7Q−5　拡張可能なC++ソースコードブラウザ — 基本設計 —

三ッ井 欽一[†]　Shahram Javey[‡]　久世 和資[†]

†日本アイ・ビー・エム(株) 東京基礎研究所　‡IBM Canada, Toronto Laboratory

1　はじめに

C++のような高機能かつ複雑な言語におけるプログラミングを支援するツールとして、ソースコードブラウザ（以下ブラウザとよぶ）を開発している。ブラウザは、コンパイルの結果得られた情報を基に、これを様々なビューで表現し、プログラムの理解、デバッグ、保守を支援するツールである。

C++は、多機能で複雑な言語である。例えば、関数名、オペレーターのオーバーロード、自動的な型変換、オブジェクトのコンストラクタ、デストラクタなど、いずれも効果的に使った場合は、高度なプログラミングが可能である。しかし、逆に、トリッキーなプログラミングも可能であるとも言え、特に他人が書いたプログラムを理解するのに非常に努力を要する場合がある。最近のC++の仕様[1]では、テンプレートや例外処理が追加され、いっそう高機能化、複雑化の様相を呈している。その結果、ますますC++プログラミングを効果的にサポートするツールの必要性が高まっている。

次節で述べるような、C++プログラミングを効果的に支援するための多様な機能は、必ずしも全て最初から組み込むことができるわけではなく、幾つかの機能は、本質的にアプリケーションごとに特化されるものと予想される。このような要求を満たすための最も重要な要素は、ブラウザが拡張性を持つことである。ユーザが、新しく有効なビューを発見した場合、それが容易に実現できることが重要である。もちろん、そのビューが、グラフィカルなユーザインタフェースを通して表示されることも必要であり、拡張性には、ユーザインタフェースの拡張も含まれる。

本論文では、以上のようなブラウザの設計上考慮した点およびアプローチの仕方について述べる。実際の実現およびその評価については[4], [5]で述べている。

Extensible C++ Source Code Browser — Basic Design
Kin'ichi Mitsui†, Shahram Javey‡, and Kazushi Kuse†
†IBM Research, Tokyo Research Laboratory
‡IBM Canada, Toronto Laboratory

2　ブラウザの機能

本節では、我々が拡張性に重点をおく理由として、ブラウザに必要とされる多様な機能を示すことにする。必要とされる機能の一部として以下のものが挙げられる。

（1）プログラムの構造全体を簡略化した形でみせる。例えば、ファイル間のインクルード関係、クラス間のインヘリタンス関係、部品関係(part-of)等をグラフ表示する。

（2）プログラムの構造、クロスリファレンスにそって情報をたどる。プログラムは、一般にサイクルを含むグラフ構造とみなすことができ、このグラフを逐次たどる（ナビゲーション）機能を提供する。

（3）特定の条件を持つプログラムオブジェクトを検索する。例えば、「特定のクラスのオブジェクトを引数とするすべての関数」や「あるクラスのオブジェクトの状態を変更しているすべての場所」といった検索は、クラスの使われ方の理解やデバッグに有効である。

（4）プログラムが特定のコンベンションに従って書かれているかを検査する。プログラムの開発において、名前の付け方やオブジェクトの作成、操作、削除の手順といったコンベンションを仮定することが多い。これらは、開発が進むにつれて追加・変更されがちである。

（5）使われていないコードをみつける。

（6）モジュール間の依存性を解析する。例えば、プログラムのある部分が変更された場合のインパクトの見積り、makefileやファイル分割の最適化などの応用がある。

（7）プログラムを再編成する場合の支援のために使う。プログラムの開発が進むにつれて、肥大化し、最適でなくなったモジュールに対して、分割・再編成を効率的に行うには、モジュール内の構造を簡略化して理解するためのツールが有効である。

3 ブラウザの設計

我々は、前節で述べた要求を考慮し、以下に述べるアーキテクチャに基づいてブラウザを設計した。

3.1 クライアント・サーバ アーキテクチャ

我々のブラウザは、プログラム情報に対する問合せを処理するプログラムデータベースサーバと、それにアクセスするユーザインタフェースとからなっている。このようなアーキテクチャを採用することにより、分散環境において複数プログラマが同時に単一のプログラムデータベースを利用するという形態をとることが可能である。

3.2 プログラムデータベース

プログラムデータベースは、コンパイラにより生成される情報を入力とし、クライアントプロセスからの問合せ要求を処理する。問合せ言語には、Prolog を採用している。我々のプログラムデータベースは、Prolog のファクトとしてデータスキーマを、ルールとしてC++固有の問合せ処理プログラムを実装している。したがって、データベースシステム自身は、対象とする言語（例えばC++）を仮定していないので、データスキーマとルールを切替えることにより別の言語をサポートすることが可能である。

我々のデータモデルは、基本的に関係モデルである。プログラムは、ネットワーク構造とみなすことができる [6] 。しかし、データの操作に柔軟性を持たせるためには、従来のデータベースモデル論で議論されているように、関係モデルが有利である。関係モデルを採用しているプログラムデータベースとしては、[2][3] 等がある。

3.3 グラフィカル・ユーザインタフェース

ブラウザのグラフィカル･ユーザインタフェースはデータベースへの問合せ結果を様々な表現を用いてウィンドゥに表示する。現在、ブラウザは、リスト、グラフ、テキストの３種類のビューを用意している。

グラフィカル･ユーザインタフェースは、OSF/Motif 上に作成されており、Xウィンドゥシステムのリソースデータベースを用いることによりユーザインタフェースをカスタマイズすることができる。しかし、リソースデータベースは、フォントやウィンドゥの背景色などの単純な指定には向いているが、新しいメニューを追加するなどのより複雑な記述を必要とする指定には適当ではない。

我々のブラウザのユーザインタフェースは、前述のビューの表示やメニュー操作に必要な基本的なウィンドゥ機能を実現する部分と、これらの機能とメニューの定義を結合して、実際に対象言語のブラウザ・ユーザインタフェースを実現している部分とに分けられる。前者は、C++で実現され、後者は、インタプリタ言語により記述されている。このインタプリタ言語を用いることにより、ユーザは、もとのシステムを変更することなく、必要なデータベースへの問合せとそれを呼び出すメニューを作成し、結合させ、さらに問合せの結果を画面上にリスト、グラフ、テキストのいずれかを用いて表示することが可能である。

更に、ユーザのアクション等のイベントの幾つかは、そのコールバック関数がインタプリタ言語で記述できる。例えば、マウスのクリックによる動作は変更可能である。別の例では、ブラウザのテキストビュー上のカーソルの移動に対するコールバック関数を定義し、外部のエディタ（例えば Emacs) にカーソル移動のコマンドを送ることでブラウザと連動させて利用することができる。

4 まとめ

本論文では、我々が開発しているC++ソースコードブラウザの設計について、特に拡張性の観点から述べた。データベースおよびユーザインタフェースの拡張性は我々の開発作業においても、評価･変更のサイクルを速める上で非常に有効であった。更に、論文中で述べた別言語への適用の例として、別の二つのオブジェクト指向言語で記述されたプログラムの理解・保守に我々のブラウザを試用している。

参考文献

[1] M.A. Ellis, B. Stroustrup: The Annotated C++ Reference Manual, Addison-Wesley, 1990.

[2] J.E. Grass, Y. Chen: The C++ Information Abstractor, Proc. of USENIX C++ Conference, 1990.

[3] M. Lejter, S. Meyers, S.P. Reiss: Adding Semantic Information to C++ Development Environments, Proc. of C++ at Work, 1990.

[4] 中村他, 拡張可能な C++ ソースコード・ブラウザ - プログラム・データベース, 第 45 回情処全大, 7Q-06, 1992.

[5] 大平他, 拡張可能な C++ ソースコード・ブラウザ - ユーザインタフェース, 第 45 回情処全大, 7Q-07, 1992.

[6] D.S. Rosenblum, A. L. Wolf: Representing Semantically Analysed C++ Code with Reprise, Proc. of USENIX C++ Conference, 1991.