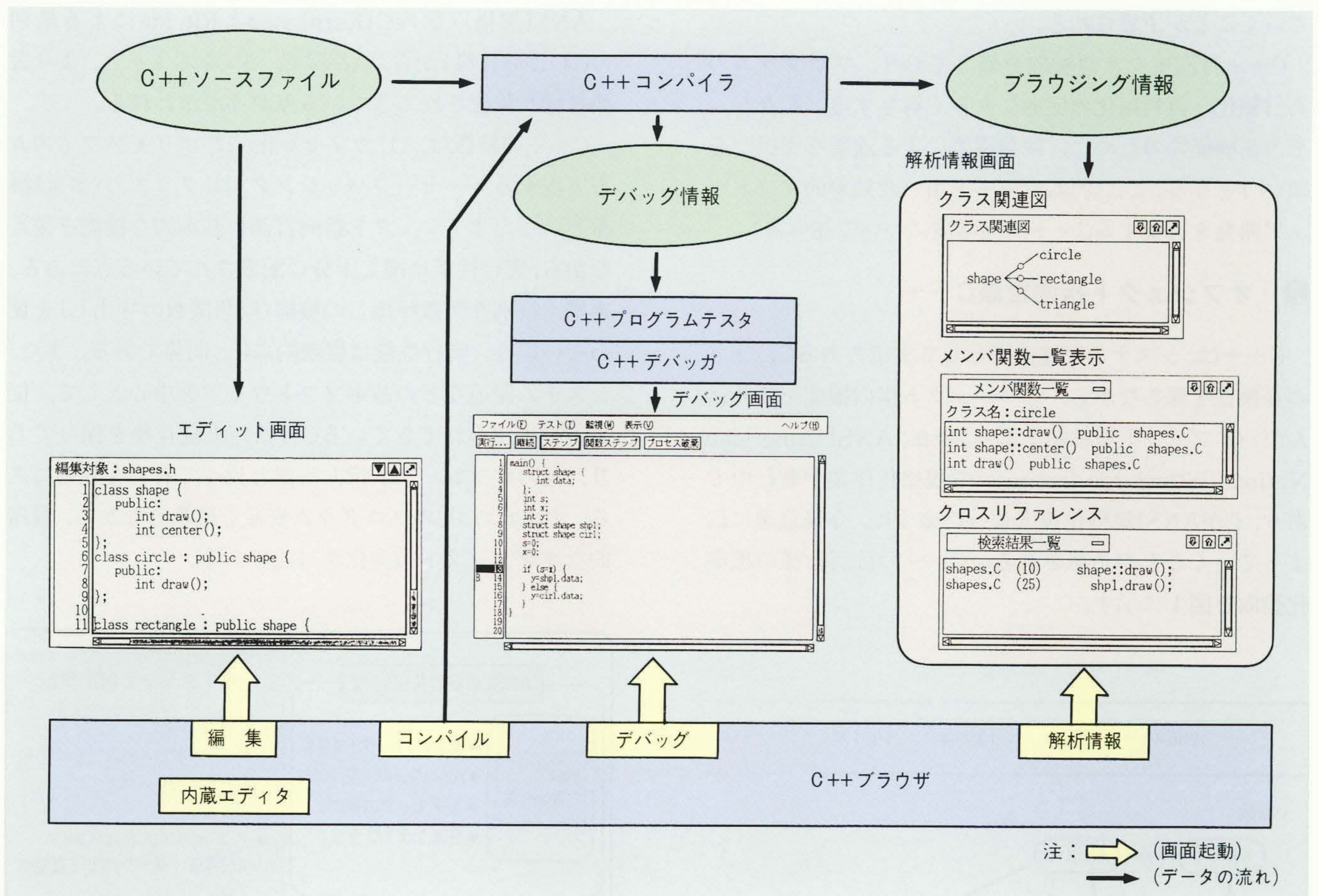


オブジェクト指向言語C++の開発支援環境

Developing Environment for Supporting Object-Oriented Language C++

森 教安* Noriyasu Mori
 長堀 正** Tadashi Nagahori
 友永佳津子** Kazuko Tomonaga
 江戸 研治*** Kenji Edo



C++開発環境 C++開発環境は、コンパイラ、デバッガおよびエディタを一体化した統合環境であり、オブジェクト指向プログラミングを支援する各種解析情報を提供できる。

オブジェクト指向を用いたプログラム開発には、差分プログラミングによる開発工数の削減や、カプセル化による高信頼化が期待されている。C++では、既存のプログラム資産となるクラスライブラリの構築・活用によって達成される。その際、クラスの解析情報を扱うブラウザが特に重要な役割を担う。

また、オブジェクト指向環境として望まれる他の側面として、ツール間の連携やGUI(Graphical User Interface)の活用による使い勝手の向上があ

げられる。そのためには、統一されたユーザーインタフェースと有機的なツールの結合が必要となる。

以上の諸点を考慮し、コンパイラ、デバッガ、エディタなどの各種ツール群を、ブラウザを用いて密連携させた統合環境であるC++開発環境を構築した。この開発環境を用いることで、Cのプログラム資産を継承しながら、段階的なオブジェクト指向化(CからC++への移行)を効率的に行うことができる。

* 日立製作所 システム開発研究所 ** 日立製作所 ソフトウェア開発本部 *** 日立コンピュータエンジニアリング株式会社 開発第2部

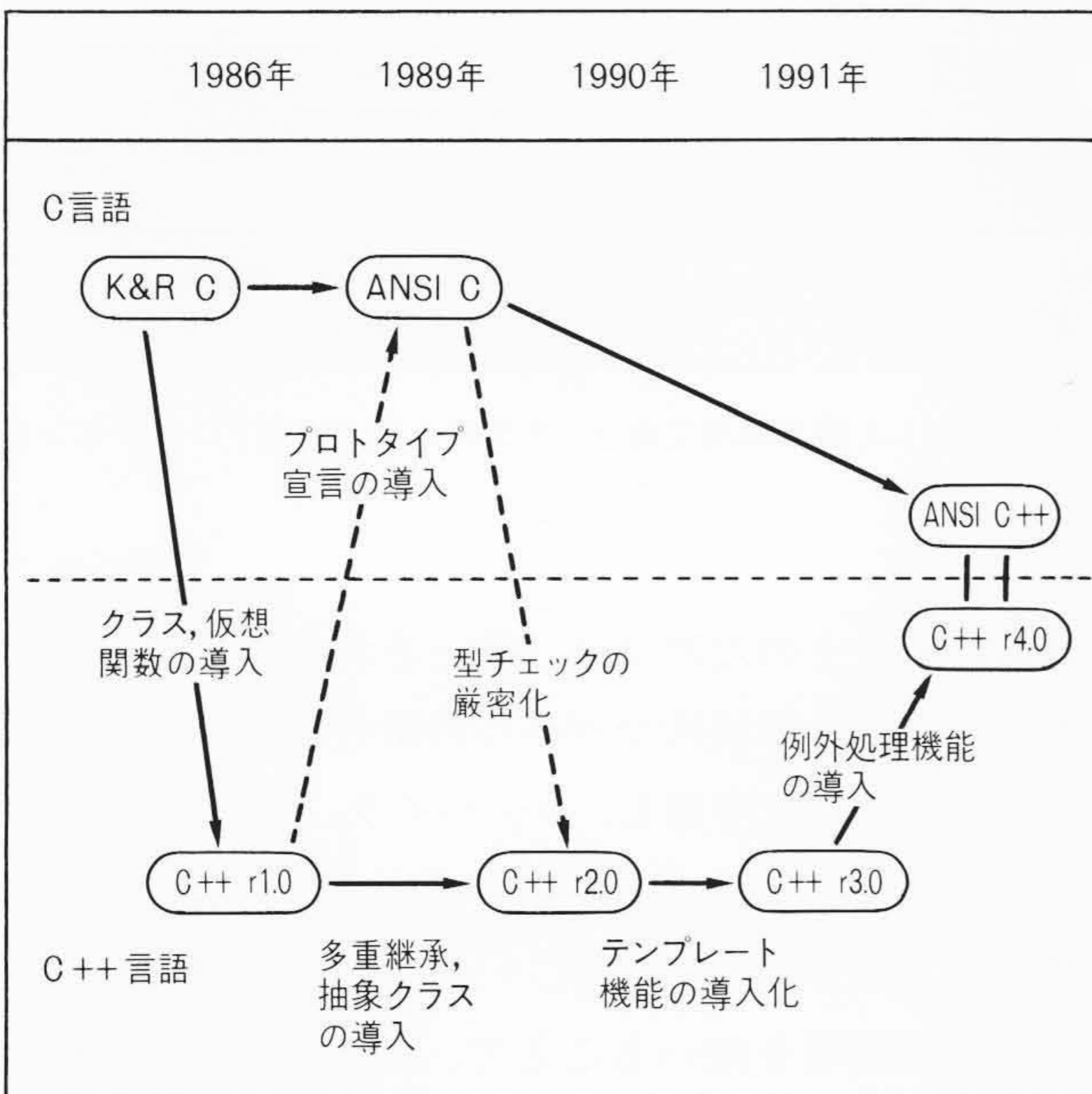
1 はじめに

近年、オブジェクト指向がソフトウェアの新しいパラダイムとして定着し、その有用性が認識されてくるとともに、実用化の時期を迎えつつある。オブジェクト指向言語として注目されているC++も、今後急速に普及していくことが予想される。

C++は高度な言語機能を備えており、プログラムの高信頼化、再利用化の促進に大きく寄与する。しかし、その高機能性のために、開発環境による適切な支援が必須(す)となる。ここでは、C++を用いた効率的ソフトウェア開発を実現するC++開発環境について述べる。

2 オブジェクト指向言語C++

C++は、システム記述言語として実績のあるC言語との互換性を保ちながら、オブジェクト指向機能を盛り込んだハイブリッド型言語である。現在、ANSI(American National Standards Institute)で規格化作業が進行中であり、CがANSI規格化後広まったように、今後急速に広まっていくことが予想される。C++の言語仕様の標準化動向を図1に示す。



注：略語説明
 K & R C (KernighanとRitchieによる最初のCの言語仕様)
 ANSI (American National Standards Institute)
 C++ r1.0 (C++のリリース1.0の言語仕様)

図1 C++言語仕様標準化の動向 C++はC言語の発展型言語として位置づけられ、現在ANSIによる規格化作業が進行中である。

図1のように、C++はCと独立したまったく新規の言語ではなく、Cをより発展させた言語と位置づけることができる。例えば、CのANSI規格化の段階でも、すでに一部の機能が取り込まれており、将来のC++への移行が考慮されている。CとC++の言語仕様上の拡張点を比較したものを図2に示す。

ANSI規格以前のC(KernighanとRitchieによる最初のCの言語仕様)から、ANSI C, さらにC++と、より言語仕様が強化されてきている点がかがわれる。

C++の特徴は、(1)カプセル化、(2)ポリモρφイズムを実現するメッセージパッシング、(3)クラスの(多重)継承といったオブジェクト指向言語の基本的な機能を備えながら、実行性能に関し十分に配慮されている点にある。実際、特別な関数呼出しの機構(仮想関数の呼出し)を使わない限り、実行性能は原理的にCと同等である。また、システム記述などの基本ソフトウェアを中心として、従来広範に使われてきているC言語の上位互換を保っており、他のオブジェクト指向言語と比べて有利な立場にある。すなわち、Cのプログラム資産を継承しながら、段階的なオブジェクト指向化が可能である。

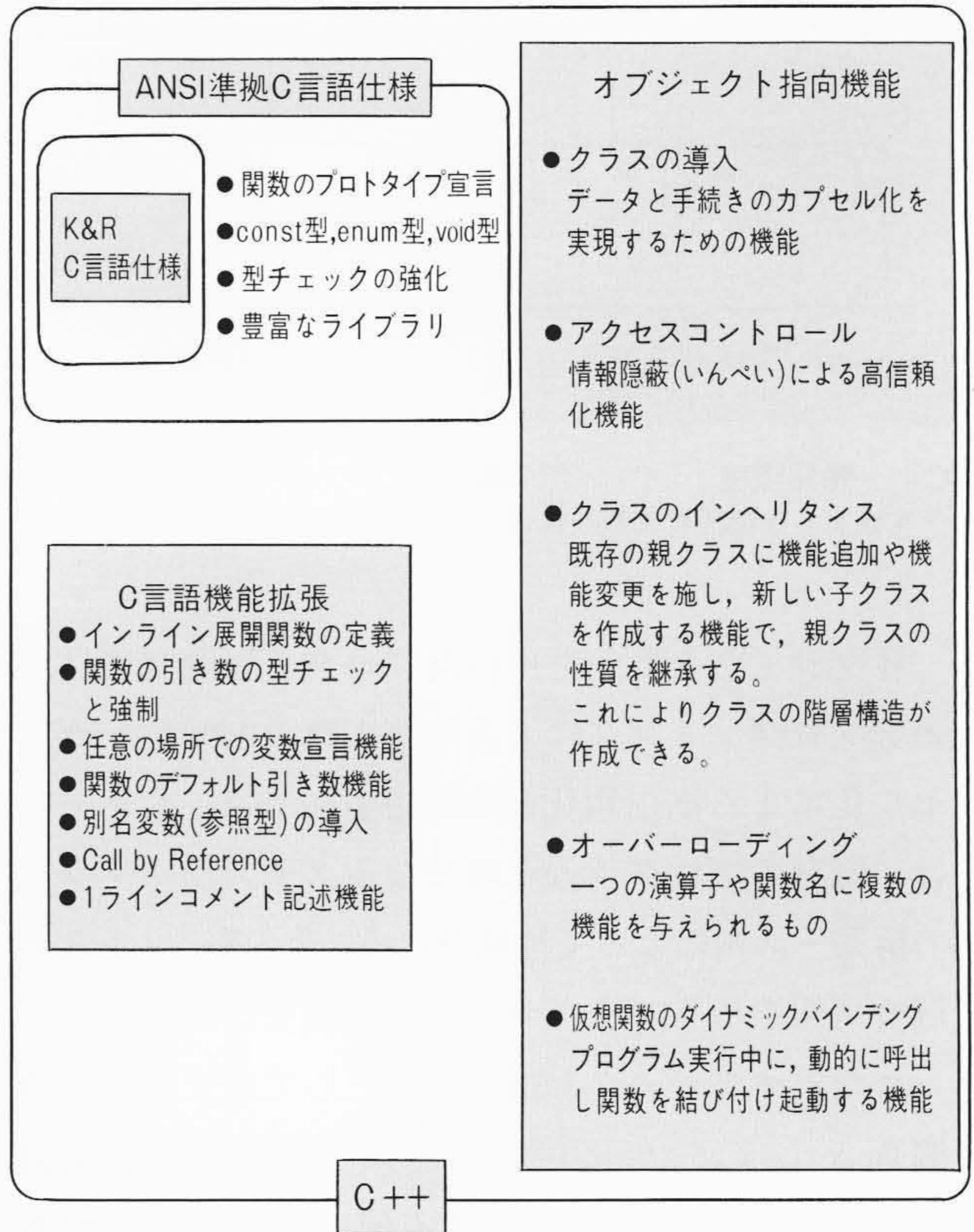


図2 C++の言語機能 C++はANSI Cの上位互換を保ち、C言語自身の機能拡張とオブジェクト指向のための機能拡張を行っている。

3 クラスライブラリによる再利用

C++の効果的活用のポイントは、クラスライブラリの構築にある。ここでは、クラスライブラリを用いたC++による差分プログラミングの方法について事例を交じえて述べる。

従業員情報を扱うクラスの定義例を図3に示す。今このクラスが既定義であるとき、所属するグループ員の一覧情報を付加したリストの作成が可能なマネージャ用クラスが新たに必要になったとする。例えばC言語などで記述されていた場合には、構造体にメンバを付加し、それを扱う関数も手直しする必要がある。C++の場合ならば、継承機能を用いることでその差分点だけをプログラムすることで対応できる。同図の(b)がそれに当たり、継承元のクラスEmployeeは変更を加える必要がない。すなわち、今まで実績を積み重ねた高信頼なソースコードを、そのまま再利用できるプログラム部品とみなせるようになる。

開発プログラムの種別は、開発部署ごとにそれぞれ千差万別である。しかし、特定の開発部署では仕様の一部だけが異なる類型的なプログラムの開発の機会が多い。また、ソーティングやリスト管理などのプログラムは、扱う対象が異なるにせよ、基本的機能は同様の場合が多い。これらの基本機能をすべての開発部署での汎(はん)

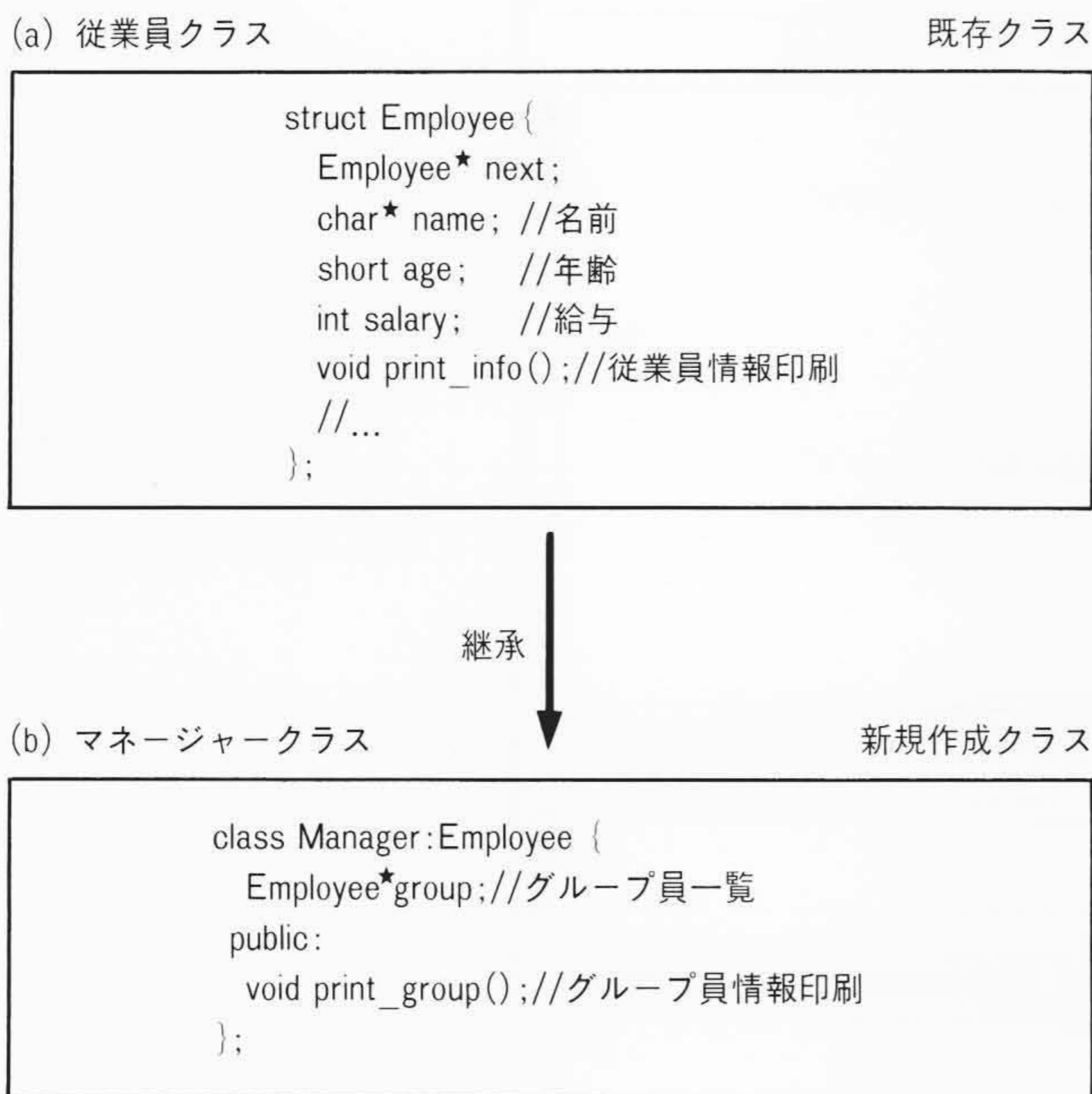


図3 C++における差分プログラミング例 既存クラスであるEmployee(従業員クラス)に変更を加えることなく、その性質を引き継いだ新規作成のManager(マネージャークラス)を作成できる。

用部品として持たせれば、開発効率・信頼性向上の効果が期待できる。これらの再利用可能な部品の集大成がクラスライブラリとなるわけである。すなわち、再利用の単位としてクラスを用い、継承関係によって系統的に整理統合したのがクラスライブラリである。ユーザー自身のクラスライブラリが構築できれば、継承関係は過去のプログラム資産のマップとなり、個々のクラスが差分プログラムによって再利用の単位となるわけである。

このようなクラスライブラリの構築を支援するC++開発環境について次章で述べる。

4 C++開発環境

C++によるプログラミング開発を支援するC++開発環境の全体像を図4に示す。

図4に示したようにC++開発環境は、言語処理系であるC++コンパイラおよびデバッガ、各種のクラスライブラリの閲覧やソースプログラムの編集を支援するC++ブラウザから成る。さらに、従来のCの資産も活用できるように、C言語にも対応している。また、C++のより高度な周辺ツールも将来的には予定されており、さまざまなツールとの協調による統合環境を目指している。

以下、C++コンパイラ・デバッガ、C++ブラウザの概略について順に述べる。

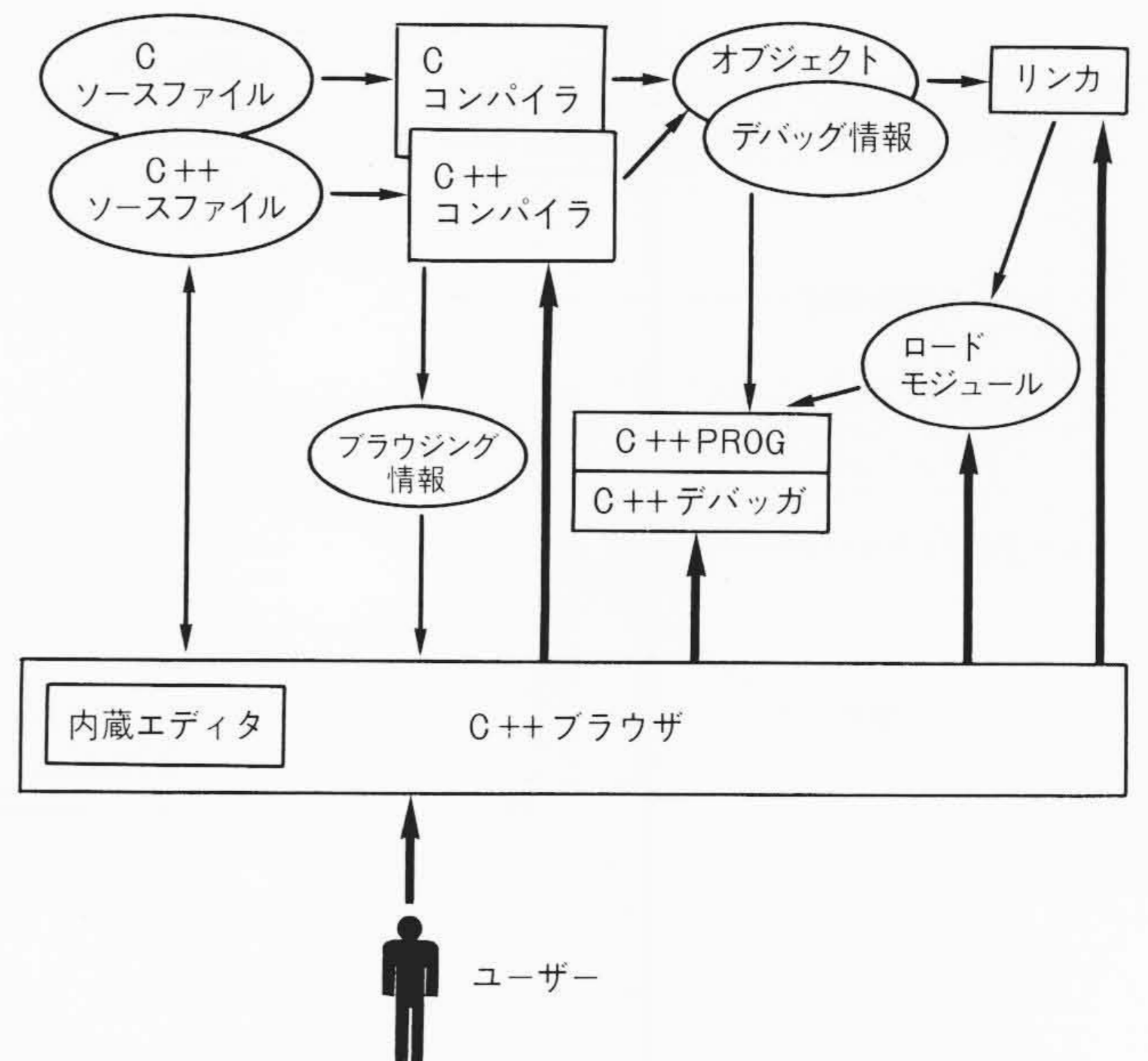


図4 C++開発環境の構成 C++による開発では、コンパイラ、デバッガと密接に連携し、エディタを内蔵したブラウザを通じ、Cと共存した環境で行える。

5 C++コンパイラ・デバッガ

5.1 最適化C++コンパイラ

C++が本格的に使用されるようになると、システムはますます大規模かつ複雑になる。これを処理するためには、高性能で使い勝手のよいコンパイラが必要不可欠となる。「最適化C++」は、強力なオブジェクト最適化機能と業界標準の言語仕様を持つコンパイラであり、ブラウザ、デバッガと連携して快適なC++プログラミング環境を提供する。最適化C++コンパイラの機能概要について以下に述べる。

(1) 業界標準のC++言語仕様サポート

C++の言語仕様はANSIで規格化の作業が進められている。その仕様内容は参考文献2)に記述されている。また、現在各社からすでに出荷されているC++コンパイラもほぼこれに沿っている。このため、最適化C++でも上記文献に従った言語仕様を採用して、他社のC++で書かれたプログラムとの互換性を維持するようにしている。将来ANSI規格化の時点では同規格に準拠する予定である。

(2) 強力な最適化機能

オブジェクト指向の概念に基づいてC++でプログラムを書くと、従来のCよりは非常に少ないコーディング

量で効率的なプログラム作成ができる。一方、コンパイラの生成コードへの依存度が大きくなるため、実行速度の向上には最適化の強化が従来にも増して重要である。

最適化C++では、不変式のループ外移動やループ展開を行うループ最適化、式の共通化などによる冗長性の削除、関数呼出しを削減する関数のインライン展開といったソースレベルでの最適化から、大域的レジスタ割当て、分岐最適化、命令スケジューリングのような機械語レベルの最適化まで網羅しており、効率的なオブジェクトコードの生成ができる。

5.2 C++PROG

C++PROGは、CまたはC++のプログラムのデバッグと構成管理を、GUI(Graphical User Interface)の環境で実現するツールである。機能概要について以下に述べる。

(1) プログラムテスト

プログラムテストはシンボリックデバッガxdbを、Motif^{*1)}、X-Window^{*2)}を用いて操作性を向上させたツールであり、C/C++プログラムのビジュアルなテスト

*1 Motifは、Open Software Foundation, Inc.の商標である。
*2 X-Windowは、米国MIT(マサチューセッツ工科大学)の商標である。

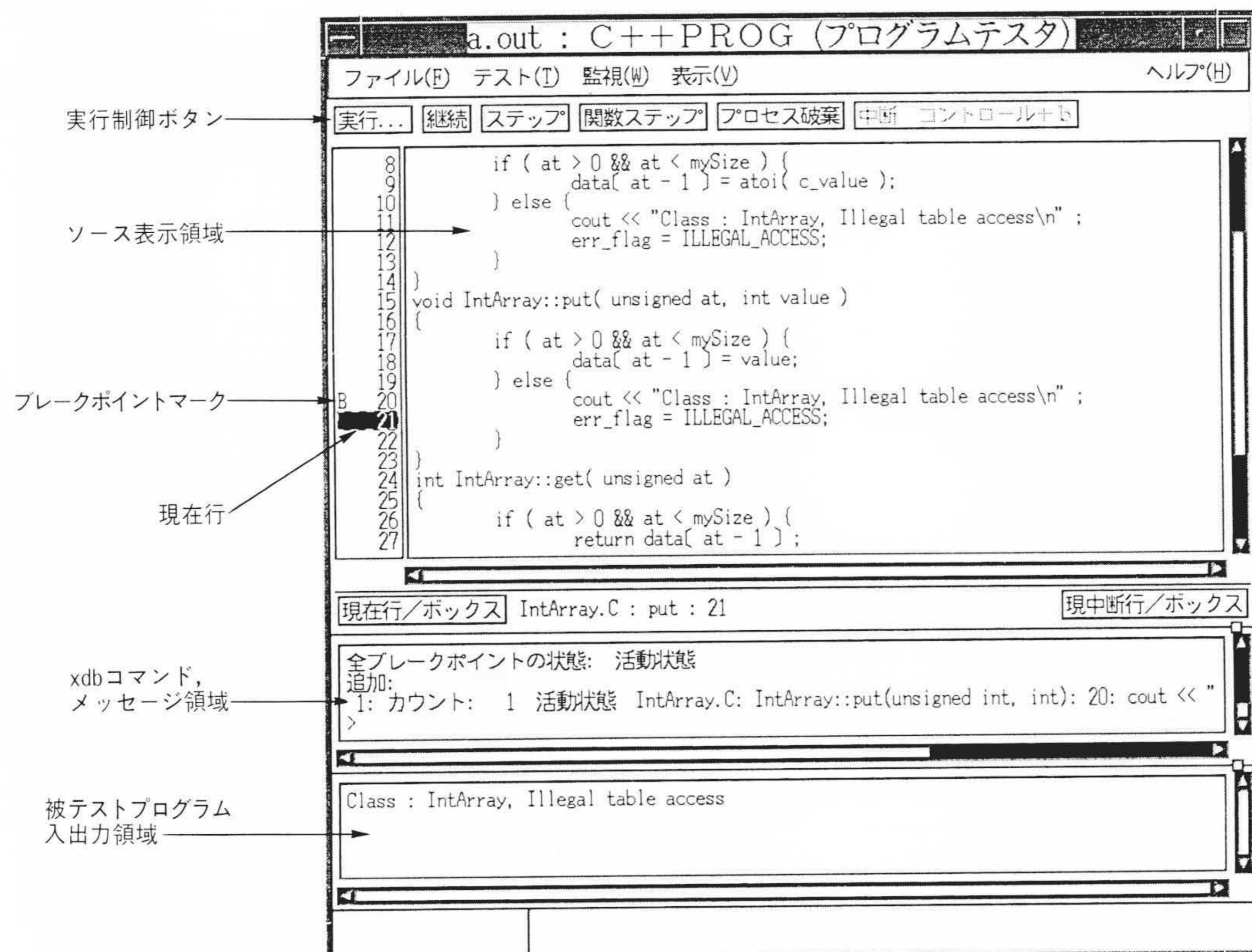


図5 プログラムテスト画面例 プログラムテストは、C/C++プログラムのビジュアルなテストができる。

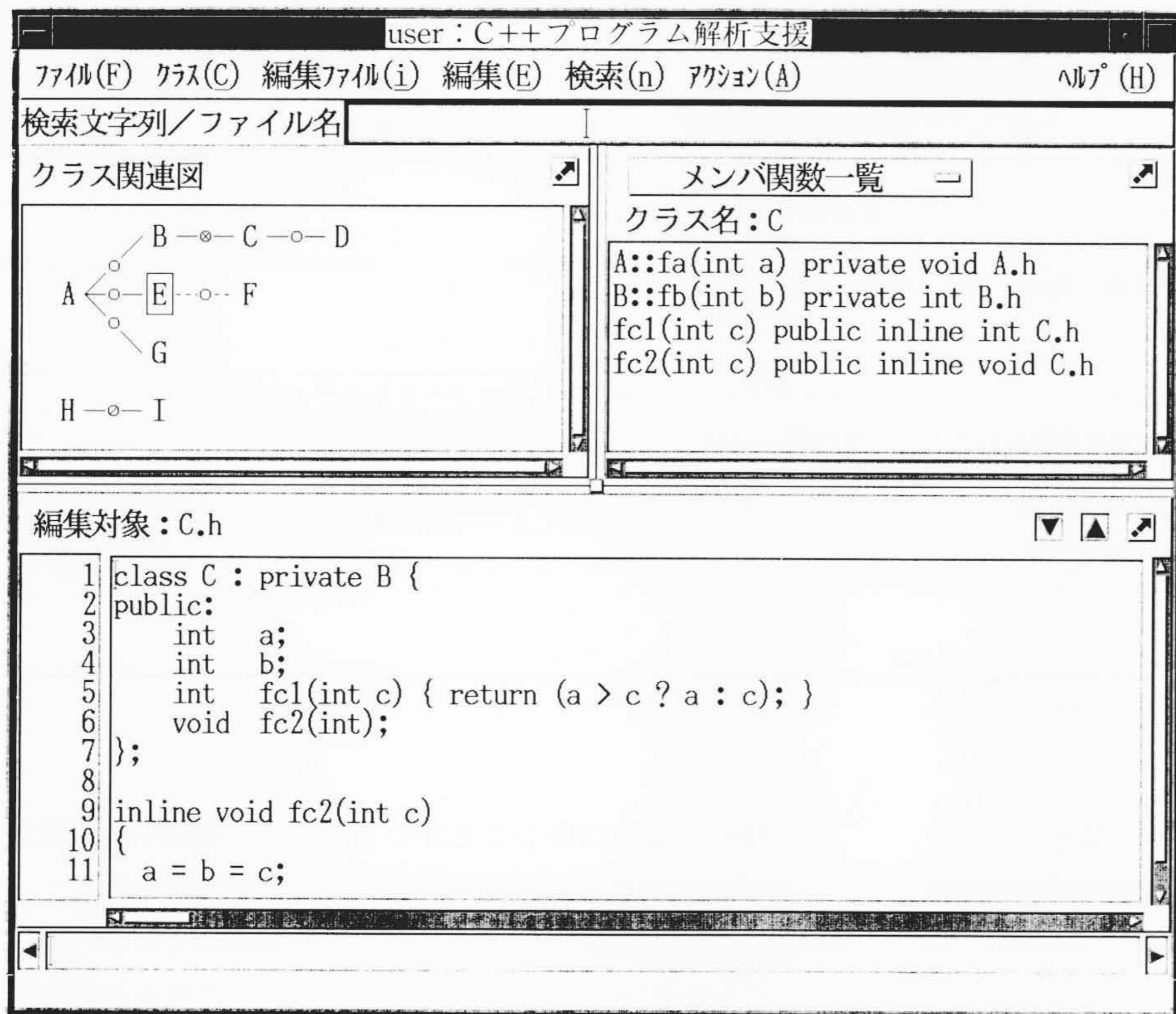


図6 C++ブラウザのメインウィンドウ構成 C++ブラウザのメインウィンドウは、クラス関連図領域(左上)、各種検索結果表示領域(右上)および編集領域(下)から成る。

が行える。

また、マルチプロセスプログラムのテストが可能であり、シェルスクリプトが起動するプログラムや、他のプログラムが子プロセスとして起動するプログラムもテストできる。プログラムテストの画面の例を図5に示す。

(2) プログラム構成管理

makeを用いてプログラムの構成を管理し、プログラムの構築を容易にするツールであり、以下の機能を持つ。

(a) Makefile生成機能

コンパイル、テスト、リンクなどの手順を組み込んだMakefileを自動生成する。

(b) make実行機能

makeの実行をビジュアルに表示する。また、コンパイルエラー発生時のエラー修正も支援する。

6 C++ブラウザ

C++ブラウザはソースのブラウジングはもちろんのこと、make実行や編集、デバッグなどC++によるプログラム開発全体を支援するC++開発環境の中心的なツールである。

6.1 C++ブラウザのウィンドウ構成

C++ブラウザのメインウィンドウを図6に示す。

ウィンドウは三つの領域から成っている。左上がクラ

ス関連図を表示する領域、右上が検索結果などを表示する領域、下が編集領域である。これらの領域の大きさは自由に変更でき、また領域の配置をカスタマイズすることも可能である。

それぞれの領域は複数の表示を重ねることができ、またメインウィンドウから切り離せるため、編集集中の複数のファイルを並べて見ることもできる。

6.2 C++ブラウザの機能概要

C++ブラウザの機能概要を表1にまとめて示す。

(1) クラスブラウジング

クラスの階層や継承の種類、抽象クラスなどをクラス関連図として表示するので、プログラムの構造が視覚的に理解できる。また、クラス関連図上でクラスの追加・削除や継承関係の変更を行うこともできる。このときソースコードも自動的に変更される。

(2) 照会機能

クラス・関数・変数に対し、定義・宣言・更新・参照など使用状態を特定した検索ができる。また、C++のスコープルールに忠実に従った検索ができる。

(3) 編集

エディタは他の領域と密接に関係を持っている。例えば、変数についての検索結果が表示されているとき、それをマウスでダブルクリックするだけで該当個所のソー

表1 C++ブラウザの機能概要 C++ブラウザは、編集、ファイルブラウジングのほか、クラスブラウジング、エラー修正支援や照会機能など豊富な機能を持っている。

項目	内容
クラスブラウジング	<ul style="list-style-type: none"> ●クラス関連図を表示 ●メンバ関数やデータメンバ、フレンド宣言の一覧表示 ●グラフィカルなクラスの編集
照会機能	<ul style="list-style-type: none"> ●クラス・関数・変数の定義・宣言・更新・参照個所の検索 ●全関数の一覧表示 ●文字列の検索
エラー修正支援	●make実行で表示されたコンパイルエラーをマウスクリックすると、該当ソースをエディタに表示
編集	●FSED2/WSと操作性を合わせたエディタを内蔵
ファイルブラウジング	<ul style="list-style-type: none"> ●ファイル一覧の表示 ●ファイルの削除・リネーム
Makefileの生成	●GUIを用いてMakefileを生成
デバッグ	<ul style="list-style-type: none"> ●GUIを用いたxdbによるデバッグ ●カバレッジ情報の取得 ●マルチプロセスのプログラムのテスト

注：略語説明 FSED2/WS(Full Screen Editor 2/Workstation)

スが編集領域に表示される。また、クラス関連図上でクラスをトリプルクリックすると、その定義個所が編集領域に表示される。

(4) Makefileの生成およびデバッグ

これらの機能はC++PROGの呼出しで実現している。

6.3 C++ブラウザの特長

(1) 開発作業全体を支援

テキスト入力からデバッグまでの開発作業を、C++ブラウザのウィンドウから操作できる。

(2) 煩わしいウィンドウ操作からの解放

ウィンドウの数が増えると、ウィンドウの選択やウィンドウのクローズなど操作が煩わしくなる。C++ブラウザは豊富な機能を持つが、これらの機能をつかさどる複数の領域を一つのウィンドウに統合し、これを解決した。

(3) エディタとの密接な関係による使いやすさの向上

プログラム開発には、エディタによる編集が欠かせない。ブラウザで各種のブラウジングを行うのも、最終的にはエディタで修正する個所を特定するためのことが多い。しかし、今まではブラウザで修正個所を特定した後、エディタ上で再度サーチしなければならないツールが多かった。C++ブラウザは、ブラウジングの結果から簡単

な操作で編集に移ることができるので、開発作業の流れがスムーズである。

(4) Cへの対応

C++だけでなくCにも対応しており、これらが混在したプログラムのブラウジングもできる。

7 おわりに

オブジェクト指向言語C++およびその開発支援環境について述べた。その基本構成は、オブジェクト指向によるプログラム開発を支援するC++コンパイラ、デバッガと、クラスのブラウジングだけでなく、各種ツールの有機的な結合によって使い勝手を大幅に向上するC++ブラウザから成る。また、C++の効果的活用の決め手は、ユーザー自身のクラスライブラリの構築にあることもあわせて示した。

C++開発環境は、SEWB3(Software Engineering Workbench 3)プラットフォームとの関係、ES/KERNEL2(Expert System/KERNEL2)との連動も行う予定である。今後も、より効果的なクラスライブラリの構築を支援できるように、さらにツール類の充実、密な関係化を図っていく考えである。

参考文献

- 1) B. Stroustrup, "The C++ Programming Language 2nd edition", Addison-Wesley(1991)
(邦訳：斎藤訳「プログラミング言語C++改訂第2版」, トッパン・情報科学シリーズ)
- 2) M. Ellis, et al.: "The Annotated C++ Reference Manual", Addison-Wesley(1990)
(邦訳：足立, 外訳「注解C++リファレンスマニュアル」, トッパン・情報科学シリーズ)