

計算機誘導形構造エディタ

Computer Guided Structure Editor

ソフトウェアの大規模化と複雑化に伴い、その生産性と信頼性の向上が急務となっている。今まで、ソフトウェアの効率的開発を支援するため、その開発工程全般にわたる方法論、技法、ツールが多数開発されてきたが、現在では、これらを体系的に結合した統合プログラミング環境が最重要視されている。計算機誘導形構造エディタPARSEはそのユーザーインタフェースとなるもので、計算機が文法誤りのないプログラムの入力を誘導する方式によって、プログラマの習熟度に応じた効率的プログラミングを実現する。

本稿では、構造エディタPARSEの開発目的と機能、及び適用効果について述べる。

田中 厚*
岸本芳典*
中所武司**
森 清三***
増井光幸****

Atsushi Tanaka
Yoshinori Kishimoto
Takeshi Chūsho
Kiyomi Mori
Mitsuyuki Masui

1 緒 言

高度情報化社会への移行、OA(Office Automation)、FA(Factory Automation)の進展など、マイクロコンピュータをはじめとする計算機の急速な普及とともに、これらの計算機を制御するソフトウェアの早期開発と信頼性向上が急務となっている。これに伴い、必要とされるプログラマ数が増加し、要求されるプログラミング能力が高度化してきた。

ところが、最近になって、供給可能なプログラマ数の不足が顕著になってきたことから、新人プログラマの短期養成による早期戦力化が叫ばれ始めている。しかし、従来のプログラミングシステムを用いたのでは、初心者に対して期待するほどの効果が上がらないことが認識されてきた。

計算機誘導形構造エディタPARSE(Production and Reduction Structure Editor)は、この問題を解決するために開発した新しいプログラミングシステム¹⁾である。従来のテキストエディタでは、入力したプログラムの文法誤りがなくなるまで修正を繰り返さねばならなかったが、PARSEでは対象とするプログラミング言語の文法を内蔵し、これに基づいて計算機が文法誤りのないプログラムを誘導するため、プログラマの習熟度に応じて効率的なプログラミングを実現する。

本論文では、計算機誘導形構造エディタPARSEの設計思想、基本機能及び適用効果について述べる。

- (1) エディタに、プログラミング技法の適用促進を支援する機能を導入する(技法への適応)。
- (2) エディタの機能を、それが対象とするプログラムを記述した言語に適したものにする(言語への適応)。
- (3) エディタの操作方法を、それが対象とするユーザーのレベルに適合したものにする(ユーザーへの適応)。
- 以上の基本方針を、それぞれプログラミング技法の誘導、プログラムへの視点の高度化、習熟度に応じた操作方式により実現する。

2.2 基本方針実現に当たっての要件

- (1) プログラミング技法の誘導
- プログラミング技法(例えば構造化プログラミング技法)は、一般にプログラミング言語には依存させないようにして抽象的に説明されるため、これらの技法を具体的なプログラムの記述に適用することが難しい。そこで、プログラミング技法の適用方法と対象とするプログラミング言語への変換方法を誘導する。
- (2) プログラムへの視点の高度化
- 従来のテキストエディタは、プログラムを単なる文字列とみなし、文字単位あるいはまとまった文字列から成る行単位

2 構造エディタPARSEの設計思想

2.1 基本方針

プログラミングには、プログラムを入力したり修正するためのエディタが不可欠である。しかし、従来のテキストエディタには、ユーザーからみて次のような問題点があった。

(1) ツールに必要なプログラミング技法の支援が全く考慮されておらず、一般のワードプロセッサ程度の単純な機能に限られている。

(2) プログラミング言語に依存しないようにして汎用性をもたせているため、機能的に低水準である。

(3) 熟練者を対象として設計されているため、操作方法が難しい。

これらが、プログラマの早期戦力化とプログラミングの効率向上を阻害する原因となっていた(図1参照)。

これらの問題に対処するためには、次の方針が有効である。

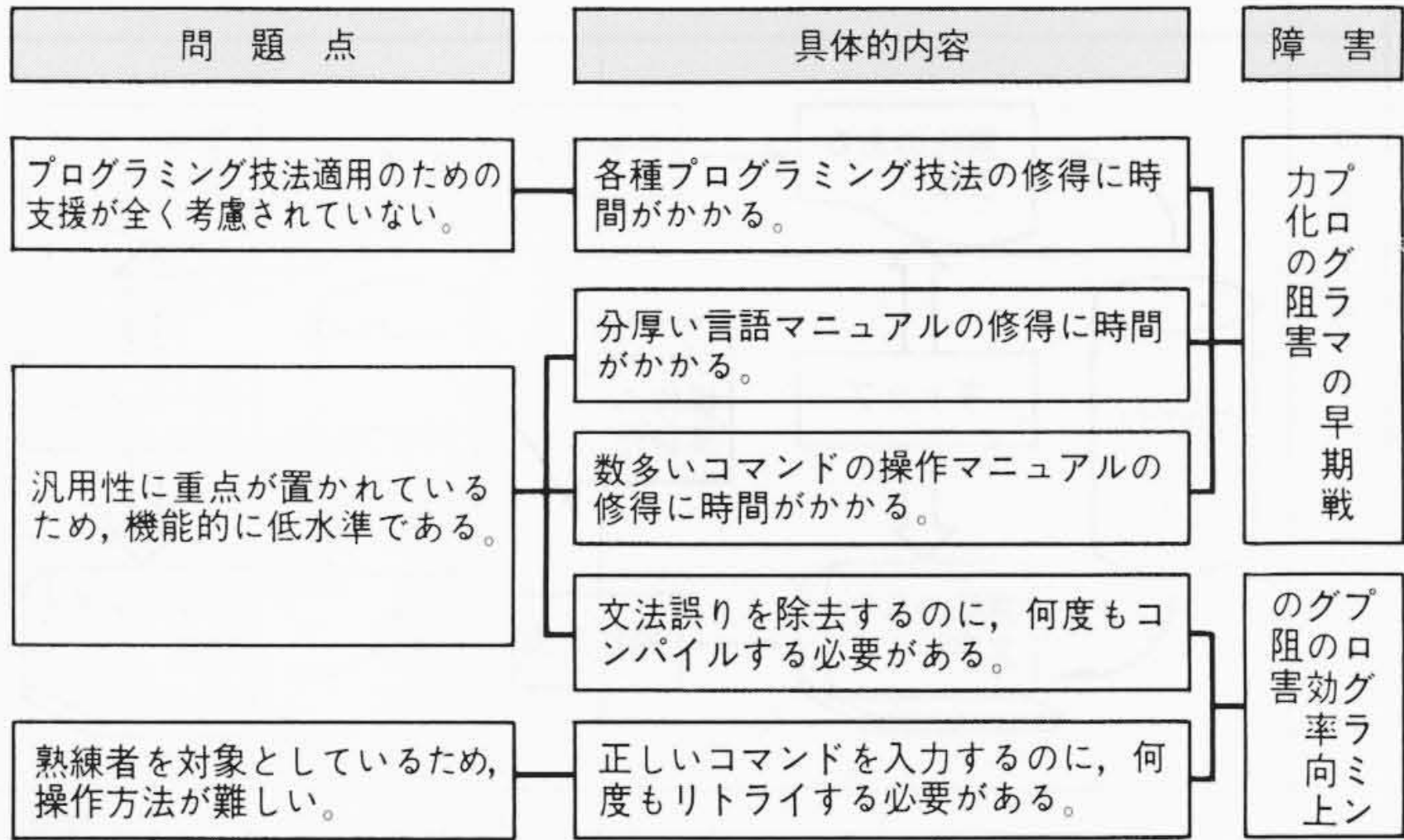


図1 従来のテキストエディタの問題点 従来は、エディタを使用する前に修得しなければならないことが多く、しかも使用開始後もその操作方法に慣れるまでは何度もリトライさせられた。

* 日立製作所システム開発研究所

** 日立製作所システム開発研究所 工学博士

*** 日立製作所大みか工場

**** 日立製作所戸塚工場

のテキスト編集を行なうものであった。例えば、「繰返し文 (REPEAT~END)を削除する」ためには、「“REPEAT”のある行から、対応する“END”のある行までを削除する」と操作しなければならない。このように、プログラマはプログラムを操作対象とするときはそれを文字とみる一方、思考対象とするときは、それを意味のある文章とみる。すなわち、書くレベルと読むレベルの間に大きなギャップがあり、これが単にプログラミング効率の低下ばかりでなく、誤り発生の原因になっている。そこでPARSEでは、言語の構文要素単位の操作を可能とすることにより、従来のエディタに関する意味的ギャップを除き、操作対象のレベルを思考対象のレベルにまで引き上げる(図2参照)。先の「繰返し文」の例では、行を意識させることなく、構文要素である「繰返し文」を指定するだけで削除できるようにする。

(3) 習熟度に応じた操作方式

プログラムの開発経験あるいはツールの利用経験の浅い初心者からベテランプログラマである熟練者まで、その学習曲線に応じた高効率プログラミングを実現するため、ユーザーの習熟度に適した柔軟な操作方式(ユーザーインタフェース)を与える。

2.3 基本機能

以上述べた基本方針と実現要件から、構造エディタPARSEには、図3に示す基本機能を設けた。

3 構造エディタPARSEの基本機能

3.1 段階的詳細化機能

E. J. Dijkstra教授の提唱する「一度に一つの決定」の原則に沿った段階的詳細化²⁾によるプログラミングを支援する。そのため、PARSEが扱うプログラムに、次のような「処理概要記述」を導入する。

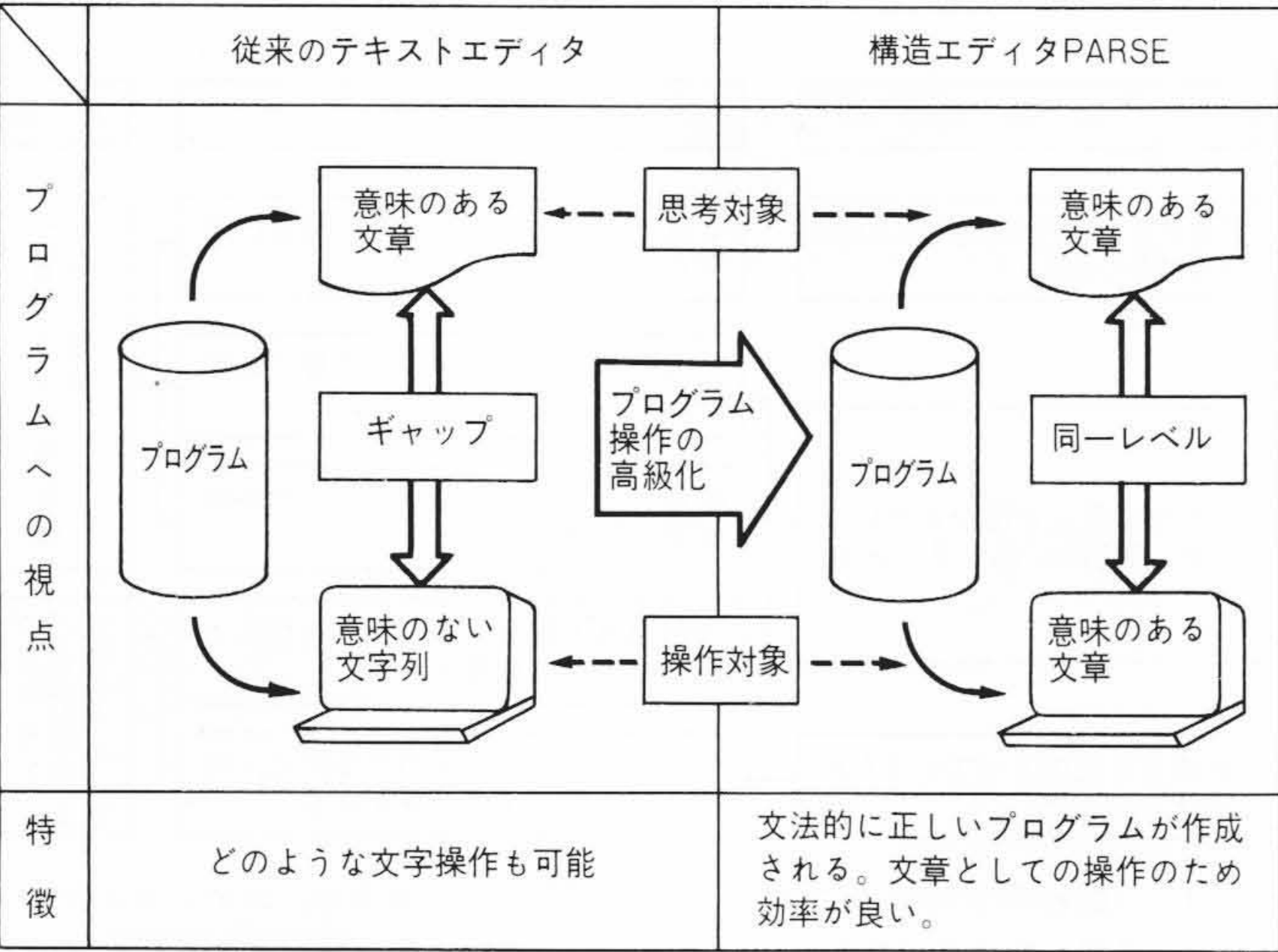
(1) 擬似文

処理の概要を、構文規則に制約されないで自由に記述できるようにする。

例：“データ ヲ クワエル”

(2) 手続きコール文

プログラミング言語自身がもっている段階的詳細化支援機



注：略語説明
PARSE(Production and Reduction Structure Editor)

図2 プログラムへの視点の高度化 PARSEでは、言語の構文要素単位の操作方法を提供することにより、操作対象のレベルを思考対象のレベルまで引き上げる。

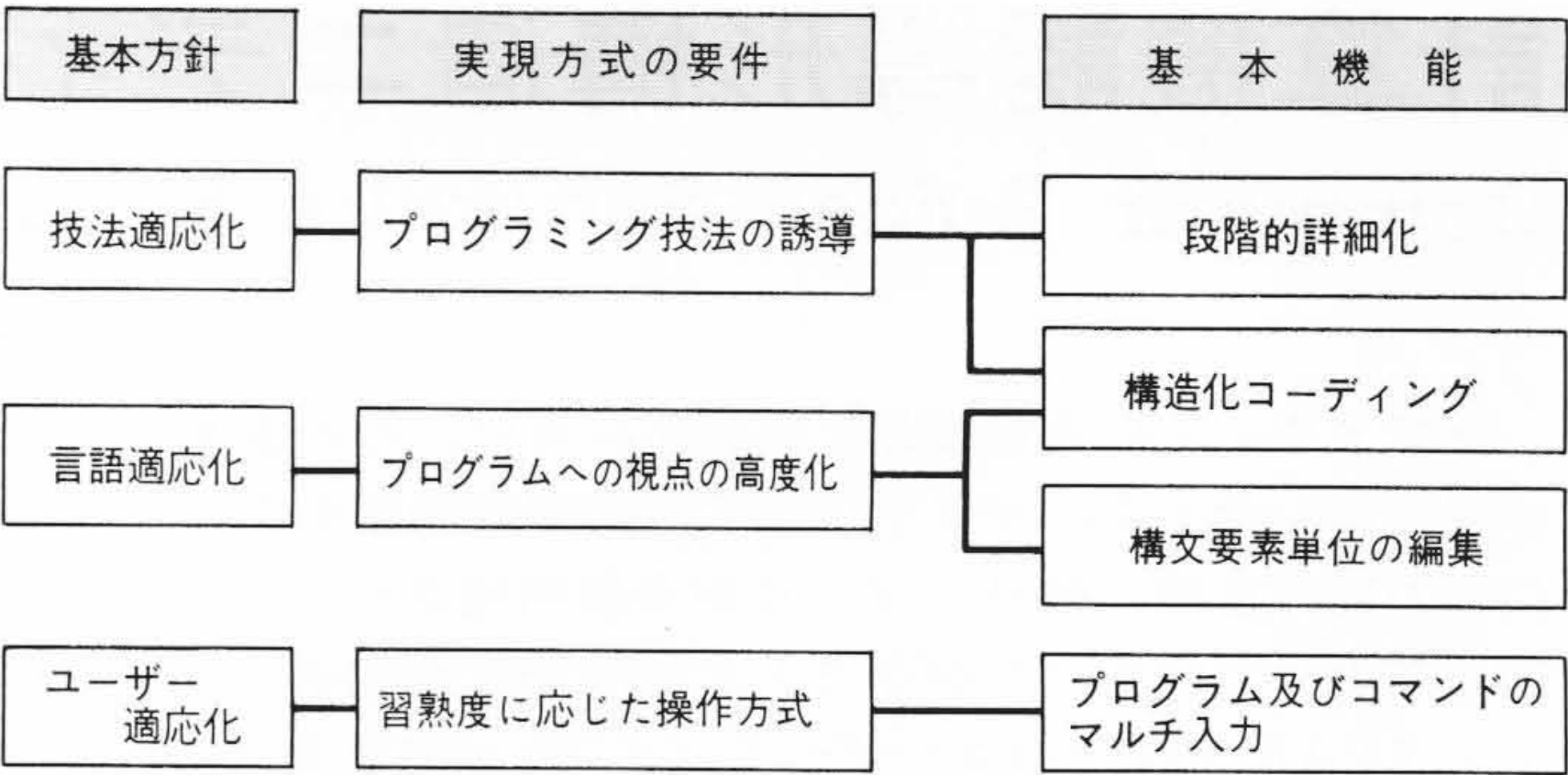


図3 構造エディタPARSEの設計方針 初心者から熟練者まで、その習熟度に応じた使い方ができるように基本機能を設定している。

能であり、処理の概要を言語の構文規則に沿って記述できるようにする。

例：Add Data (S, A, B)

PARSEでは、これらの「処理概要記述」を用いて次のように段階的詳細化を行なう(図4参照)。

擬似文で記述した処理概要(図3の“データ ヲ クワエル”)の詳細化を要求すると、擬似文は対象とするプログラミング言語のコメントに変更され、同時に「文」の入力状態になって、処理概要の詳細が記述可能となる。

手続きコール文(図3のAdd Data (S, A, B))の詳細化を要求すると、その手続きの定義側の枠組みが自動表示され、その処理の詳細が記述可能となる。この手続き本体の記述を終了すると詳細化要求前の画面に戻り、引き続き別の「処理概要記述」の段階的詳細化が可能となる。

このように、従来は設計者が机上で行なっていた段階的詳細化を、PARSEでは計算機が誘導してくれるため、経験の浅いプログラマでも段階的詳細化技法が適用できるとともに、詳細化のレベルがプログラム中にコメントとして保存されるため、ドキュメント性が向上する。

3.2 構造化コーディング機能

基本制御(選択, 反復, 複合)文を用いた構造化コーディングを支援する(図5参照)。そのため、入力可能な「文」として、先に示した擬似文, 手続きコール文のほかに、基本制御文の一覧をメニューとして表示する。ユーザーはメニューから一項目を選択するだけで、希望する文が入力できる。このとき、PARSEは選択された文の種別に応じてその枠組みを自動生成し、清書(プリティプリンティング)を行なって表示する。

このように、初心者でも文法の複雑な基本制御文の入力が簡単に行なえるため、文法誤りがなくなり、キー操作数が削減されるとともに、従来はプログラマに任されていた字下げをエディタが自動的に行なうことにより、ソースプログラムの書式の標準化が図れる。

3.3 構文要素単位の編集機能

「宣言文」, 「実行文」, 「条件式」などの構文要素単位のプログラム編集を可能にするため、ユーザーが入力すべき部分(「利用者入力部分」と呼ぶ。)と、その典型的な記述形式を定義した型紙(「テンプレート」と呼ぶ。)を導入する。

以下では、プログラム編集機能を「入力」と「修正」に分けて説明する。

(1) プログラム入力

画面中のソースプログラムには「利用者入力部分」(現在注

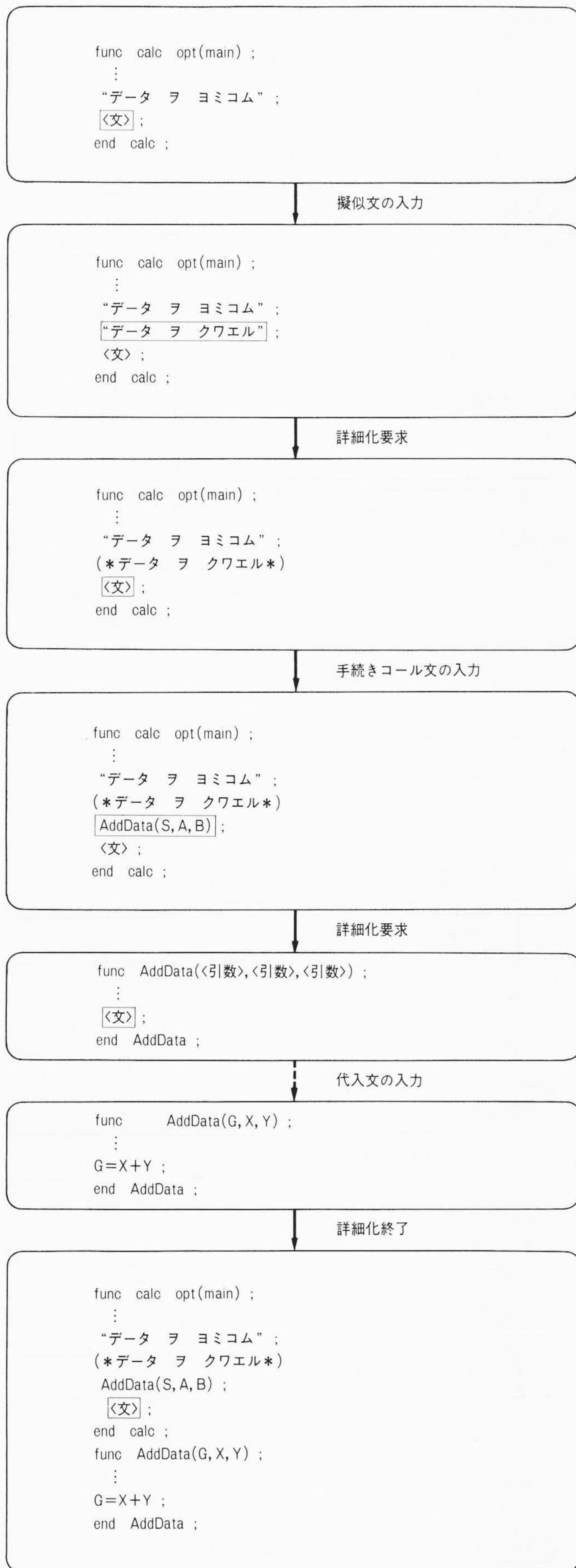


図4 構造エディタPARSEによる段階的詳細化 擬似文あるいは手続きコール文に対して詳細化を要求すると、PARSEが次に行なうべき作業内容を誘導する。なお、 囲みは現在の注目点を表わす。

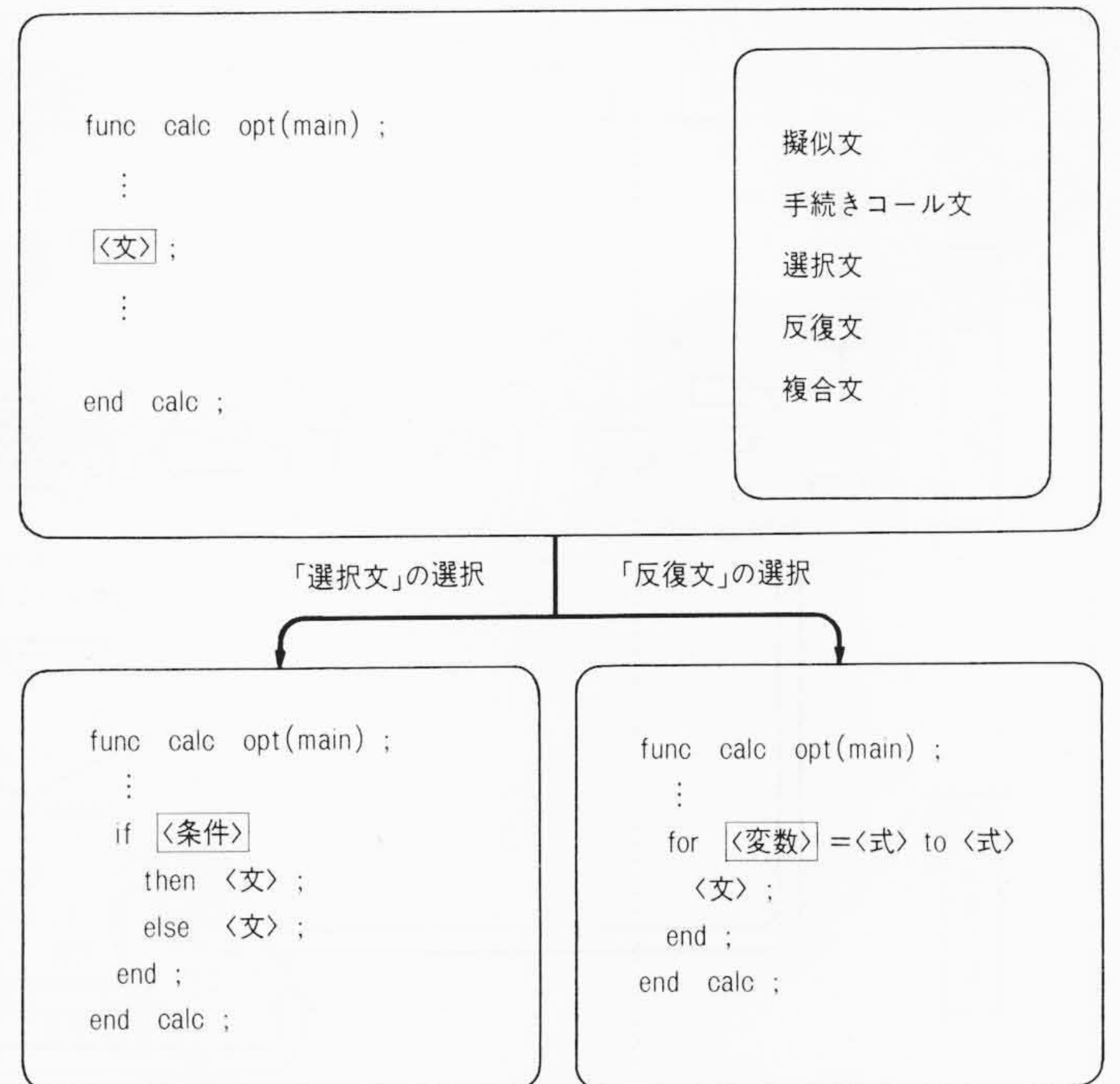


図5 構造エディタPARSEによる構造化コーディング 入力したい基本制御文(選択文、反復文、複合文)を選択するだけで希望する文が入力でき、PARSEが自動的に清書表示を行なう。

目している「利用者入力部分」を「注目点」と呼ぶ。)が表示され、メニューには入力可能な「テンプレート」の一覧が表示される。ユーザーは入力したい「テンプレート」を選択すると、「注目点」は選択された「テンプレート」で置換される。この繰返しにより、プログラムを入力する。

(2) プログラム修正

プログラム修正は構文要素単位に行なえるようにして、修正時のエラー誘発を防止する。修正用機能としては、挿入、削除、移動、複写、交換、探索を設けた(図6参照)。

これらの機能により、従来は分厚いマニュアルを見ながら行なっていたコーディングが、計算機の誘導により正確な文

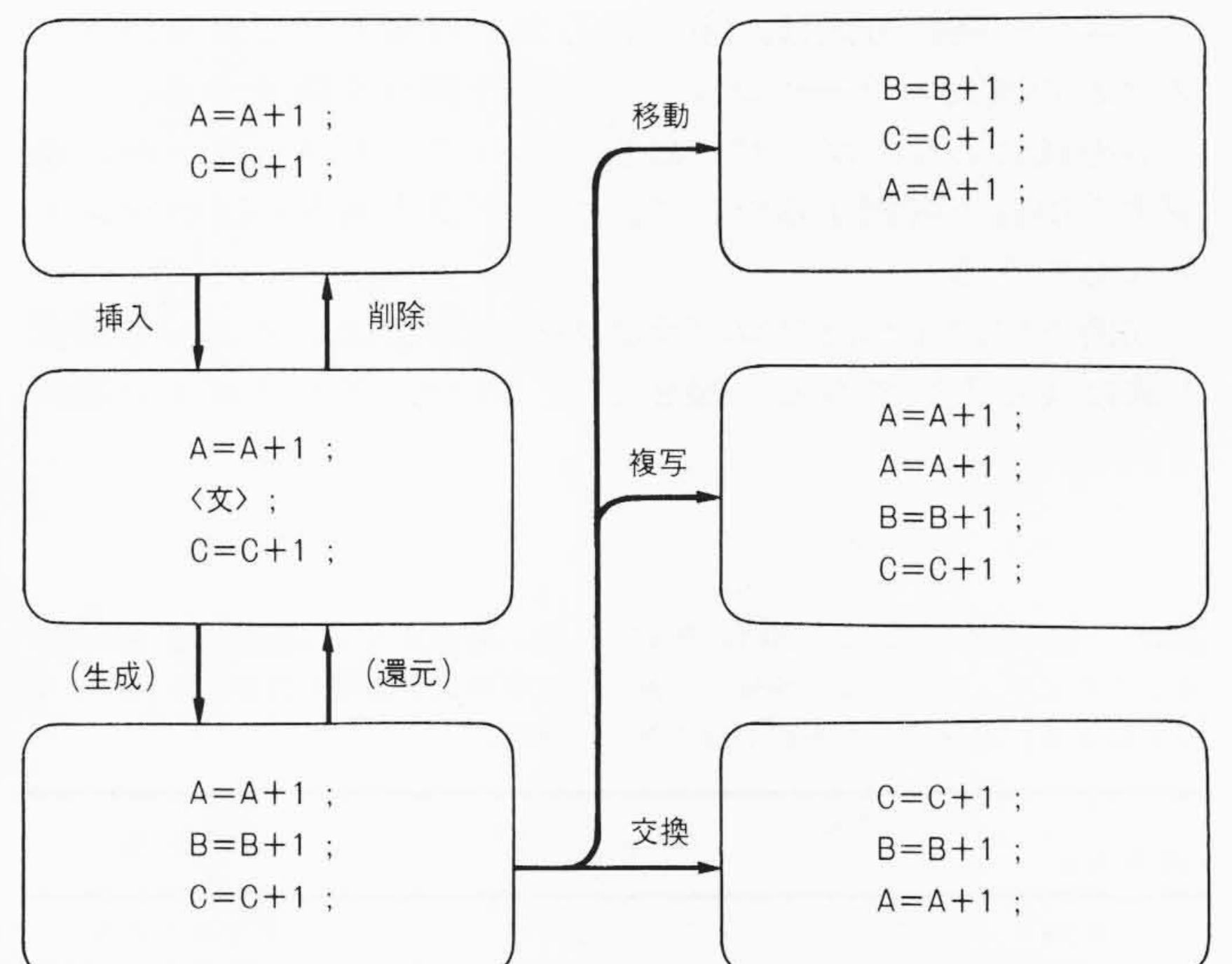


図6 構造エディタPARSEによる構文要素単位の編集 コマンドを入力し、修正対象位置をカーソルで指定する。修正後のプログラムは、入力のとときと同様に常に文法的に正しいことが保証される。

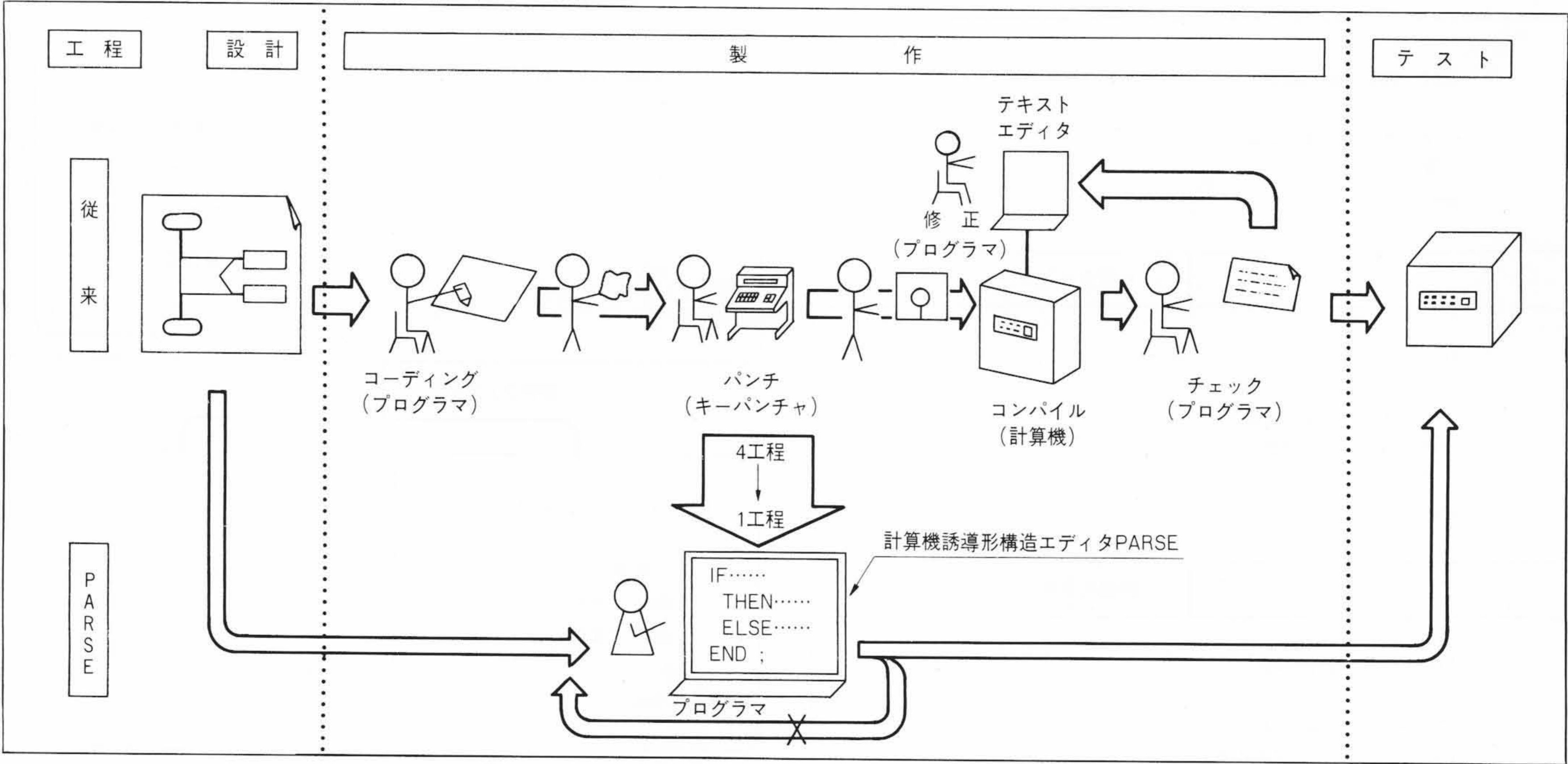


図 7 構造エディタ PARSE を用いたプログラム開発方法の特徴 従来はマニュアルを見ながら机で行なっていたプログラムの記述が、PARSE では計算機の誘導により行なうことができる。また、従来コンパイラを用いて行なっていた文法チェックを、PARSE ではプログラムの入力時に実施するため、エラーの発見が早くなる。

法を知らないユーザーにも簡単に行なえとともに、編集されたプログラムは常に文法的に正しいことが保証されるため、プログラムの信頼性が向上する。

以上述べた構造エディタ PARSE の機能を用いたプログラム開発方式の特徴を図 7 に示す。このように、PARSE では、プログラムの段階的詳細化から文法エラーの除去までを、計算機の支援により一貫した作業工程で行なうことができる。

3.4 プログラム及びコマンドのマルチ入力

初心者から熟練者まで、その習熟度に応じた柔軟な操作方式を提供するため、プログラム及びコマンドは、それぞれメニュー選択方式とテキスト直接入力方式の両方式で入力できるようにした(表 1 参照)。

(1) メニュー選択方式

メニュー選択方式は、主に初心者を対象としており、プログラムの構文エラーやコマンドの操作誤りを防止する。

本方式により、ユーザーは表示されているメニューから希望する項目を選択するだけで、プログラムあるいはコマンドを入力できる。

前節までに述べたプログラムの入力方法は、メニュー選択方式によるものである。図 8 には、コマンドのメニュー選択方式の例を示す。

表 1 習熟度に応じた操作方式 初心者はメニュー選択方式を利用することにより、誤りのない操作を、熟練者はテキスト直接入力方式を利用することにより、高速な操作を行なうことができる。

使用者	初 心 者	熟 練 者
作業内容		
コマンド入力	メニュー選択	テキスト入力
プログラム入力	構文テンプレート表示 + メニュー選択	テキスト入力

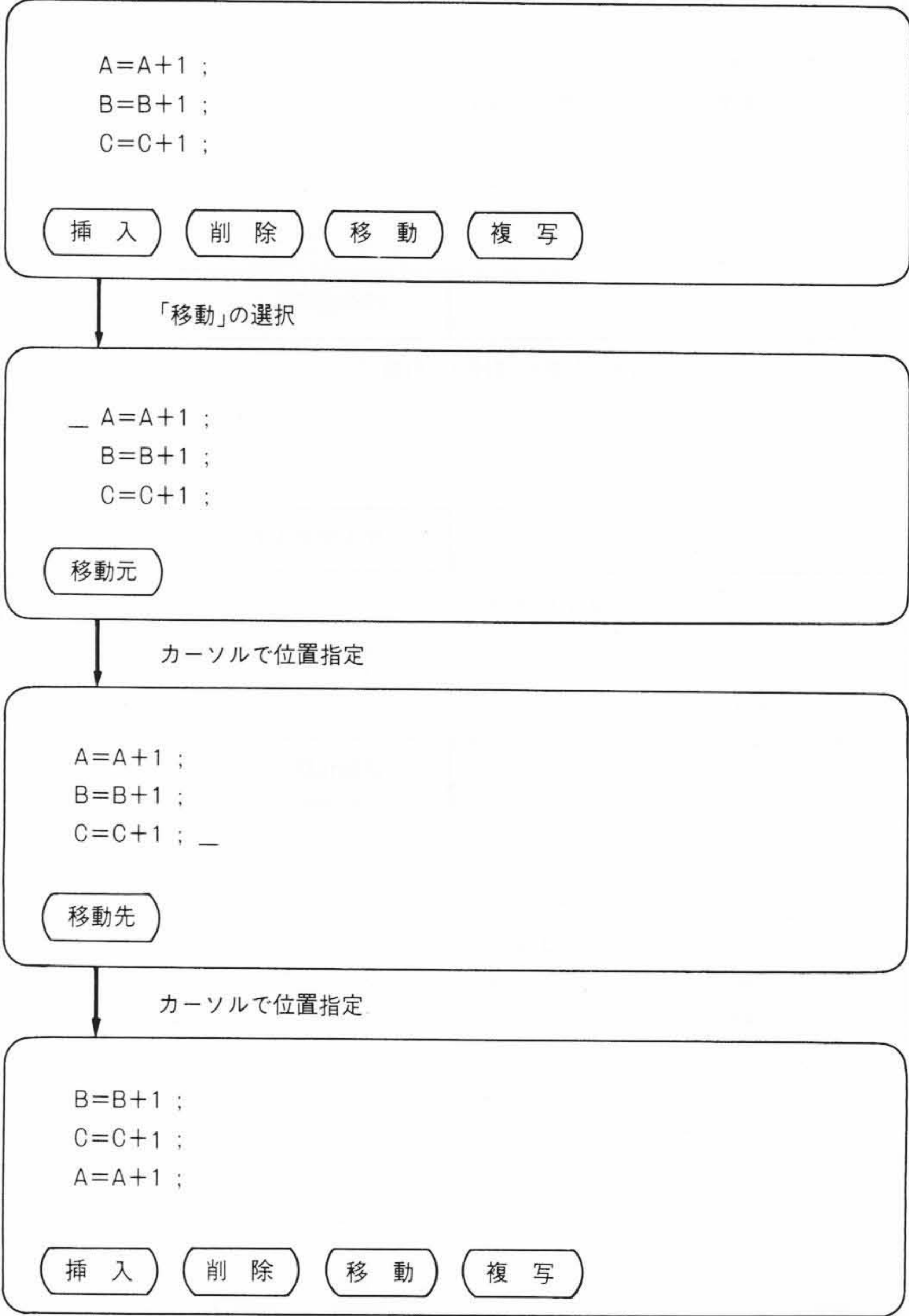


図 8 メニュー選択方式によるコマンドの入力誘導 PARSE は、入力可能なコマンドあるいはオペランドを表示し入力を誘導するので、ユーザーは希望する項目を選択するだけでよい。

(2) テキスト直接入力方式

メニュー選択方式では、慣れてくるとその操作が煩わしくなる。そこで、熟練者向けのテキスト直接入力方式を導入し、プログラマの習熟度に応じて、メニュー選択方式との使い分けができるようにした。プログラムあるいはコマンドの構文を熟知しているユーザーは、テキストで直接入力できる。入力されたテキストは、文法的な正しさを保証するため即時に文法チェックを行ない、エラーがあれば修正の後再度入力できる。

3.5 その他の主要機能

- (1) 直前のプログラム操作の取消し
- (2) プログラムのファイルへの退避とシステムの終了
- (3) 手続き単位の画面スクロール
- (4) 構文要素単位の「注目点」移動

構造エディタPARSEの基本的なコマンドの一覧を表2に示す。

3.6 拡張機能

今までに述べた基本機能のほかに、構造エディタPARSEの使用環境に合わせて、Auto-DS(Automated Documentation Support Tool)コメントの誘導機能を導入した。すなわち、マイクロコンピュータ用ソフトドキュメント自動生成システムのAuto-DSが定めるコメントの記入規則を内蔵させ、コメントの入力を誘導するようにした。図9にその例を示す。

本機能により、すべてのユーザーに対してコメントの記入が促進されるため、コメント形式の標準化と共通化が図れるとともに、ソースプログラムのドキュメント性が向上する。

4 機能比較

従来方式によるフルスクリーン形式のテキストエディタと構造エディタPARSEとの機能比較を、表3に示す。

従来のエディタに比べ、PARSEは機能性・操作性ともに勝っていることが分かる。

5 適用効果

現在、構造エディタPARSEは、標準PASCAL³⁾、制御計算

表2 構造エディタPARSEの基本コマンド PARSEのコマンドを組み合わせることによって、より複雑なプログラム編集を行なうことができる。

大分類	小分類	コマンド	機能
プログラム編集	プログラム入力	refine	擬似文の詳細化
		define	手続きコール文の詳細化
		produce	構文要素の詳細化(生成)
	プログラム修正	reduce	構文要素の抽象化(還元)
		insert	構文要素単位の挿入
		delete	構文要素単位の削除
		move	構文要素単位の移動
		copy	構文要素単位の複写
		exchange	構文要素単位の交換
		find	構文要素単位の探索
エディタ操作	コマンド入力	help	ヘルプメッセージの出力
		cancel	コマンド入力の途中取消し
		etc	コマンドメニューのスクロール
	画面制御	scroll	ソースプログラムのスクロール
		movecr	注目点の移動
	システム制御	save	プログラムのファイルへの退避
		end	システムの終了
		undo	直前操作の取消し

表3 従来方法との機能比較 従来のテキスト操作形エディタに比べ、構造エディタPARSEでは機能性及び操作性が格段に向上している。

比較項目		従来エディタ(VOS3 DESP)	構造エディタPARSE
1	段階的詳細化機能	×	自然語の処理概要からプログラムへの変換方法の誘導
2	プログラム作成方式	文字列入力	同左+ { 構文テンプレート表示 メニュー選択
3	プログラム修正単位	文字, 文字列	構文要素単位(文, 式, 名前など)
4	文法誤り検出機能	×	○
5	コマンド入力方式	テキスト入力	同左+ メニュー選択入力
6	画面表示形式	ユーザー入力形式	清書(プリティプリンティング)
7	誤操作回復機能	×	直前操作の取消し

注：略語説明 VOS3 (Virtual storage Operating System 3)

```
CALC : DO ;
/*****
/**** SPECIFICATIONS ;
/**** NAME      = CALC : CALCULATIONS ;
/**** FUNCTION  = ADD, SUB, MULT, DEV
/**** AUTHOR    = <AUTHOR_NAME>
/**** DATE      = <CREATE_DATE>
/**** END OF SPECIFICATIONS ;
/****
"Read Data"
"Add Data"
"Sub Data"
"Print Data"
END CALC ;
```

図9 Auto-DSコメントの入力誘導 マイクロコンピュータ用高級言語S-PL/Hを対象にしたPARSEでは、ソフトドキュメント自動生成システムAuto-DSの形式に合ったコメントの入力を誘導する。ユーザーは必要な部分を「穴埋め」していけばよい。

機用高級言語SPL(Software Production Language), マイクロコンピュータ用高級言語S-PL/H(Super Programming Language of Hitachi micro computer)の各言語が開発済みであり, HITAC MシリーズVOS 3(Virtual storage Operating System 3)上で社内ツールとして稼動中である。

これらの構造エディタPARSEの適用により, 次に述べる効果が得られる。

(1) プログラム作成工数の低減

段階的詳細化機能と構文要素単位のプログラム編集機能により, 従来方式に比べて端末操作数, 端末作業時間, プログラム記述と文法誤り修正工数を大幅に改善できる。

(2) ソフトウェアの品質向上

構造化コーディング機能と構文要素単位のプログラム編集機能により, プログラムの構造化が徹底され文法誤りの混入が防止されるため, ソフトウェアの品質向上が図れる。

(3) プログラムの保守と再利用の促進

段階的詳細化技法に基づくプログラム開発が励行されるため, 概略仕様とプログラムとの対応が明確化しドキュメントが標準化され, プログラムの保守と再利用が促進される。

(4) 初心者短期養成と早期戦力化

プログラミング技法, プログラミング言語及びエディタの使用方法が誘導されるため, 初心者短期養成と早期戦力化を図ることができる。

6 結 言

構造エディタPARSEは, ソフトウェアの段階的詳細化とプ

ログラミング言語の構文要素単位の編集を支援するプログラムエディタで, プログラミングの初心者から熟練者まで, その習熟度に応じた操作を可能としている。

今後は, PARSEを中心に, テストデバッガやコードレビューツールなどを有機的に結合し, 機能性及び操作性を更に向上させた言語適応形統合プログラミング環境の構築を推進する予定である。

参考文献

- 1) T. Chusho, et al.: A Language-adaptive Programming Environment Based on a Program Analyzer and a Structure Editor, 9th World Computer Congress, IFIP '83
- 2) E. W. Dijkstra: Notes on Structured Programming, Academic Press, London and New York, 1~82(1972)
- 3) N. Wirth: Programming Language PASCAL, Acta Informatica 1, 1, 35~63(1971)
- 4) 林, 外: 制御用トップダウンストラクチャードプログラミング言語-SPL-, 日立評論, 60, 235~240(昭53-3)
- 5) 神野, 外: 16ビットマイクロコンピュータ用高級言語S-PL/Hの開発, 情報処理学会マイクロコンピュータ研究会資料, 17, 3(昭56-7)
- 6) 橋本, 外: マイクロコンピュータ用ソフトドキュメント自動生成システム"Auto-DS", 日立評論, 68, 5, 361~364(昭61-5)

論文抄録

日本文入力方式評価法の研究

日立製作所 中山 剛・黒須正明

情報処理学会誌 26-11, 1390~1397 (昭60-11)

日本文入力は, 漢字の字数が数千あるため, どのようにすればユーザーに負担をかけずに, しかも効率的に入力できるかが問題であった。この問題をユーザーの入力操作をも含めて定量的に検討し, 今後の日本文入力方式のあるべき姿を考えるために, 仮名及び英字キーボードによる, 数種類の日本文入力方式について長期の入力実験を行ない, その結果を説明するモデルを構成した。

入力実験では(A)仮名入力表示選択方式, (B)仮名2ストローク記憶コード方式, (C)仮名漢字変換方式, (D)ローマ字入力表示選択方式, (E)英字3ストローク記憶コード方式及び(F)ローマ字仮名漢字変換方式の6方式を取り上げた。実験は仮名グループ5人, 英字グループ6人の計11人の女子大学生を被験者とし, 約1年間にわたって行なった。両グループとも, タッチ打法による仮名又は英字けん盤の打けん基礎訓練から始まり,

20時間の訓練の後に, 記憶コードの学習とそれによる入力, 表示選択方式での入力及び仮名又はローマ字仮名漢字変換方式による入力を, ランダムな順序で30分単位で行なわせた。

次いでこの結果から, これらのデータを説明できるモデルを構築した。このモデルは一種の時間モデルで, 各入力方式の入力手続きを分析し, 各ステップで必要とする時間の推定値から, 入力全体に必要な時間を予測するものである。文節単位の仮名漢字変換方式に例をとると, まず文節ごとの読みを仮名キーを打けんして入力し, 変換キーを押す。正しい変換結果が得られる場合を正変換と呼ぶと, これに必要な時間は次のようにして求まる。まず読みの入力時間は, 読みの数を N とし, t 時間訓練後の仮名キー打けん時間を $\tau_k(t)$ とすると $N \times \tau_k(t)$ となる。次に変換キーを打けんする時間は $w \times \tau_k(t)$ とする。 w は変換キーの打

ん方法によって1~3の値をとる。変換キーを押してから画面に結果が表示されるまでの待ち時間は τ_p とする。次に, 変換結果が正しいかどうかをユーザーが確認する時間であるが, これは実験から, $Ae^{-\gamma t}$ の形をとり, $A=0.64$, $\gamma=1.2 \times 10^{-2}$ で近似できることが分かった。1文節の正変換に要する時間はこれらの和となり, 1字当たりの平均入力時間は, これを文節中の漢字と仮名文字の数で割った値に正変換率 P_c を乗じたものとなる。実際には, 第1位に正しい結果が得られず, 幾つかの候補の中から選ぶ多変換過程, これでも必要な結果が得られず, 単漢字入力モードで入力する誤変換訂正過程も考慮して, 仮名漢字変換入力方式の入力時間が予測される。

このようなモデルによれば, 設計段階で入力手続きと変換率, システム処理時間などを設定することにより, 事前にその入力方式での入力に要する時間を予測できる。