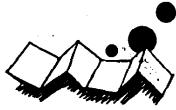


解説

Smalltalk-80 におけるエディティング機能†



上 林 憲 行††

1. はじめに

Smalltalk-80™* は Xerox PARC/SCL (Palo Alto Research Center/Software Concept Laboratory) で、1970 年代初頭から、研究・開発が進められたものであり、統合プログラミング環境として、オブジェクト指向言語として、グラフィックス利用者インタフェースとして、それぞれについて一つの理想的な姿を探求し、具体的に結実させたものとして、優れた評価がなされている^{1)~4)}。

Smalltalk-80 には、通常のコンピュータシステムで利用できるテキストエディタに相当するものは存在しない。Smalltalk-80 では、編集可能なテキストを内容とするビュー(ウィンドウ)は、すべて利用者からは、標準的な(システムが用意してある)編集コマンドを利用してテキスト編集が可能である。さらに、その編集可能なテキストオブジェクトに対して編集コマンドに対応するメッセージを送れば、プログラムレベルでもテキスト編集が可能である。この両者は、ただメッセージの送信主体が違うだけである。

Smalltalk-80 は、探索的に試行錯誤的なプログラミングプロセスを効果的に支援する Unified Programming Environment である。その目指すところは、(ソフトウェア)設計とその実装に本質的に付随する、様々な複雑さ(Complexity: 問題の複雑さ、作業工程管理の複雑さ、試行錯誤的なプログラミングプロセスの管理)をあらゆる角度から体系的に管理し、その複雑さの管理をできるだけ利用者(設計者)から解放することにある。そして、設計者には、より本質的な問題解決に対する知的的好奇心と旺盛な創造性の発露の空間的・時間的・心理的な余裕を与えることである。そのためには、システムを自在に制御できる。そして、システムが自分自身を完全に理解・把握(self

contained) して、その内容を説明できること、そうした相互(プログラマとシステム)の意志の伝達が容易にできる(他の獲得知識を必要とせず)ことがあげられる。プログラミングやソフトウェア設計における意志の伝達・表明の第一ステップは、編集インタラクションである。その意味で、Smalltalk-80 の編集システムは、編集に付随する煩わしさで前向きな意志が挫折されるようなことのないように注意深く配慮されている。付属品としてのエディタではなく、システムの不可分の要素として重みが大きな特徴である。

本解説では、まず、Smalltalk-80 におけるテキストエディタの利用者インタフェースおよびそのモデルについて述べる。

2. テキスト編集における利用者インタフェースの特長

(1) 表示スタイル

Smalltalk-80 が前提としている表示スタイルの環境を形成する要素は、高解像度のビットマップディスプレイである。

1) マルチウィンドウ

マルチウィンドウによって、ディスプレイ画面上に、複数の入出力チャンネルを設定するコトが可能となり、利用者は同時に、画面を切り換えないで複数の情報を参照し、かつ並行的に、編集作業が可能である。また、あるウィンドウのテキストの全部または一部を別のウィンドウの任意の個所に転記することが、同じウィンドウ内の操作と同様に行うことが可能である。

2) マルチフォント

Smalltalk-80 には、標準的には、10 種類の表示フォントがシステムに組み込まれている。また、標準オプションとして、さらに 20 種類の表示フォントを使用することが可能である。この表示フォントの種類により、テキスト中の強調したい部分の抑揚ができ、その内容を視覚的に的確に認識することが可能になる。

3) 任意サイズの編集ウィンドウ

† Editing Facility for Smalltalk-80 by Noriyuki KAMIBAYASHI (Software Technology Center, Fuji Xerox Co., Ltd.).

†† 富士ゼロックス(株)

* Smalltalk-80: Trademark of Xerox Corporation

Smalltalk-80のウィンドウは、リソースの許す範囲で任意サイズのウィンドウを画面の任意のロケーションに設定することが可能である。また、ウィンドウのレイアウトは、テキストの内容または利用者のスタイルによって、縦長、横長の自在なレイアウトができる。

(2) 編集スタイル

1) 電子切り貼りスタイル編集

Smalltalk-80の編集インタラクションに使用するコマンドは電子切り貼りスタイル(cut, copy & paste)のものである。コマンドの数も、3種類で、概念的にもよく整理されているので、編集コマンドを学習するというフェーズが不必要となる。

2) モードレス編集

モードレス編集(modeless editing)は、編集インタラクションにおいて、モードつまり、キーボードからの入力ストリームの解釈(データとして、コマンドとして判断される)が、そのモードによって変化する状況が発生しない編集スタイルである。Smalltalk-80では、標準キーボードからの入力ストリームは、データとしてみなされ、コマンドを標準キーボードから与えることはない。

(3) コマンドインタラクションスタイル

1) マウス選択

編集の基本は、選択(テキストロケーション)である。Smalltalk-80では、このポインティングは、マウスによって行われる。マウスを用いることによって、画面を構成する画素の任意集合を、自在にポインティングすることが可能である。マウスボタンの機能は、自由に設定できるが、原則は、左ボタンは、ポインティング & 選択、中央ボタンは、オブジェクトに依存したポップアップメニュー、右ボタンはウィンドウ操作メニュー用に使用する。

2) ポップアップメニュー

編集は、マウスの中央ボタンによって表示されるポップアップメニューの中のコマンドを選択実行することによって行うことが可能となる。ポップアップメニューを用いることによって、目を画面から離すことなく、指先の操作で、軽快に編集インタラクションが可能である。

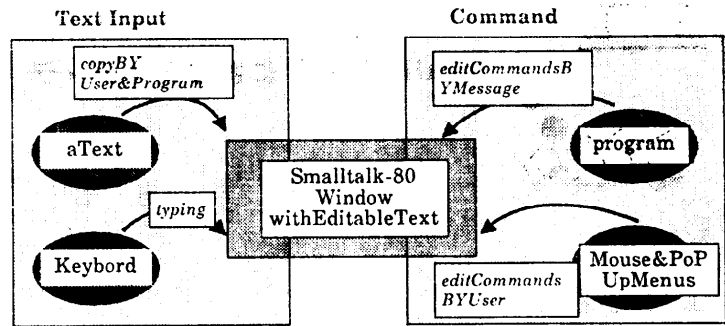


図-1 Smalltalk-80 Window and Editing Facilities

3. テキスト編集の利用者インタフェース

3.1 テキスト編集可能なビュー(ウィンドウ)

システムに標準的に用意されているテキスト編集可能なビューは、次のとおりである(図-1 参照)。

Workspace, Project, File List, System Browser, System Transcript, Chat, Change List, Inspector, Fillin The Blank 等

3.2 テキスト編集仮想マシン

Smalltalk-80の標準編集コマンド体系が想定する仮想マシンの仕様は、次に示すとおりである。

① テキストバッファとそれに対する入出力インタラクション

選択されたテキストに対して、copy または cut コマンドを作用させると、その内容は、テキストバッファに格納される。そして、paste コマンドは、指示された(マウスおよびプログラムによって)個所に、テキストバッファの内容を出力する。新たに、copy または、cut コマンド(typing)が実行されるまで、テキストバッファの内容は、保存される。

② Smalltalk メッセージバッファとそのメッセージの再実行機能・作用取り消しインタラクション

テキスト編集ウィンドウにおいて、最後に行ったメッセージを again コマンドによって再評価実行することが可能である。

③ 状態保存バッファとその復元インタラクション

テキスト編集ウィンドウにおいて、最後に行ったメッセージの作用を取り消して、そのメッセージの実行前の状態にテキストを復帰させるために undo コマンドが用意されている。

図-2 にテキスト編集仮想マシンの概念アーキテク

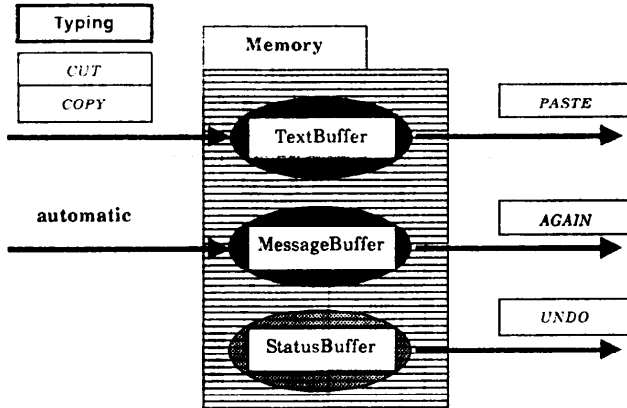


図-2 Virtual Machine for Smalltalk-80 Text Editing

チャを示す。Smalltalk-80 の基本的なテキスト編集操作はこの5つのコマンドとマウス、マウスボタンクリック（ボタンを数回連続的に押すこと）によって、簡単で軽快なインタラクションが可能である。

3.3 利用者モデルと基本操作

以下、Smalltalk-80 で機能するテキスト編集操作・方法について述べる。

(1) テキスト選択

◇位置決め：テキスト中の位置は、マウスによって直接ポイントすることが可能である。通常のエディタにおけるポインティングのためのコマンドおよびカーソル移動操作は、このマウスによって不要となっている。現在選択されている位置を表示する記号であるキャレット(↑)は、マウスによって自由にテキスト中を移動することができる。キャレットはマウスのカーソル

の位置の近傍の文字の前後のいずれかの位置が現在の選択位置となる(図-3 1), 2)参照)。

◇テキスト選択：テキストの一部およびその全体を選択する機能としては、英文の記述にそくした、意味的なテキスト選択機能が用意されている。任意のテキストを選択する場合は、まず、そのテキスト列の始点または終点にマウスのカーソルをもってゆき、マウスの左ボタンを押しながらその終点または始点までカーソルを移動する。そのカーソルの移動に連れて選択されたテキスト列は、ウインドウ上で白黒反転する(図-3 3), 4), 5)参照)。

◇ダブルクリック：ダブルクリックによる選択機能も、Smalltalk-80 の編集インタラクションを特長づけるものである。ダブルクリックによる選択機能としては、句読点のような区切り記号で区切られた word の選択(図-3 6)参照)。または、ピリオドで終端している文章の始点にマウスのカーソルを持ってゆき、ダブルクリックするとその文章(Smalltalk-80 のメッセージ expression はピリオドでその終端を表す)全体が選択される(図-3 8)参照)。同様な、意味的なダブルクリック選択は、表示テキスト全体、(), [] の中のテキストを選択する場合に使用される(図-3 9)参照)。

(2) テキスト挿入

表示されているテキストの任意の位置に、新たにテキストを挿入する操作は、まず、マウスのカーソルを

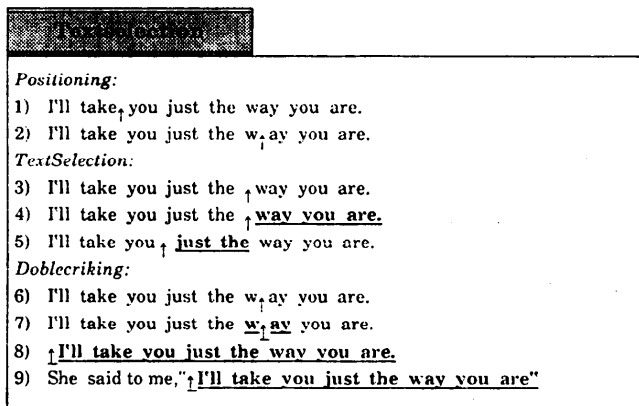


図-3 Text Selection in Smalltalk-80 Text Editing

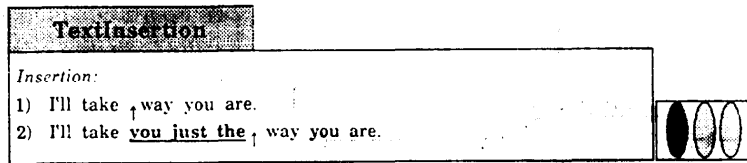


図-4 Text Insertion in Smalltalk-80 Text Editing

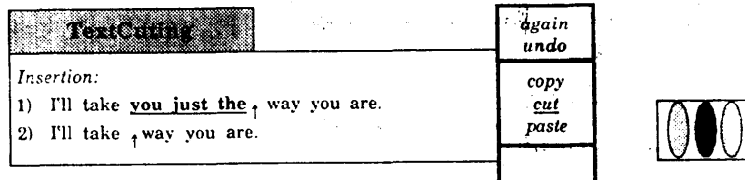


図-5 Text Cutting in Smalltalk-80 Text Editing

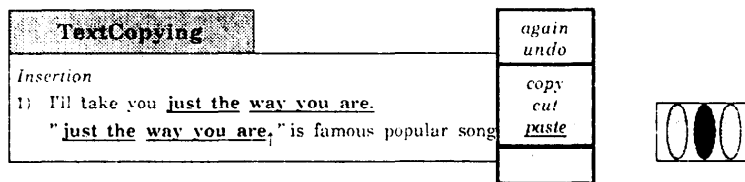
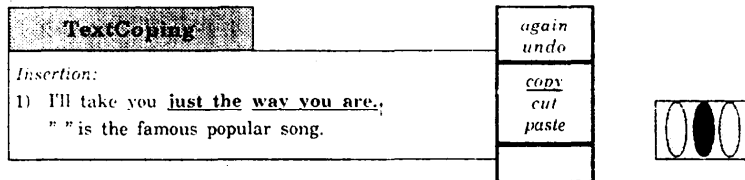


図-6 Text Copying in Smalltalk-80 Text Editing

その個所に移動させ（キャレットが表示された状態）として、タイピング（paste コマンド）をすれば、そのキャレットの後に、テキスト列が挿入される（図-4）。

(3) テキスト編集

すでに、入力されているテキストを編集する機能としては、以下のものがある。

◇削除：テキストの任意の部分を、削除するには、その削除を希望するテキストを、(1)で述べた機能を使用し、テキスト選択する。そして、ポップアップメニューの cut コマンドを選択すると、選択されているテキスト列は、表示画面からは削除される（図-5）。

◇転記：テキストの任意の部分を、他の、任意の個所（ウィンドウをまたがって）に、転記する場合には、転記すべきテキスト列を、(1)で述べたテキスト選択の方法で選択し、ポップアップメニューの copy コマンドを実行する。次いで、マウスのカーソルを使用し

て、転記先を指示する。そして、ポップアップメニューの paste を実行することによって、転記が可能となる（図-6）。

◇置き換え：任意のテキスト列を、他の任意のテキスト列に置き換える操作は、置き換えたいテキスト列（ソース）を、(1)で述べたテキスト選択の方法で選択し、ポップアップメニューの copy または cut コマンドを実行する。次いで、置き換えの対象となるテキスト列（ディステネーション）を選択し、ポップアップメニューの paste を実行することによって、そのテキスト列がソースのテキスト列と置き換わる。また、置き換えの対象となるテキスト列を選択したのち、置き換えたいテキスト列をタイプインすることによっても置き換えが可能となる（図-7 参照）。

(4) 取り消し・繰り返し・検索操作

以上述べた、電子切り貼り（cut & paste）スタイルの編集コマンド以外に、これらの編集コマンドを組

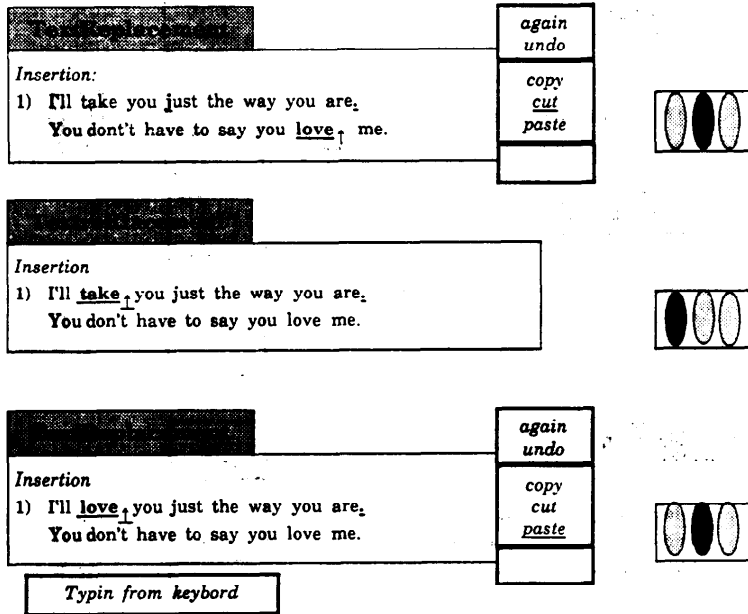


図-7 Text Replacement in Smalltalk-80 Text Editing

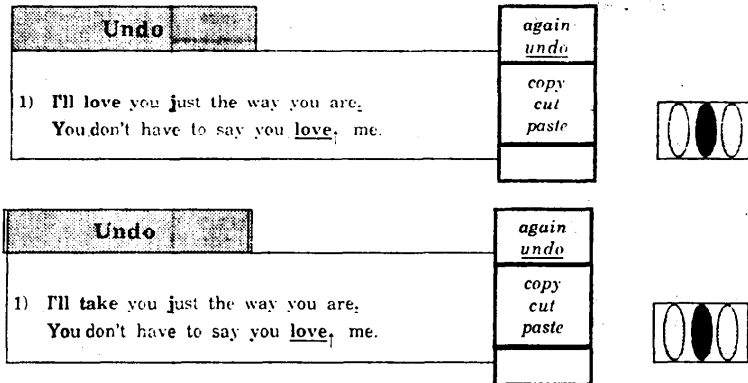


図-8 Undo in Smalltalk-80 Text Editing

み合わせたり、その作用を取り消したり、いま実行したメッセージを再度繰り返し実行する場合に必要なコマンドが用意されている。ただし、これらのコマンドは、テキスト編集のために特別に用意されたものではなく、Smalltalk-80と利用者の間のインタラクションコマンドの一部を形成するものである。以下は、テキスト編集における、こうしたコマンドの利用方法を述べる。

◇取り消し：テキスト編集操作において、誤動作や、ある操作あの実行結果が意に似わない場合に、その操作あの作用を取り消し、その操作（タイピングも）の

実行前の状態に、テキストを復元させる(図-8 参照)。

◇繰り返し：直前の電子切り貼り (cut & paste) スタイルの編集コマンドを、繰り返し実行する。例えば、テキスト中の、“you”を“mariko”に置き換える操作（置き換え）は、図-9のように、again コマンドを使用すると、軽快に行うことができる。また、ある文字列をカットする (cut) 操作も、again コマンドを使用することによって、次に一致した文字列が自動的にカットできる。

◇文字検索操作：テキスト中の任意文字パターンを検索し、その個所をハイライトする操作は、again コマ

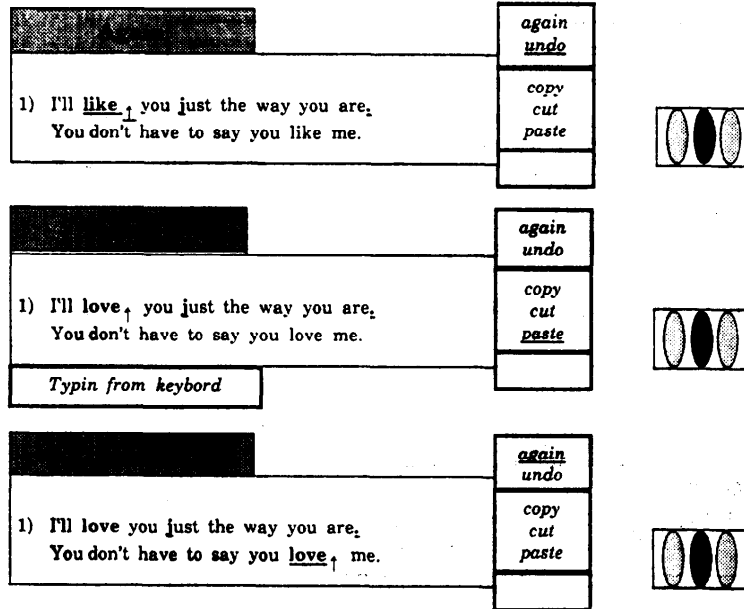


図-9 Again in Smalltalk-80 Text Editing

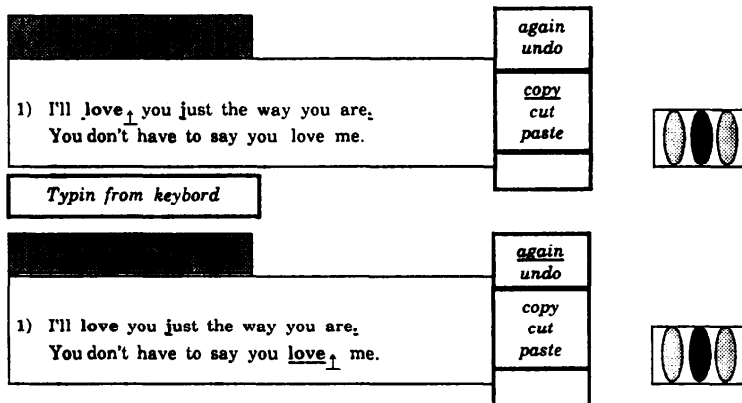


図-10 String Finding in Smalltalk-80 Text Editing

ンドによって行うことができる。検索したい文字をタイプして自動的に、again コマンドをポップアップメニューから、選択実行すれば、次に一致したその文字列をハイライトする (図-10 参照)。

(5) ジャスティフィケーション

利用者インタフェースレベルでは、ジャスティフィケーション (左揃え, 中央揃え, 右揃え) の機能は表層にでてこないが, 表示テキストに関連するクラスには, そうしたメソッドが用意されているので, 容易に利用者インタフェースのレベルで利用することが可能である。

(6) スクロール

システムの標準的なビュー (ウィンドウ) はすべてウィンドウの右端に, スクロールを指示する機能が表示される。Smalltalk-80 のスクロール機能は, 3種類用意されている (図-11 参照)。

◇ダウン (アップ) スクロール: マウスのカーソルをスクロールエリアにもってゆき, スクロールバーの左 (右) にカーソルをもってゆくと, 下向き (上向き) の矢印が表示され左 (右) ボタンを押すと, 前向き (後戻り) のスクロールができる。スクロールエリアの上端からの矢印までの距離が, 相対的なスクロールピッ

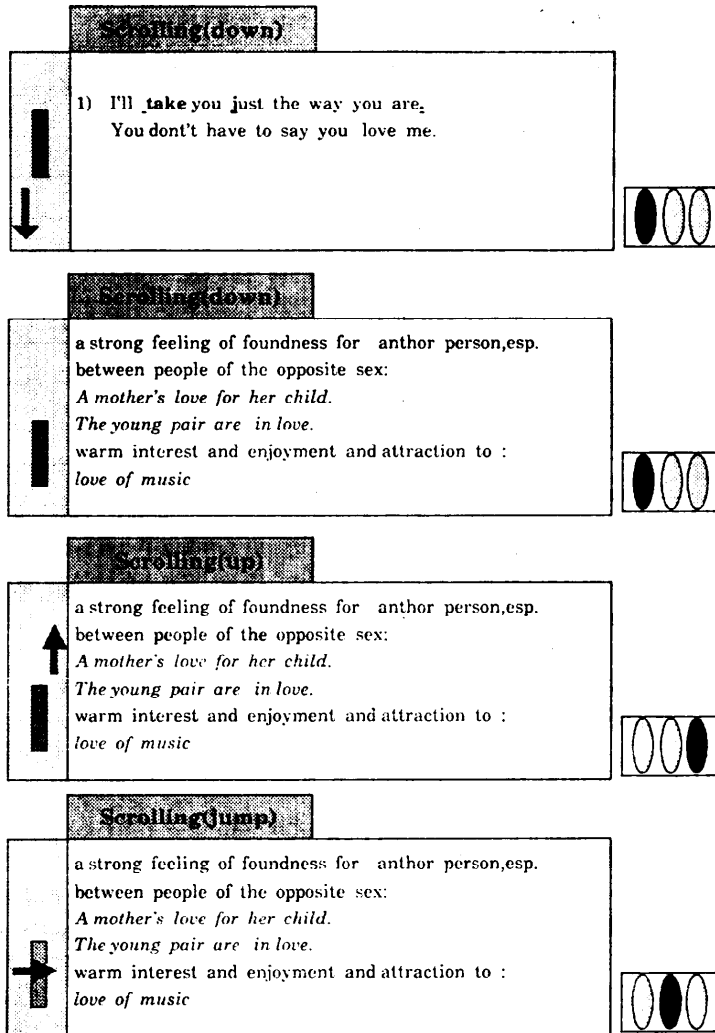


図-11 Scrolling in Smalltalk-80 Text Editing

チ(幅)となる。

◇ジャンプスクロール: スクロールバーの中央にカーソルをもってゆくと、スクロールバーを自在に移動でき、そのスクロールバーはマウスのカーソルの動きに追従し、任意のテキスト位置に、飛び越してスクロールが可能である。

4. プログラミング環境に適応したテキスト編集・フォーマット機能

前章までは、一般的なテキストを対象とするエディタについて言及したが、Smalltalk-80 のプログラミング環境という性格に合致した、プログラムソースの

テキスト編集・フォーマット機能に関して述べる。とくに、プログラミングプロセス上に重要な役割を提供する System Browser の Code View を例にとり、プログラミング環境向きの機能について説明する。

1) テンプレート

Smalltalk-80 の世界では、“Everything is object” という考え方で、一貫したプログラミングスタイルが暗黙に成立している。こうした、Smalltalk-80 プログラミングスタイルをスムーズにナビゲーションする役割を果たすのが、System Browser の機能である。そのために、この System Browser 上で、クラスの定

表-1 Control Keys Facilities of Smalltalk-80 Editing

Control Keys	Meaning	Example
ctrl t	ifTrue:	1 < 2 (ctrl t) ifTrue:
ctrl f	ifFalse:	1 < 2 (ctrl f) ifFalse:
ctrl -	make undelline	'abcd' put: \$e
ctrl b (ctrl shift b)	make bold face	'abcd' put: \$e
ctrl w	delete selectedtext	'abc' put: \$e
ctrl	[selectedblock]	'abc' do: [each] each asAscii]
ctrl ((selectedmessage)	'abc' put: (name next)
ctrl <	<primitives >	< primitive: 115 >
ctrl '	' strings '	'abc' put: \$e
ctrl "	' comments '	'abc' put: \$e "puting \$e"

義・変更, クラスカテゴリの追加・変更, メソッドカテゴリの追加・変更, メソッドの定義・変更などは, まず, 最初は, その定義の方法の記述が, テンプレートとして, 表示される. プログラマは, そのテンプレートに意図する情報を書き加え, それをシステムに登録することによって作業を進める.

2) つづりの訂正

メッセージを評価・実行する場合 (doit, printit, accept) に, そのメッセージを構成する word (オブジェクト名, メソッド名) に対して, すでに登録 (定義) されている各種変数辞書, メソッド辞書の中で, 近似するつづりに訂正する機能が用意されている.

3) フォーマット

Smalltalk-80 には, 特定の言語の構文を絶えずチェックする構文向きエディタに類する機能は用意されていない. 原則的に, フリーフォーマットの記述ができる. ただし, 言語の構文および使用変数名がシステム認知された (accept) メソッドの記述に対しては, ポップアップメニューの format コマンドによって, 清書することが可能である.

4) プリント機能

Smalltalk-80 には, そのメッセージを評価・実行した結果 (オブジェクト) をウィンドウに表示する printit コマンドが用意されている. このメッセージの出力結果も, ウィンドウの中では, 編集可能なテキストとして扱われるので, プログラムの一部として利用することができる.

5) 構文向きタイピング機能

Smalltalk-80 のメッセージ表記に頻繁に使われる,

区切り記号や演算子 (セクタ) を効率良くタイピングする機能が用意されている (表-1 参照).

5. まとめ

本解説では, 統合プログラミング環境である Smalltalk-80 の利用者インタフェースを形成するのに不可欠の機能であるテキスト (プログラム) エディタについておもに利用者の視点でその編集インタラクションとテキスト (プログラム) エディタの利用者概念モデルについて述べた.

Smalltalk-80 には, ここで紹介した, テキスト編集機能の外にも, フォームやビットマップを編集するツールが要されている. また, Smalltalk-80 の拡張性を利用することによって, さらに, ある目的に適した編集機能や利用者インタフェースを容易に追加できる.

Smalltalk-80 環境に組み込まれたテキスト (プログラム) エディタ機能は, 利用者 (プログラマ) のソフトウェア設計, プログラミングという目的に至る手段としての数々の煩雑な作業を, 実際的な実利を伴う能率の改善と, それとの相乗効果としての, 心理的な負担の軽減をもたらすように配慮されている. こうした, 目的に至る手段としての煩雑な作業からの解放は, プログラマに創造的な意志の高揚をもたらし, 良い設計への確かな指針となる.

ソフトウェア工学華やかであるが, プログラマが対面する時間が長いテキスト (プログラム) エディタについて, もう少し, 本格的な研究が必要であると思われる. ソフトウェア工学上の新しい技法より, 身近な

プログラミングツールの改善に関心を向けることが、ソフトウェア生産性向上への確かな一歩であると思われる。プログラミングツールを考えることは、設計、ソフトウェア、プログラミングの本質を考えることにつながる。こうした、洗練されたプログラミングツールがあるソフトウェア方法論と融和した一つの姿が、Smalltalk-80 である。

謝辞 本稿をまとめるにあたり、貴重なお教示をいただいた慶応義塾大学情報科学研究所助教授原田賢一博士に深謝いたします。

参 考 文 献

- 1) 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 1100SIP, 情報処理, Vol. 25, No. 2, pp. 143-156 (1984).
- 2) Xerox PARC Learning Research Group: Special Issue on Smalltalk, Byte, Vol. 6, No. 8 (1981).
- 3) Goldberg, A.: Smalltalk-80: Interactive Programming Environment, Addison-Wesley (1984).
- 4) Goldberg, A. and Dave Robson: Smalltalk-80: The Language and Its Implementation, Addison-Wesley (1983).

(昭和 59 年 5 月 8 日受付)