

解説



ビットマップディスプレイとマウスを利用したエディタ†

伊東 健†† 田中 往成††

1. はじめに

ゼロックスのビットマップディスプレイとマウスを利用した画面エディタの歴史は長い。ハードウェアは1973年のAltoに始まって、D0, Dorado, Dandelionなどがあり、ソフトウェアは主なもので Bravo, BravoX, Gypsy, UGH, Markup, Draw, Sil, (J)Starなどがある。また、1100 SIP上で動く Interlisp-Dや Smalltalk-80におけるエディタもあるが、本特集の別稿にあるのでここでは特に触れない。

このうち、Bravo, BravoX, Gypsy, UGHはテキストエディタであり、Markup, Draw, Silは基本的にはページ単位のグラフィックスを中心としたエディタである。(J)Starはエディタの名前ではなく、ゼロックスが製品化したワークステーションの名前であるが、そのソフトウェアの中にテキストとグラフィックスが一体となったエディタを含んでいる。このドキュメント編集機能の本特集では便宜的にJStarのエディタと呼ぶことにする。ちなみにJStarのエディタが動いているハードウェアがDandelionである。その他のエディタとしては、Dandelionをベースにしたソフトウェア開発環境で使用しているプログラマ用のエディタがあるが、これも特に名前がないので便宜上、ゼロックスのプログラミング用エディタと呼ぶ。

Bravoは1975年にパロアルト研究所で作られ、Altoのビットマップディスプレイの特徴を本格的に取り入れたマルチフォントエディタであり、後のJStarのエディタの雛型となった。用途は印刷並に美しい仕上がりができるドキュメンテーションとプログラミング用のテキストエディタである。Bravoでは画面を水平に幾つかに分割し、同一ドキュメントの別の部分を表示したり、他のドキュメントを表示し相互にエディティングできる。この各ウィンドウは上下方向にの

み拡大や縮小できるものであり、ウィンドウ間のオーバーラップやウィンドウの任意方向サイズや位置を自由自在に許す本格的なマルチウィンドウ方式ではない。

マルチウィンドウを本格的に駆使したものとしてJStarのエディタがある。以下、これを中心にしてビットマップディスプレイとマウスを利用したエディタについて解説する。

2. ハードウェア

ディスプレイ

横1024ドット×縦808ドットの17インチビットマップディスプレイを使用している。JStarのエディタは、画面に表示されるレイアウトと実際にプリントされた結果とが一対一に対応する“*What you see is what you get (WYSIWYG)*”の概念で設計されているが、このディスプレイの解像度はそれが考慮されている。つまりこの約72ドット/インチの解像度は、印刷における1ポイントが72分の1インチであるので画面上の1ドットが1ポイントの大きさに匹敵している。

実際、“ポイント”の単位は文字サイズをはじめとしてプリントの際の極めて重要な単位であり、72ドット/インチという解像度はWYSIWYGを画面上で実行させるエディタの設計において大変便利である。

ポインティングデバイス

二つボタンのマウスを使用している。(Altoや1100 SIPにおいては三つボタンマウスである。)覚えやすく間違いにくく、且つ機能的であるためのボタンの数について実験が重ねられた結果、二つボタンが採用された。

キーボード

図-1にキーボードのレイアウトを示す。このキーボードの特徴は出来るだけファンクション・キーを減らして単純化を狙っている点にある。さらに、マウスを右手で扱う人が大部分であるので、編集時に頻繁に使うファンクション・キーは左側に集中させてある。

† Editors on Bitmap Display with Mouse by Ken ITO and Michinari TANAKA (Office Systems Division, Fuji Xerox Co., Ltd.).

†† 富士ゼロックス(株)オフィスシステム事業部開発部

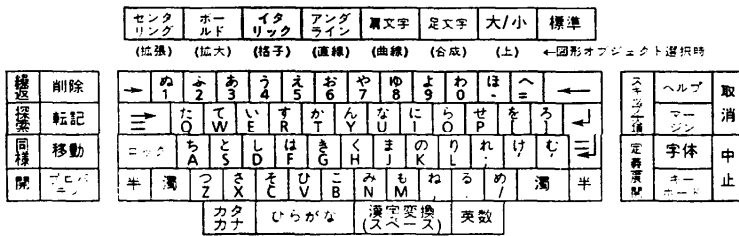


図-1 JStar のキーボードレイアウト

3. JStar のエディタのテキスト編集

JStar のエディタは、OA 用多機能ワークステーションのエディタとして、単なる文字列編集にとどまらず、図形編集やレイアウト編集が包含されている。設計の目標は、印刷なみの仕上がりを有するオフィス文書を編集できる画面エディタの実現であった。そのため文字は、字体、大きさ、フェースなどの属性を有し、図形はビジネスグラフィックスを編集できる機能を有している。

レイアウト編集機能も豊富で、上下左右のマージンの指定は勿論、行間隔や段落間隔を1ポイント単位で微調整できる。加えて頁の上見出しや下見出しの内容と体裁の指定、頁番号の位置や文字サイズの指定等、キメ細かなオフィス文書を編集できるよう設計されている。

本稿ではそういった JStar のエディタが有する各種機能を網羅する事は割愛し、“エディタ”としてどのような特徴を持っているかについて述べる。

モードレスエディタ

従来のエディタ、特にキャラクタ・ディスプレイを使用したエディタにおいては、ハードウェアの制限上モード・エディタが多かった。これは入力/編集等のコマンドを特別のファンクション・キーを用いないで、入力されるデータと同じキーより入力する方式であるが、コマンドとデータの区別がしにくく、入力/編集のミスが頻発するという操作上の煩わしさがある。

これに対して、モードレス・エディタは編集のコマンド用キーをテキスト入力用キーから分離し、操作性(ユーザ・インタフェース)を向上した点にある。つまりエディタは常に入力モードにあり、入力される文字はキャレットの位置に入る。しかし、これを実現するためには、テキストの追加/変更がなされている部

分が常に画面上に表示されて、その様子が逐次確認できることが必要である。

JStar のエディタは、「名詞」「動詞」の形式によるモードレスエディタである。テキスト編集においては、マウスカーソルによる位置指定後すぐキーボードからの文字入力可能な状態となる。文字挿入以外の操作にお

いては、まず最初にオブジェクト(名詞)をマウスで選択し、次にオペレーションの内容(動詞)をキーボードから入力する。ユーザインタフェース設計上特に注意が払われた点は、「動詞」の種類を極力減らす事である。特にキーボード左側にある「削除」「転記」「移動」「プロパティ」「繰返」の五つのキーはテキストやグラフィックスに限らず、JStar のエディタのあらゆる状況で共通に使用される「動詞」である。

オブジェクト

名詞となるオブジェクトには、以下のものが含まれる：

- ① 文字、単語、文、段落などのテキスト・オブジェクト
- ② 点、直線、三角形、四角形、円、楕円、合成図形などのグラフィック・オブジェクト
- ③ 文書、レコードファイル、フォルダ、ドロア、プリンタ、電子メールボックスなどのデスクトップ*上のアイコン・オブジェクト**

選択(セレクション)

文字列の選択は基本的には、始点をマウスの左ボタンで、終点を右ボタンで指定する。マウスでカーソルを所望の文字の上へ移動し、マウスの左ボタンを押すとその文字が選択され、更にもう一度(2度目)押すとその文字を含むワード(単語)、即ち文字や数字の連続列、またはスペースやタブの連続列が選択される。更にもう一度(3度目)押すと、文全体が選ばれ、続いて更にもう一度(4度目)押すと、パラグラフ(段落)全体が選択される。

選択において日本語と英語の違いは、英語が単語と単語の境界がスペースなどで明確になっているのに対

* “デスクトップ”の例を図-2に示す。JStar ワークステーションにおいては、オフィスの事務机の上をシミュレートして画面に表示している。

** “アイコン”はデスクトップの上に表示される絵文字のことであり、図-2に例が示されている。文書、フォルダなどがアイコンとして表示されている。

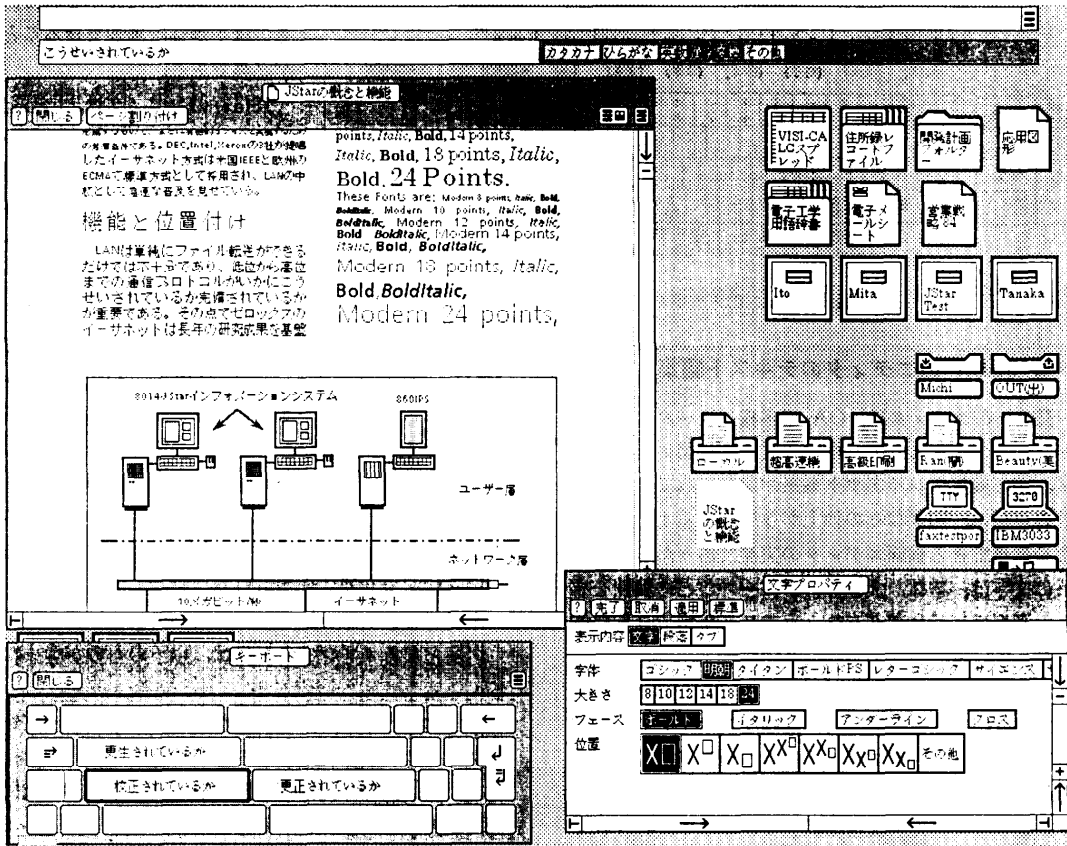


図-2 JStar 画面の表示例

し、日本語では、コードだけで判別できない。複雑な解析をして時間をかけて単語を厳密に選択するメリットはあまり無く、それよりは、エディタとしてのパフォーマンスを重視し、JStarのエディタにおいては2度目のマウスクリックで漢字の連続列、ひらがなの連続列、カタカナの連続列が選択されるようにしてある。

マウス左ボタンによる選択範囲を必要に応じて右ボタンによって拡張や調整をする。拡張の単位は、左ボタンによる選択が文字の場合は文字、単語の場合は単語、文の場合は文、段落の場合は段落となっている。

シンプルコマンド体系

動詞となるコマンドには以下のものが含まれる：

- ① 「削除」「転記」「移動」「プロパティ」「繰返」

この五つのコマンドは、ほとんどのオブジェクトに対して共通に使用される。「プロパティ」は、選択されているオブジェクトの色々な属性をサブウィンドウに表示させるコマンドで、図-2の右下に文字プロパ

ティの表示例を示す。「繰返」は、直前に実行したオペレーションを繰り返す時に使用するもので、大変便利なコマンドである。例えば、ある文字列を削除して別の文字列を挿入する操作をした後、同じオペレーションを別の文字列に対しても実行したいとき、その文字列を選択し、「繰返」キーを押せば良い。

② 「同様」

①に加え、テキストとグラフィックスオブジェクトに対して共通に使用されるコマンドである。これは選択されているオブジェクトが有している色々なプロパティ（属性）を、同種の別のオブジェクトのプロパティに一致させる時に使用する。例えば、選択されているオブジェクトが文字列の場合、その文字プロパティ（サイズとかスタイルなど）を別の文字のプロパティと同一にするために使用する。この際は「名詞」「動詞」「名詞」の順序となり、最後の名詞は、一致させたいターゲットオブジェクトを指す。

③ 「探索」

所望の文字列を捜し出すコマンドで、単純に捜し出すのみの場合と、捜し出した後、別の文字列に置換する場合とある。対象となる文字列は、サイズや字体などの属性まで含んだ一致を条件とする場合と、そのような属性を無視する場合のどちらで探索するかを指定できる。また、探索の範囲は文書全体であるのか、文書の残りの部分であるのか、あるいは選択されている範囲であるのかも指定できる。文書編集の効率をあげる効果的なコマンドである。

④ 「センタリング」「ボールド」「イタリック」「アンダライン」「肩文字」「足文字」「大」「小」「標準」

これらは、選択されているオブジェクトがテキストの時に使用されるコマンドである。「大」「小」「標準」は、それぞれ、文字サイズを一段階大きくする時、小さくする時、標準サイズにする時に使用する。

⑤ 「拡張」「拡大」「格子」「直線」「曲線」「合成」「上」

これらはグラフィックオブジェクトに対して使用されるが、詳細は後述する。

⑥ その他

以上の各コマンドはキーボードから直接入力できるものであり、その他のコマンドはマウス操作によって補助メニューウインドウを開きその中に表示させてマウスで選択する。

スクロール (Scroll)/サム (Thumb)

スクロール 上下のスクロール機能に加え、左右のスクロール機能がある。WYSIWYG の概念で表示されるため、横幅の広い文書を表示する時や、マルチウインドウで文書を多数表示させていて、横幅全体を表示できない時などは左右スクロールが必要となる。スクロールは、図-2 の例にあるように、文書ウインドウの右端と下端にそれぞれ上下左右の矢印が表示されており、そこをマウスでポイントして左ボタンを押す事によって実行する。

サム 本を書棚から取り出して手に取り、最初の辺りを開いたり、真ん中辺りを開いたり最後の辺を開いたりする時に親指で操作するが、英語でこの事をサミング (thumbing) という。これと同じように、何ページにもあるいは数十頁にもなる文書の適当な位置をサミングする機能がある。

具体的には、図-2 の文書ウインドウにおいて、右上端の下向き矢印の下のマイナス記号と、右下端の上向き矢印(図では他のウインドウに隠れて見えない)の上

のプラス記号(一部見える)の間のエリアが、該当文書の先頭から最後までに対応しており、“サミング”したい位置にカーソルを置いてマウスの左ボタンを押せばその位置に対応する頁が表示される。この機能は文書ファイルが大きくなればなるほど便利な機能である。

豊富な文字種を使用した文書編集

明朝体、ゴシック体の2種類のスタイルについて、8、10、12、14、18、24ポイントの6種類のサイズが使用できる。英数字においては、ボールド体、イタリック体、ボールド・イタリック体もある。図-2 のドキュメントウインドウに各種のスタイルやサイズの文字が表示されている例が示されている。

フォントファイルは固定ディスクに書かれているが、その中には文字のビットパターンそのものに加え、一文字ごとに文字幅、高さ、ベースライン位置などの情報が盛り込んである。この情報は、プリントされる時の高解像度文字パターン(12ドット/mm)の一文字ごとの情報と対応づけられている。この情報を元に、画面上で1ビット単位の任意の位置に文字を表示して WYSIWYG を実現する。

マルチウインドウ

最大6個のドキュメントウインドウを表示でき、ウインドウ相互間で編集ができる。

仮想キーボードによる多言語処理

カナ漢字変換による日本語入力の際の同音語表示/処理のためと、多言語入力処理のために、キーボードレイアウトを小さなウインドウに表示させている。図-2 の左下にその仮想キーボードに同音語が表示されている例を示す。

文法処理を含んだカナ漢字変換の際の同音語は、一度に最大8個まで表示でき、選択は所望の同音語が表示されている位置に対応するキーを打鍵すればよい。

ドイツ語、フランス語、ロシア語や、特殊な数学記号などを入力する際には、この仮想キーボードをそれぞれ所望の言語モードに設定すればよい。

4. JStar のエディタの図形編集

図形編集機能には、以下に示す基本的な図形と表、グラフ等が用意されているが、オフィス内でのいわゆるビジネスグラフィックスはこの基本的図形だけで大部分作成可能である。ここでは基本的図形について述べる。

入力できる基本図形には、点、直線、曲線、三角、四角、円、楕円等がある。

① 選択(セレクション)の図形への拡張
 カーソルをある図形上に移動し、マウスの左ボタンを押すとその図形が選択されるが、図形には拡張性があるため図形を選択したときのカーソルの位置により選択のされかたを区別している。例えば、四角形の場合は 図-3 に示すように8カ所のポイント

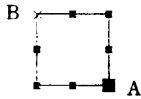


図-3 四角形の選択

トがあり、A地点付近にカーソルを置いて選択すればAが選択される。A点が拡大縮小の際の移動ポイントとなる事を明示するため、四角形の他の点よりもA点の■マークを大きくする。

② プロパティの図形への拡張

テキストにおけるプロパティの考え方は図形にも適用されている。例えば 図-4 に

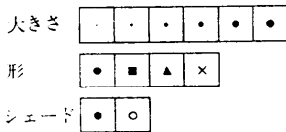


図-4 点プロパティ

示すように、点においても大きさ、形、シェードといった三種類のプロパティを備えている。

③ 図形用コマンドキー(仮想キー)

7個の図形用コマンドキーを仮想キーボード上の最上段機能キーにマッピングしている(図-1 参照)。

拡大(縮小)と拡張

拡大あるいは拡張キーを用いて行うが、選択されたポイントは次のように呼ばれる。

誘導ポイントと固定ポイント

図形のどこかを選択したのち拡大/拡張キーを押すと、マウスカーソルが一番近い点(例えば図-3のA点)の誘導ポイント(■マーク)と反対側のB地点に固定ポイント(×マーク)が現れる。誘導ポイントはマウスの動きに応じて移動するが、固定ポイントはそのままである。拡大と拡張は似通った機能であるが、前者は原形に対して相似形に変化するが、後者は変形

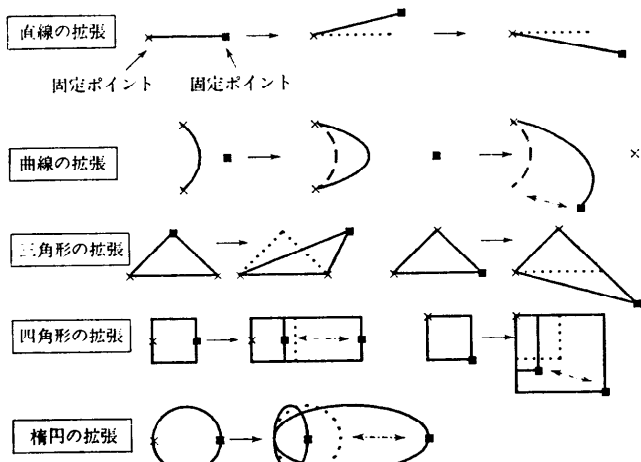


図-5 基本図形の拡張

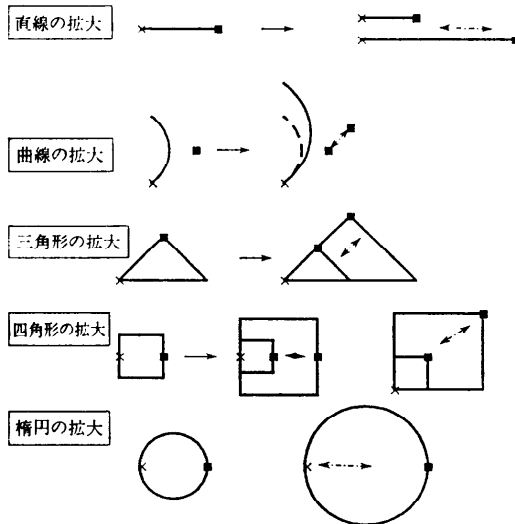


図-6 基本図形の拡大

を伴う(図-5、図-6)。即ち三角形を例にとると、拡大の場合は、誘導ポイントを含む一辺が平行移動をし相似形が保たれる一方、拡張の場合は誘導ポイントは自由に動き、他の二点が固定ポイントとなる。

その他の機能

合成

複数の図形が組み合わせられた複雑な図を部分的に移動/コピーするとき、細かな部分を個々に選択するのは困難であるが、それらをまとめて一つの図形として定義できれば取り扱いが非常に容易になる。この定義を合成とよんでいる。一旦合成された図形は、単体の

図形とまったく同様に扱うことが可能で拡大/拡張も任意の倍率にできる。当然ながら、合成された図形を分解する機能もある。

シェーディング

閉じた図形には5種類のシェーディング(網かけ)がかけられるが、さらにその上に5種類のパターンが重ねられる。そのパターンは上乘せ式であるために $2^5=32$ 通りで、シェーディングと組み合わせると合計 $5 \times 32=160$ 通りある。

重ね合わせ

複数の図形が重なる場合、上下の位置関係を規定する必要がある。このために仮想キーボードの「上」キーを押すことにより、トグルスイッチの要領で重なり合った図形の上下関係を指定できる。

格子

複雑な図形や正確さを要求される図形は、バックグ

ラウンドに格子(目盛り)があると描き易い。5種類の間隔と3種類の目盛りパターンの合計15種類の中から適当な格子をプロパティ・ウィンドウで設定できる。その目盛りは「格子」キーによってオンオフできる。

5. プログラミング用エディタ

通常、プログラム開発用エディタにはJStarのエディタのような出版印刷並の多種類の文字種、字体、サイズは要求されない。また、美しい文書を作成するのが目的ではないので、WYSIWYGの機能は特に要求されない。それよりもプログラム用エディタとしてより便利な機能を有した方が望ましい。その場合、基本的な機能だけを内蔵し、特にプログラミング/デバッグに必要な機能はオプションとして追加が可能な方式をとった方がよい。

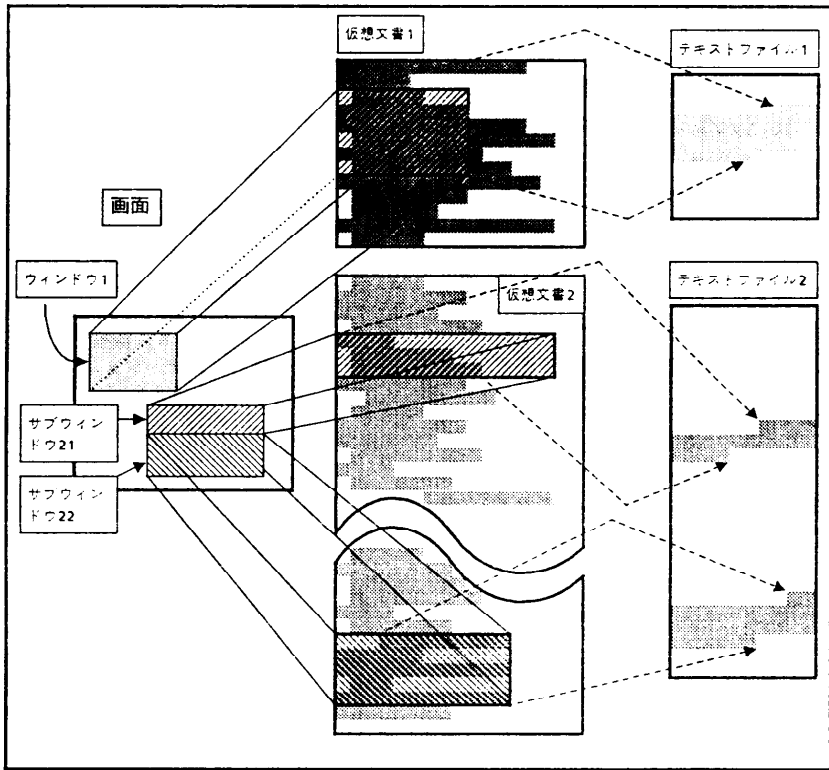


図-7 ビットマップディスプレイとマルチウィンドウによるプログラミング用エディタの概念図

画面上のウィンドウ1には、テキスト・ファイル1が読み込まれてその一部が表示されている。テキスト・ファイルは単なる文字コードの羅列であるが、ウィンドウからはその文字列を読みやすくするために内部的に作成された仮想文書の一部を覗いている。ウィンドウは分割(スプリット)することにより、同一ファイルの異なった部分を覗くことができる(サブウィンドウ21とサブウィンドウ22)。なお、サブウィンドウ22では、テキストはウィンドウの右はしからはみ出さないように折返しをしてある。

ゼロックスのプログラミング用エディタは、JStarのエディタと同様にモードレスエディタであり、削除、転記、移動、繰返などの基本的なコマンドや機能は共通してある。オフィスワーカー用のエディタであるJStarのエディタには無いが、プログラミング用エディタは以下の機能を有している。

1. 多数のファイル(テキスト)を同時に参照できるようにするためのマルチ・ウィンドウのハンドリングを更に高度にし、ウィンドウサイズや表示位置を自由に変更したり、重ね合わさったウィンドウの上下関係を自由に変更することが可能。

2. 一つのウィンドウ内でテキストの前後関係を見るために分割(スプリット)が可能。またその境界線の位置は任意に移動が可能。

通常のファイルは図-7に示すようにファイルそのものが仮想的文書に対応づけられ、その部分的なイメージを画面上的ウィンドウ(任意の大きさ)より覗き込む感覚で参照する。プログラミング用エディタには、プログラムの参照や解読のみを目的とするスタティックな使い方と、ソース・コードの作成や変更をするダイナミックな使い方がある。したがって、両者の区別をもうけることにより、前者の使用目的の際のソース・コードの変更ミス等を防ぐことが可能になる。なんらかのコマンドで両者の切り替えをするとき、ファイルの内容はバック・アップに書きこまれ、変更は仮のファイルになされる。それは、編集の途中で変更の取り消しをする場合や、システムの不意のクラッシュの際のデータ損失を防止するためである。

6. ま と め

以上、JStarのエディタを中心にゼロックスのエディタについて述べた。ビットマップディスプレイとマ

ウスによるエディタの特徴は、WYSIWYGの実現と多機能なマルチウィンドウの実現がしやすい事、およびテキストエディティングとグラフィックエディティングを一体化したエディタを実現しやすい事などがあげられる。

今後ハードウェアコストが更に下降するに従い、このような方式のエディタがワークステーションのみならず、ワープロやパソコンにも急速に普及していく事は想像に難くない。

参 考 文 献

- 1) Smith, D., Irby, C. and Kimball, R.: The Star User Interface: An Overview, AFIPS Proceedings of National Computer Conference, Vol. 51, pp. 515-528 (1982).
- 2) Smith, D., Irby, C., Kimball, R., Verplank, B. and Harslem, E.: Designing the Star User Interface, Byte, Vol. 7, No. 4, pp. 242-282 (Apr. 1982).
- 3) 伊東: 8012-JStarにおける入力方式, 情報処理学会日本文入力方式研究会資料(1983).
- 4) 上林, 伊東, 上田: オフィスプロフェッショナルワークステーション(JStar)と統合プログラミング環境ワークステーション(1100 SIP), 情報処理, Vol. 25, No. 2 (1984).
- 5) 鈴木: 高機能ワークステーションについて, 情報処理, Vol. 25, No. 2 (1984).
- 6) 上林: ワークステーションユーザインタフェースの構成技術, 情報処理, Vol. 25, No. 2(1984).
- 7) 上林: ワークステーションの実際, サイエンス別冊(ニュービジネス特集号)(1983).
- 8) 伊東: ワークステーションの考え方と機能, 事務と経営(1984年3月).

(昭和59年4月11日受付)