

プログラムの動作可視化ツールキットの試作

岡崎次朗

1. 背景と目的

プログラムはプログラム初心者や入門者などの論理的思考になじみが薄い人達にとっては理解しづらいものである。その理由として以下が挙げられる。

- (1) プログラムの動作が直感的に把握できない。
- (2) コンピュータの記憶、制御のイメージが理解できない。

しかし現状においては、ツールによるサポートはデバッガの機能としてしかなく、記憶や制御は人間が頭の中で再現する必要がある。しかしながら、その考え方が捉えにくいものであるため、直感的に把握できるようにする必要がある。以上の事由により、データの制御の流れを捉えやすくするため、プログラムの可視化ツールキット(以下、ツールキットと略)の試作を行った。

2. ツールキットの試作

2.1 概要

試作したプログラム動作の可視化ツールキットはC++言語に組み込む形で提供する。C++言語の first class オブジェクトをラップするクラスとして提供し、演算子をオーバーロードすることでその演算子処理の中で対象のオブジェクトの動作を可視化するための情報を外部ログファイルに出力する。演算子の中で処理し、元の演算子の動作をエミュレートすることで、ソースコード上では既存ソースとの差異を最小限に押さえる事ができる。

2.2 機能

ツールキットは以下の機能を提供する。

- (1) ツールキット 変数データの移動履歴情報(ログ)の出力機能。変数の計算処理時のデータの流れをログとして出力する。
- (2) 履歴管理 変数データの移動履歴情報の管理機能。アニメーションする上での3Dオブジェクトの管理や移動スケジュールを管理する。
- (3) ログビューア 複数オブジェクトの並列アニメーション機能。タイマを利用して同時に複数の3Dオブジェクトをアニメーションさせる。
- (4) 表示 GLUT ライブラリを用いてオブジェクトを表示する機能。スケジュール情報をもとに OpenGL 命令へ変換し、表示する。

2.3 利用法

利用者はプログラムソースコードを以下のように変更することでツールキットを利用することができる。“int integerVariable = 10;”のような場合、“Int integerVariable = 10;”のように first class オブジェクトの名前の最初の一文字を大文字として表記したクラスを用いたプログラムを動作させることで motion.log が出力される。出力した motion.log を logViewer.exe の引数として、実行することで変数間でのデータの動きを表示する。

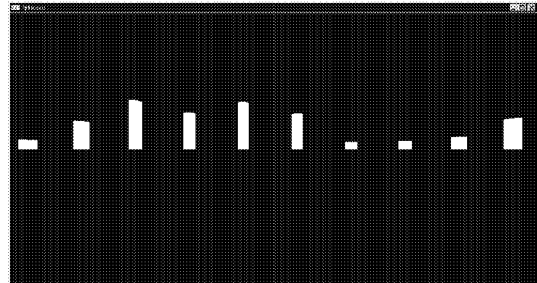


図1 ログビューア実行画面

2.4 出力例

プログラムを実行後、出力されたログをログビューアに処理させることにより、図1の画面上で、変数間でのデータの動きを表現する。

3. 評価と考察

3.1 クイックソートの事例

3.1.1 イベント主導プログラムとツールキットの比較
イベントドリブンとして作られたプログラムではマルチスレッド化しない限り、ソート処理と画面表示処理を交互に行う必要があるため、そのままでは表示がうまくできない。しかし、ログベースであるツールキットでは、実行履歴をもとに表示データを作成するため、特にユーザインタフェース処理を考慮すること無く対象プログラムを作成することが可能である。

3.1.2 履歴を利用する方法の長所と短所

- (1) 長所 特別なプログラム作成を意識すること無く元のプログラム可視化が可能である。
- (2) 短所 可視化処理はあくまで実行結果をトレースするだけであるため、インタラクティブに処理途中に動作を変更することができない。

4. 課題

複数の項からなる式の場合、項ごとに計算結果を示す一時オブジェクトが作成されその間のデータの動きも記録される。そのためプログラム上で記述したもの以外のデータの流れも表示されてしまう。これらは動作を煩雑に見せるためうまく抑制する方法が必要である。

参考文献

- [1] Robin Hunter. コンパイラ構成論 Pascal を用いた設計と実装. 近代科学社, 1991.
- [2] Tucker. 憂鬱なプログラマのためのオブジェクト指向開発講座 C++による実践的ソフトウェア構築入門. 翔泳社, 1998.
- [3] 五月女健治. yacc/lex プログラミングジェネレータ on UNIX. テクノプレス, 1996.