディア文書処理、第32回システムと制御研究発表講演会予稿集、255/256 (1988)