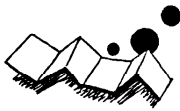


解説

高機能ワークステーションの応用†



斎藤 信男††

1. はじめに

高機能ワークステーションの応用を考える場合、その特徴をとらえ、それらをいかす適用場面や分野を考えてみるのがよい。このような目安となる特徴としては、次のものがある。

(1) 従来の CRT 端末に比べて、グラフィックス機能がすぐれていること。

ビットマップディスプレイやポインティング装置を装備しているので、高度なグラフィックスの機能が実現されている。

(2) 個人用のワークステーションであること。

高機能のワークステーションではあるけれども、使い方はあくまでも個人用であり、単一ユーザの使用環境を前提としたシステムである。

(3) ネットワーク機能を備えていること。

個人用のワークステーションが高度な応用場面で活きてくるためには、何らかのネットワーク機能（特にローカルエリアネットワーク LAN の機能）を標準的に備え、他のワークステーション、端末、周辺装置、後置の大規模ホストなどとの情報の交換が可能でなければならない。

上記の特徴ごとにそれらを活かした応用場面について展望をすることも一つの方法ではあるが、高度な応用では、恐らく上記の特徴の複合的効果がそれらの実現を支援していることになるであろう。したがって、次節以下では、応用分野ごとに展望を進めることとする。

高機能ワークステーションは、LSI などのハードウェア技術の進展、グラフィックスやネットワーク機能などのソフトウェア技術の深化、ユーザインタフェースの重視などシステム技術あるいは応用技術の確立などが相まって登場してきた製品であり、いわゆる改良型の技術の産物である。したがって、その応用も、高

機能ワークステーションが初めてもたらしたということよりも、従来の応用場面で問題となっていた部分が飛躍的あるいはある程度改善されたということの方が多くであろう。したがって、ここでは、従来の端末装置などを利用した応用と比較して、どんな改良点が得られどんな差があり得るのかということを、各応用場面について特に強調するものとする。

応用分野としては、上記の特徴を融合して総合的な技術として実現されるソフトウェア開発環境、やはり総合的な技術となるオフィスオートメーション(OA)、グラフィックス機能に重点を置いた CAD/CAM や CAI、ネットワーク機能に重点を置いた分散処理について考え、それらの展望を示す。

高機能ワークステーションは、商用の製品として作られてからあまり多くの時間を経過せず、実際に大規模な応用システムを実現している例は少ないと考えられる。したがって、本稿は、現状の解説よりも、従来の応用場面と比較しながら高機能ワークステーションの応用のされ方への筆者の予想、期待を中心とした展望になることをあらかじめ断っておく。

2. ソフトウェア開発環境への応用

2.1 開発環境で有効になる高機能ワークステーションの特徴

ソフトウェアの開発における生産性を向上させることは、ソフトウェア工学の主要な目標の1つである。そのために、概念の形式化、方法論の追及などとともに、ソフトウェアの開発を計算機システムに支援させるために使うソフトウェア工具類の実現がいろいろ試みられてきた。また、それらの工具類を集大成し一貫した1つのシステムとしてまとめたソフトウェア開発環境を実現することが、最近の話題となっている¹⁾。従来から整備されてきた Lisp (特に Interlisp) 系の環境、より一般的な基盤となる Unix システムなどがよく使われ、また、より高度な開発環境の実現も試みられている²⁾。

† Application of High Level Workstation by Nobuo SAITO
(Department of Mathematics, Faculty of Science and Technology, Keio University).

†† 慶應義塾大学理工学部数理科学科

高度なソフトウェア開発環境を実現するためには、高機能ワークステーションが有効であることが予想される³⁾。その主な理由は、次の通りである。

(1) 高度なユーザインタフェースを効率よく実現できること。

(2) ネットワーク機能により資源や情報の共用が効率よく実現できること。

ソフトウェアの開発は、大規模な集団によって実施されることが多いが、その基盤となる作業は個人ベースである。したがって、プログラマやシステムエンジニア各個人が、計算機システムと質の高いユーザインタフェースで結ばれていれば、良い開発環境を実現することが可能となる。

高度なユーザインタフェースを提供できる高機能ワークステーションの機能として、簡便なグラフィックス機能、多重ウィンドウ機能、使い易いポインティング装置などがある。グラフィックス機能とポインティング装置により、従来、文字列として表示していた種々の入力コマンドや出力結果が、適当な大きさのシンボル、表、グラフ、図形を介して表示される。一般に、人間にとっては、抽象的に記述した文字列を理解するよりも直接対象を表わしたシンボルの意味を理解する方が容易である。このように対象を直接表わすシンボルなどを操作するユーザインタフェースを、B. Schneiderman は直接操作 (direct manipulation) と呼んでいる⁴⁾。直接操作により、システムの操作の学習が容易になること、システムに習熟した者は操作をより速く行えること、エラーメッセージがほとんど必要ないこと、ユーザが作業の大枠を常に把握できることなど、いろいろの利点があると考えられる。これらの利点は、人間の個人的な作業に土台を置く、ソフトウェア開発環境において、有効に働くものと思われる。

高機能ワークステーションは、個人用計算機システムとして設計されるから、大容量のファイル装置、高機能の印刷装置、高機能の入力装置 (イメージスキャナなど) などのハードウェア資源をネットワークを介して共用できる環境が望ましい。さらに、ソフトウェア開発においては、言語処理系、文書処理系、仕様記述系など多くのソフトウェア資源を利用する。これらのソフトウェア系は、各ワークステーションに移植するよりもネットワークを介して共用した方が効率が良い。もちろん、ソフトウェア系へのアクセスの能率が低下しないことを保証できるだけのネットワークの性能を実現しておかなければならない。

一方、ソフトウェア開発においては、多数の開発参加者の間で必要な情報を随時交換して作業を進めていかなければならない。また、必要なドキュメンテーションの作成や参照も随時行わなければならない。これらの操作に対しても、ネットワーク機能は有効に働く。情報の交換は、ネットワーク上に実現された電子メールにより能率良く行える。電子メールは、一般にログをとってあるから、作業の履歴の点検にも役立つ。また、ネットワーク上の分散データベースにより、ドキュメントの相互参照も容易に行える。

高度なユーザインタフェースによる効率的な個人の作業、ネットワーク機能による開発プロジェクト全体の管理、この2つの機能により高度なソフトウェア開発環境が実現されるであろう。

2.2 ソフトウェア工具における具体例

以下では、ソフトウェア工具において高機能ワークステーションが有効に働くと思われる場面を、具体的に示す。

(1) 編集系

テキストの編集作業は、ソフトウェア開発において最も頻繁に行われるものであり、それを支援するテキスト編集系にはいろいろの工夫の余地がある。従来の行型編集系 (たとえば QED や TECO) が、最近では画面編集系 (たとえば vi や emacs)⁵⁾ にとって替わられようとしている。画面編集系は、前述の直接操作の1つの例であり、一度これを使ったユーザは、行型編集系を再び使用する気にはなれないと言われている。

画面編集系でも、表示こそ画面全体を使った2次元領域を使っているが、各操作のコマンドは、文字、単語、行を対象としたものが基本となっている。2次元領域 (ある文字から別の文字までの領域) を操作するためには、まずその領域を定義するコマンドをいくつか実行しなければならない。これに対し、ビットマップ表示装置とポインティング装置とを備えた高機能ワークステーションでは、領域の指定や表示が容易に行える。Alto システムの画面編集系の Bravo⁶⁾ や Star 8000 の編集系では、単語、行、パラグラフなどの領域の指定が容易にできる。

また、カーソルの移動は通常の画面編集系ではやはりカーソル移動のコマンド (next line, next page, search など) の実行により実施するが、高機能ワークステーションでその操作を能率よく実行する1つの例として図-1 に示すようにスクロールバー (scroll bar) にカーソルを移しポインティング装置を上下す

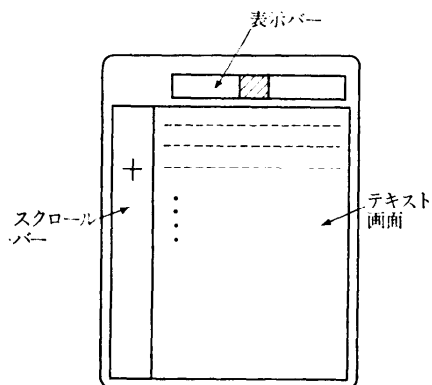
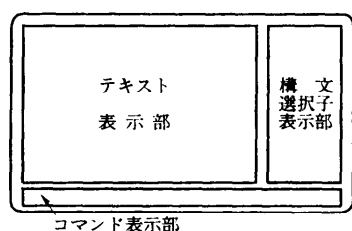


図-1 画面編集系

図-2 構文編集系の画面の使い方
(通常の画面端末装置を使用した場合)

ることにより容易に行う方法がある。丁度、巻紙を巻いているように扱うことができ、わずらわしいコマンドの指定がポインティング装置の動きだけで済んでしまう。また、図-1 に示すように画面上部の表示バー (thumb bar) の任意の場所をポインティング装置で指定し、ファイルの指定された近傍にまでカーソルを進め、その部分を1画面分表示する方法もみられる。

プログラムの原始コードの編集をする時に、プログラミング言語に関する構文などの情報をあらかじめ編集系が知っていて、正しい構文のテキストだけが編集結果として得られるように制御する編集系を、構文向き編集系 (syntax oriented editor)⁷⁾ という。これは、構文木を構成してゆくための編集系と見ることができる。すなわち、図-2 に示すように、画面をテキスト表示部と構文選択部に分け、選択した構文要素がもし分解できるならば下位の構文部分木を生成してテキスト表示部に示される。構文木の終端部節点に対応する識別子や定数などを入力してゆくことにより、全体の構文木が生成される。このような構文向き編集系を従来の画面端末装置で実現すると、木の節点間のカーソルの移動は適当な制御コマンドまたは制御キーの指定により行わなければならない、わずらわしい。ま

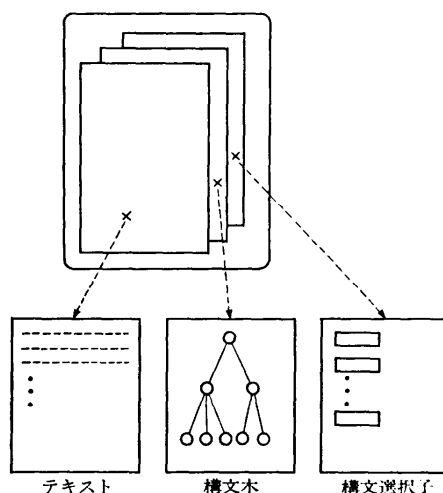
た、構文選択部に表示する選択子が多数あると、画面を複数回切り替えて表示しなければならない。構文向き編集系の1つの問題点と思われる会話の回数が多くなることを避けるために、高機能ワークステーションが有効に働くと思われる。

たとえば、多重ウィンドウの機能を使い、図-3 に示すように1つにはテキスト情報、1つには木構造、1つには構文選択子を表示する。木構造の表示は、PERQのファイルシステムで既に使っている。ポインティング装置によりカーソルを任意の節点に移動すれば、容易に木の走査が可能となる。多重ウィンドウにおいて、1つのウィンドウに表示できる情報が減少して却って使いにくいことがあるけれども、その点は、文字フォントの縮小、ウィンドウの重ね合わせなどにより工夫する余地がある。

(2) 仕様記述/解析系

ソフトウェア開発におけるいろいろのレベルの仕様記述とその解析を支援するソフトウェア工具は、従来から多くの人達によって試みられてきた。要求仕様記述/解析系として、ISDOS⁸⁾ や SREM⁹⁾ があり、実際の開発プロジェクトでも使われている。また、プログラムの内部仕様の記述方法論として、いわゆる流れ木を基盤とした PAD¹⁰⁾ や HCP¹¹⁾ のような図式表現法が我が国で提案されている。

このような仕様記述/解析系は、ソフトウェア開発の設計段階で使用される系であり、入力した仕様記述を何らかの意味で解析し、その解析出力を人間が判断し

図-3 構文編集系の画面の使い方
(高機能ワークステーションを使用した場合)

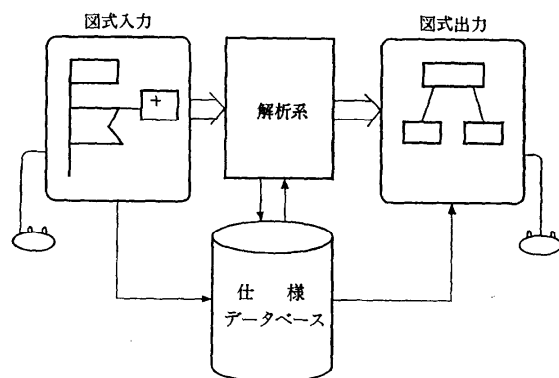


図-4 仕様記述/解析系

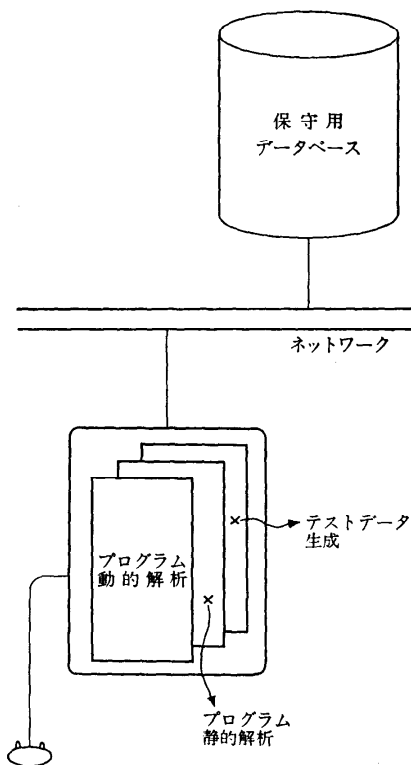


図-5 ソフトウェア保守工具系

ながら仕様記述の修正/追加などの次のステップに進んでゆく。この場合、仕様記述の書き易さ、解析系にとっての解析の容易さ、仕様記述や解析結果の読み易さの3つの要因が、記述の手段、方法論や解析系の設計などを左右する。設計段階という比較的大まかな判断や理解を要する場面では、図形情報の利用が有効に

働くと考えられる。PAD や HCP が図式表現で仕様記述をしようとしたのは、その方が仕様の書き易さ、読み易さともに向上すると判断したからであろう。IBM 社の IPT で使われた HIPO も、この方向に沿ったものと考えられる。一方、ISDOS の記述言語 PSL は、文形式の記述を採用しているが、解析系は、図式表現に比べてこの方法がはるかに容易である。しかし、解析結果の読み易さを向上させるために、解析系 PSA のレポートには、図形レポートや行列レポートのような図形情報を出力することを、行印字装置に対してさえ試みている。このような図形情報を利用しようとする試みは、いくつかの仕様記述システムでも行われている^{12), 13)}。

このような仕様記述/解析系に対して、高機能ワークステーションは、どのように役立つであろうか。高機能ワークステーションでは、簡便に図形情報が扱え、また、それらの操作がポインティング装置により容易に実施できるので、人間と解析系との会話もスムーズになる。これらの特徴を利用すれば、図-4 に示すような仕様記述/解析系が実現可能となり、ソフトウェア設計の工具として一層有効に働くであろう。この場合、図式入力と図式出力は、多重ウィンドウ機能を使って同時に画面に表示してもよい。入力した結果が、即座に解析されて出力結果として表示されることは、それだけ人間の思考を助けてくれる。また、解析系は、一般に文字列に対して実現した方が容易であり、図式表現と文形式表現の相互変換が内蔵されていれば、解析系の実現も容易になり、仕様データベースに格納する情報の形式も自由に選択できる。

(3) 保守用工具

ソフトウェア開発における保守のフェーズに対する費用は、大きな比重を占めると言われており、その費用を減少させるためのさまざまな工具類が考えられる。我が国でも、協同システム開発(株)の下でソフトウェア保守技術開発計画が進められ、いくつかの工具のプロトタイプが Unix の上で実現されている¹⁴⁾。そこでは、プログラム解析、再構成、テストデータ生成、回帰テスト、システム構造解析、データ名統一化、ドキュメント生成、ワークスルー支援などの工具類がみられる。これらは、すべて会話型システムとして実現されているが、1つの工具

でも、いろいろの形式や種類の入力、出力があり、従来の画面端末装置で実現すると、会話の操作がわずらわしい。また、1つのソフトウェアの保守のためには、当然、いくつかの工具類を適用して必要な処置をしてゆく。このようなことを考えると、多重ウィンドウの機能を持つ高機能ワークステーションを有効に利用できるであろう。図-5に示すように、多重ウィンドウを介した会話型工具により、保守技術者は保守作業を進めてゆく。ネットワークを介して保守用データベースを利用できれば、一層有効な工具系となるであろう。

3. オフィスオートメーションへの応用

ここでは、いわゆる事務処理型の応用も含めて、オフィスオートメーションという。これらは、対象が素人のユーザというだけで、高機能ワークステーションとの関係はソフトウェア開発環境の場合と同じであると考えてよい。ソフトウェア開発環境は、計算機の玄人を対象としているのに対し、オフィスオートメーションでは素人が相手であるから、却って高度な情報処理を要求される¹⁵⁾。特に、ユーザインタフェースの設計には、気を配らなければならない。

3.1 図形情報の利用

オフィスオートメーションの対象となるユーザは、いわゆるエンドユーザであり、計算機システムに対しては必ずしも深い理解や経験があるわけではない。したがって、文字列による入力/出力操作を行うよりも、図形で入力/出力ができた方が使い易い。

たとえば、コマンドの入力を、オフィス機器など具体的な対象の図形(アイコン)を介して行う方法がStar 8000システムで採用されている。また、事務処理用プログラムを記述する言語を表形式で行うQBEやOBE¹⁶⁾は、ビットマップ表示装置を備えた高機能ワークステーションにとっては扱い易いものとなる。QBE(Query by Example)は、関係データベースへの問い合わせを、表形式で行う言語であり、一般性のあるメニュー言語と考えてもよい。OBE(Office Procedure by Example)は、QBEの自然な拡張であるが、更に一般的な言語で、ドキュメント生成系、電子メール系、レポート生成系、応用ソフトウェアなどと結合して、高度な情報処理システムを作成することができる。

さらに、応用ソフトウェアの出力結果をグラフ表示するビジネスグラフは、エンドユーザにとってわかり易い。これは、従来からあるグラフィックス機能付き画面端末装置を使ったものが開発されているが、高機

能ワークステーションはここでも有効に働こう。いろいろの種類のグラフ、カラー表示の利用など工夫の余地が残されている。

3.2 文書処理

計算機システムによる文書処理は、オフィスオートメーションでは特に重要な機能である。Bravoのように、編集系に既に文書整形の機能を持たせる方式も考えられるが、高度な文書処理を行う独立の処理系がいくつか開発されている。Unix上に開発され手軽に使えるnroffやtroff¹⁷⁾、素人にも使い易い機能を豊富に揃えてあるscribe¹⁸⁾、数学における複雑な式を柔軟に扱えるTEX¹⁹⁾などがその代表的なものである。

nroffなどの文書処理系は、一般にそのサブコマンドを必要な箇所に加えた文書をまずテキスト編集系で作成し、それを文書処理系で翻訳して最終的な整形をした文書として出力する。このとき、TEXのように複雑な数式を扱う場合には、特殊な文字フォントが必要となるので、いわゆるグラフィックプリンタを出力機器として使用しなければならない。実際にハードコピーを見てみると、出力結果がわからず、思いがけず資源の無駄遣いをすることもある。

Knuthは、TEXのような文書処理系の次の問題は、会話型の系にすること、および、図形情報も扱えるようにすることであると言っている。この2つの問題点は、高機能ワークステーションを利用することにより解決されよう。図形情報の入力は、ビットマップ表示装置により可能となる。ファクシミリを利用することも可能である。また、整形した後の出力結果を表示することも、種々の文字フォントを生成しておくことができるので、容易に実施できる。更に、高機能ワークステーションの多重ウィンドウや会話型機能を利用すれば、図-6に示すような会話型文書処理系が可能となるであろう。ハードコピーは、ビットマップ表示装置上の表示をそのまま印刷することによって得られる。

我が国においては、オフィス用文書では日本語の扱えることが必須のこととなる。漢字のフォントを多種類用意することは費用がかかるので、何らかの制限が必要となるであろう。高機能ワークステーションでは、任意の形の文字フォントの表示が可能であるから、出力結果を取りあえずビットマップ表示装置に表示することは容易である。

以上のように、文書処理において高機能ワークステーションの果たす役割は大きいと予想される。

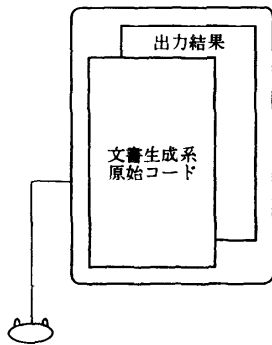


図-6 会話型文書生成系

3.3 電子メール

電子メールも、オフィスオートメーションにおいては重要な機能の1つである。個人用のワークステーションが、ネットワークにより結合されていることは、電子メール実現のために必須の条件であり、ネットワークを介した電子メール系は、単一の TSS 上の電子メール系を拡張することにより実現できる。

高機能ワークステーションを利用した電子メールでは、従来の画面端末装置を利用した場合に比べて、送受信できる情報の種類がふえる。すなわち、図-7に示したように、ファクシミリなどを利用した図形情報の伝送、および音声入力装置を利用した音声情報の伝送である。音声出力装置を利用すれば、文字テキストを音声情報に変換して音声として出力することもできる。図形情報は、地図、物体の形状、表、グラフなど実際に電子メールとして伝送したい対象が多数あり、従来の電子メールと比較してはるかに便利なものとなるであろう。

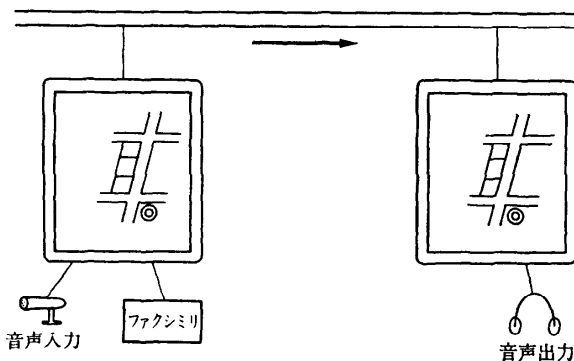


図-7 電子メール

4. CAD/CAM と CAI

コンピュータグラフィックスの機能を利用したいろいろの工業製品の設計支援システムと、それで作成された情報を総合的に把握し、生産過程の総合的管理をする管理支援システムは、最近の大きな話題の1つである。高機能ワークステーションは、ビットマップ表示装置を備えており、かなりの精度のグラフィックス機能を実現できる。したがって、CAD/CAM の中で、高機能ワークステーションを有効に利用する場面があるであろう。

総合的な CAD/CAM システムは、モデリング、データベース、マンマシンインタフェース、および、設計生産の4つの基礎技術の上に成り立つと考えられる²⁰⁾。モデリング技術は、コンピュータグラフィックス技術と密接に結びついており、高価なグラフィックス装置を利用したものが多い。高機能ワークステーションが有効に働く場面は、むしろマンマシンインタフェースの向上に於いてであろう。モデリングを実施してゆく過程においては、ユーザがいろいろの入出力媒体を使ってユーザのアイデアがモデルに十分反映されるような会話が必要である。これは、パーソナル化したワークステーションが大いに役に立つであろう。

また、CAD/CAM は、対象物のモデリングから始まり、生産過程に関する情報を共用のデータベースに格納し、それを中心としたいいくつかの機能を実現するモジュールから成り立つ。図-8は、そのようなシステムの構成の例²¹⁾である。ここでは、グラフィックス機能を中心とした CAD だけでなく、生産計画、生産管理、品質管理のような情報処理、ドキュメント処理や情報検索のようなオフィス事務処理などの機能が

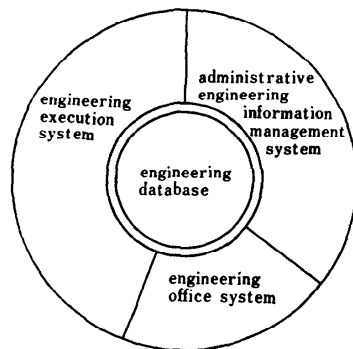


図-8 CAD/CAM の構成図

見られる。このような情報処理は、高価なグラフィックス装置を使うのではなく、ネットワークを介して接続されたパーソナルな高機能ワークステーションで実行する方が適している。

一方、コンピュータを利用した教育系、学習系は、従来からいくつかの試みがなされてきた。たとえば、PLATO²²⁾では、プラズマディスプレイ装置を開発し、図形の表示、スライドと図形出力の重ね合せなどによりユーザとのインタフェースを向上させる努力をしている。医療技術などの教育には、図形情報の表示は不可欠のものであり、それを低価格で実現することが必要となる。また、子供に対する教育には、明らかに文字情報よりも図形情報の方が有効であると考えられる²³⁾。高機能ワークステーションを利用すれば、このような図形情報の表示は能率よく実施できる。また、音声の入出力装置を備えれば、より有効なCAIシステムが実現できるであろう。なお、CAIシステムは教育機関で使用するだけでなく、ソフトウェア開発環境におけるモジュールベースのモジュールの学習、マニュアルなどの検索と学習などに広く応用できるものである。

5. 分散処理系

高機能ワークステーションは、あくまでも個人用のシステムであり、その環境で扱えるプログラムやファイルの規模は限定されてしまう。これを越えるためには、ネットワーク機能を基礎にした分散処理系の開発が必要となる。このような分散処理系では、ネットワークを介した電子メール、分散型データベースなどが可能となる。中央集中型のシステムと比較した場合、高度なユーザインタフェースを実現する機能を、ワークステーションに機能分散したものが、この種の分散処理系であると考えることができる。

高機能ワークステーションに基づいた分散処理系の例として、カーネギーメロン大学計算機科学科のSpiceシステム²⁴⁾がある。この目的は、1985～1990年代の同学科の研究環境を整えることである。約200人の研究スタッフに対して、1人2万ドルの投資をすると考え、これを高機能ワークステーションとローカリエリアネットワークに基づいた分散処理系の構築にあてる計画である。このワークステーションのハードウェアの最終目標値は、次のようなものである。

- 実行速度 3 MIPS
- 主記憶 2メガバイト

- 制御記憶 16キロワード
- 番地空間 2^{31} バイト
- 2次記憶 100メガバイトウインチェスタ型
ディスク+フロッピディスク
- ネットワーク機能 10メガビット/秒
- 高密度画面端末 1000×1000ビットマップ
表示装置
- ポインティング装置
- 音声入出力装置

現在、Three Rivers社のPERQを100台近く導入している。

このプロジェクトは、ソフトウェアプロジェクトであり、新しいハードウェアを開発することはない。提供する言語として、一般の研究者にはAda、人工知能関連の研究者にはLispを考える。特にLispは、最近仕様が決められたCommon Lispにする予定である。

オペレーティングシステムは、高度な分散処理をねらったものを新しく開発する。分散処理の核としては、柔軟な多重プロセッシングを実現するものを設計する。現在、Accent²⁵⁾と呼ばれる核が実験的に作成されている。これは、図-9に示すように、各ワークステーションにはKernelとnet server processがあり、ユーザの仕事はプロセスを複数個使って実行してゆく。この時、柔軟なプロセス間通信を介してプロセスの協同作業が行われるので、ユーザから見ればプロセスがネットワーク上のどの節点に存在するかということ意識しなくてもよい。

ファイルシステムは、既存の計算機システムのファイルも扱えるような中央ファイルシステム(CFS)と、各ワークステーションの局地的なファイルを扱うスパイスファイルシステム(SFS)とが階層構造をなし、両者の情報の転送が柔軟に行えるように制御する。

ビットマップ表示装置を使った高度なユーザインタフェースを実現するために、プログラムの実行環境を

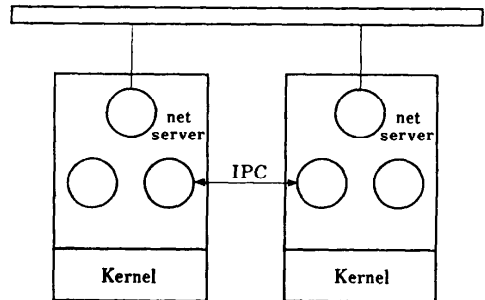


図-9 Accentの機能

管理する Environment Manager と、多重ウィンドウに任意の図形を書いて表示することを制御する CANVAS が用意されている。

応用プログラムとしては、テキスト編集系、ドキュメント生成系、音声や図形の情報も扱える電子メール、回路図などの作成をする描画システムなどが実験的に開発されている。

6. おわりに

高機能ワークステーションを利用する適用分野と、そこにおけるワークステーションの役割などについて筆者の見解を中心に展望を述べてきた。現在、この種のワークステーションの価格は高く、未だ本格的な応用システムを開発しているところは少ない。これは、技術的方向が誤っているのではなく、単なる価格の問題が障害になっているに過ぎない。ハードウェア技術の進展により、その問題は解決できよう。したがって、高機能ワークステーションを本当に活かすことができる応用分野やシステムを真剣に考えることが早急に必要であろう。

参 考 文 献

- 1) Osterweil, L.: Software Environment Research: Directions for the Next Five Years, Computer (IEEE Computer Society), Vol. 14, No. 4, pp. 35-43 (Apr. 1981).
- 2) 笈 捷彦: Ada と支援環境, 情報処理, Vol. 22, No. 2, pp. 138-142 (Feb. 1981).
- 3) Gutz, S., Wasserman, A.I. and Spier, M.J.: Personal Development Systems for the Professional Programmer, Computer, Vol. 14, No. 4, pp. 45-53 (Apr. 1981).
- 4) Sehneiderman, B.: Direct Manipulation: A Step Beyond Programming Languages, Computer, Vol. 16, No. 8, pp. 57-68 (Aug. 1983).
- 5) たとえば Gosling, J.: Unix Emacs User's Manual, Carnegie-Mellon University (1982).
- 6) Lampson, B.W.: Bravo Manual, Xerox Palo Alto Research Center (Sep. 1979).
- 7) たとえば, Medina-Mora, R., Notkin, D.S. and Ellison, R.J.: ALOE Users' and Implementors' Guide (Second Edition), The Second Compendium of Gandalf Documentation, Dept. of Computer Science, Carnegie-Mellon University (May 1982).
- 8) Teichroew, D. and Hershey, E.A. III.: PSL/PSA: A Computer-Aided Technique for Structured Documentation and Analysis of Information Processing Systems, IEEE Trans. on Software Engineering, Vol. SE-3, No. 1, pp. 41-48 (Jan. 1982).
- 9) Bell, T.E., Bixler, D.C. and Dyer, M.E.: An Extendable Approach to Computer-Aided Software Requirements Engineering, ibid., pp. 49-60 (Jan. 1982).
- 10) 二村良彦, 川合敏雄: PAD によるプログラムの開発, bit, Vol. 12, No. 4-6 (March-May 1980).
- 11) 花田収悦: プログラム設計図法, 企画センター (Feb. 1983).
- 12) 米田 潔, 中野会利子, 藤林信也, 岩元莞二: SDMS/設計サブシステムの図形出力機能, 第24回情報処理学会全国大会論文集 (Mar. 1982).
- 13) 南 俊郎, 杉尾俊之, 武内 惇, 椎野 努: 日本語をベースとした仕様記述言語: NBSG の図形入力について, 第27回情報処理学会全国大会論文集 (Oct. 1983).
- 14) ソフトウェア保守技術開発計画テクニカルレポート, No. 2, プロトタイプ特集, 協同システム開発株式会社 (June 1983).
- 15) 石田晴久: オフィスオートメーションの今後の展開, 情報処理, Vol. 22, No. 10, pp. 930-937 (Oct. 1981).
- 16) Zrooff, M.M.: QBE/OBE: A Language for Office and Business Automation, Computer, Vol. 14, No. 5, pp. 13-22 (May 1981).
- 17) Typing Documents on the Unix System: Using the-ms Macros with Troff and Nroff, Bell Laboratories (Nov. 1978).
- 18) Scribe Introductory User's Manual, Unilogic, Ltd. (May 1980).
- 19) Knuth, D.E.: TEX and Meta Font, American Mathematical Society and Digital Press (1979).
- 20) 木村文彦: CAD/CAM システム構築のための基礎技術, 情報処理, Vol. 24, No. 1, pp. 17-25 (Jan. 1983).
- 21) Okada, H.: IBM Total Application Concept in Engineering (TACE), Proceeding of Inter Graphics' 83, Session A1 CAD/CAM Interface and Expandability (May 1983).
- 22) Obertino, P. et al.: Elementary Reading on PLATO IV, Computer-Based Education Research Laboratory, University of Illinois (1977).
- 23) Papert, S.: Mindstorms: Children, Computers, and Powerful Ideas, Basic Books, Inc., New York (1980).
- 24) Ball, J.E., Barbacci, M.R. et al.: The Spice Project, Computer Science Research Review 1980-1981, Dept. of Computer Science, Carnegie-Mellon University (1982).
- 25) Rashid, R.F. and Robertson, G.G.: Accent: A Communication Oriented Network Operating System Kernel, 8th SOSP Proceedings (Dec. 1981).

(昭和 58 年 10 月 28 日受付)