

プログラム理解を支援するシーケンス図表示システムの開発

塩谷 祐介[†] 岩澤 京子[‡]

拓殖大学 工学研究科 電子情報工学専攻[†] 拓殖大学 情報工学科[‡]

1 背景と目的

Java のようなオブジェクト指向のプログラムでは、オブジェクト間のメッセージ通信によってシステムが動作する。しかしソースコードからプログラムの動的な振舞いを読取って理解することは困難である。そこでシーケンス図を用いてメッセージ通信の様子を可視化すればプログラムの理解を支援できると考えた。しかしシーケンス図をシステムの動作の表示に用いるのは、ユーザの手書きでは膨大な手間がかかる上にミスも発生してしまう。

本研究ではアスペクト指向によって動作ログを取得し、その後ユーザプログラムの実行とはオフラインでシーケンス図を表示するシステムの開発を行うことにした。

2 システムの機能と設計方針

システムの構成を図 1 に示す。本システムには 3 つの処理部がある。まず織込み処理部がユーザプログラムからクラス名を取得し、ログ出力プログラムにクラスを上書きする。そのプログラムをユーザプログラムにアスペクトコンパイラによってウィーブする。この織込み後のプログラムを実行すると、インスタンス生成とメソッド呼出しの実行に割り込んでログファイルにログ情報を出力する。次にログデータ解析処理部がログファイルを解析し、シーケンス図の表

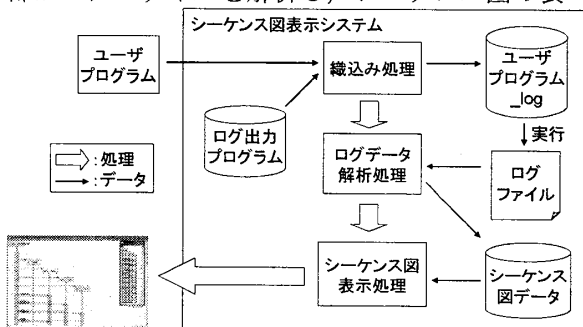


図 1 システム構成図

示に必要なデータを用意する。最後にシーケンス図表示処理部がそのデータからシーケンス図を作成し GUI に表示する。

プログラムの実行の全体像なしでシーケンス図を表示するのは、リアルタイムにシーケンス図を描くシステムの実装が難しかった。そこで動作ログを取得し、ユーザプログラムとはオフラインでシーケンス図を表示することにした。またシーケンス図には、多くのメソッド呼出しが行われる複雑なプログラムを表示したときに全体像を把握できなくなる問題があった。そこで本システムではシーケンス図を縮小した全体図とクラス間の呼出し関係をまとめた図を用いて階層化を行うことにした。更に通常のシーケンス図には無い情報として、メソッド呼出し時の引数と戻り値の情報を表示することでシステムの有効性を向上させる。

3 動作ログの取得

シーケンス図の出力に必要なログデータはログ出力プログラムをウィーブした後にユーザプログラムを実行することで得られる。ログ出力プログラムにクラス名を入力する処理やプログラムをウィーブする処理は本システムの織込み処理部が行うため、ユーザはプログラムに余計な手を加えずにログデータを得られる。ログ出力プログラムの例を図 2 に示す。

```
//メソッドのログ
after():call(クラス名+*(.)) && !within(S_logger){
    System.out.println("Method "+thisJoinPoint.getSignature().getName());
    System.out.println(thisJoinPoint.getSignature().getDeclaringType());
}
//引数のログ
before():call(クラス名+*(.)) && !within(S_logger){
    System.out.println("Argument "+thisJoinPoint.getSignature().hashCode());
    System.out.println(thisJoinPoint.getArgs()[0]);
    System.out.println(thisJoinPoint.getArgs()[0].getClass());
}
```

図 2 ログ出力プログラム

メソッド呼出し時に収集する動作ログは次の 4 つの項目を含んでいる。

- ・呼出されたメソッド
- ・そのメソッドのインスタンス
- ・引数
- ・戻り値

ログデータ取得部は呼出されたメソッドやそ

Prototyping of layered sequence graph display system for understanding Java program

[†] Yuusuke Shioya, Graduate School of Electronic Information Engineering, Takushoku University.

[‡] Kyoko Iwasawa, Information Engineering, Takushoku

のインスタンスのソースコード上の名前を取得する。仮引数と戻り値は名前以外に型と値の情報も取得する。

インスタンス生成時に収集する動作ログは、インスタンス名とそのクラス名に加えて生成を行ったメソッドの名前とそのインスタンス名も取得する。

4 ログデータの解析処理

ログファイルの状態は時系列で収集した情報なので、ログデータの解析を行ってシーケンス図表示処理部で使用するデータ構造体を作成しなければならない。シーケンス図データのクラス図を図3に示す。

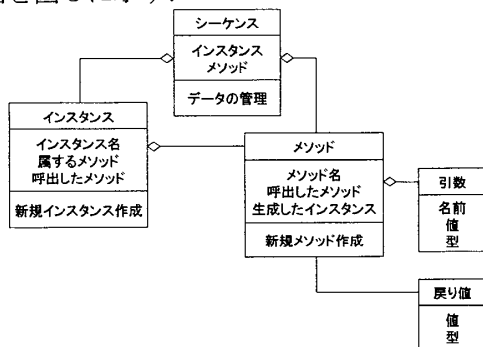


図3 シーケンス図データのクラス図

シーケンス図のクラスは関連のある変数で繋がっており、シーケンス図作成時にデータを最短距離で参照することができる。

シーケンス図クラスは全てのインスタンスとメソッドのリストを持っている。シーケンス図表示処理ではまず始めにこのクラスから呼出して次に表示するオブジェクトを得る。次にインスタンスクラスは名前と関連するメソッドのデータを持っている。関連するメソッドとは、各インスタンスクラスのメソッドと、そこで呼出したメソッドの2種類である。図3中央のメソッドクラスは名前と呼出したメソッドのデータを持っており、更にインスタンス生成を行った場合にはそのインスタンスのデータも持つ。引数クラスでは名前と値と型を格納する。一方戻り値は名前が必要ないので、値と型のみにした。

5 シーケンス図の表示

シーケンス図表示処理部ではシーケンス図データのインスタンス情報からシーケンス図のオブジェクトを作成し、メソッド情報からメッセージの配置とライフラインの長さを計り、シーケンス図クラスの情報に基づいて図を作成する。

図4のシーケンス図の特徴としては、メッセージ通信の矢印上に引数と戻り値の情報を表示

している。複雑なメソッド呼出しではプログラムとシーケンス図の対応が取りにくい場合があるが、この機能でシーケンス図がより読み取りやすくなっている。

しかしこのシーケンス図では更に複雑で巨大なプログラムを表示した場合に全体像が把握できなくなる問題があった。そこで全体を見るためにシーケンス図をできるかぎり縮小した2つ目の図を表示することにした。シーケンス図の例を図4に示す。

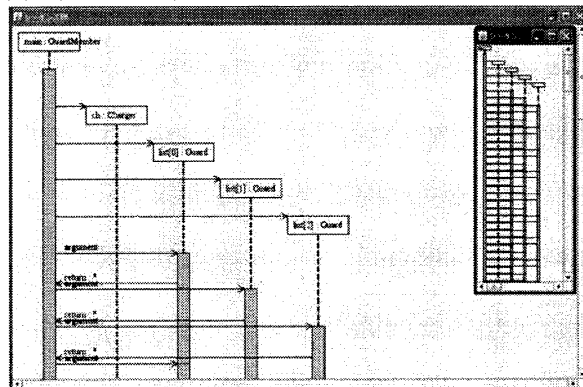


図4 シーケンス図表示例

更にこの2つのシーケンス図は互いに対応しており、2つの図で見比べたい部分をクリックすることで色が変わり、対応関係を分かりやすく表示する。

6 まとめと今後の課題

ユーザプログラムの実際の動きをシーケンス図を用いて可視化することができた。本システムによりユーザが手書きで作成した場合に比べてより手軽にシーケンス図を表示することができる。更にシステムの特徴として動的な引数の値を表示している。また大規模なプログラムでシーケンス図の全体像が見にくくなるという問題に対しては、シーケンス図を縮小表示することで対応した。

今後は更にシーケンス図の改良を行っていく。現在開発中の機能として、クラス同士の呼出し関係図を用いて更に大規模なプログラムに対する有効性を向上させたい。

参考文献

- [1] 松本啓之亮著. ソフトウェア工学 オブジェクト指向・UML・プロジェクト管理. 森北出版, 2005年3月, 185頁
- [2] Widodo Erwin, Ohnishi Atsushi. クラス図とシーケンス図間における矛盾の識別. Technical report of IEICE SS, Vol.106, No.120, pp.25-30