

オブジェクト生成の観測に基づくプログラム実行の要約の抽出

中野 佑紀 渡邊 結 石尾 隆 井上 克郎
大阪大学大学院情報科学研究科

1 はじめに

デバッガとブレイクポイントを用いてプログラムの実行時動作を確認するとき、すべての命令をステップ実行していくことは現実的ではないが、次のブレイクポイントまで一気に実行するとその間に行われている処理を知ることはできない。そのため、ある実行区間の概要を知る方法が必要である。

オブジェクト指向プログラムでは、複数のオブジェクトが相互にメッセージ通信を行うことにより処理が行われる。開発者にとって意味のあるタスクを実行するとき、その途中では多くのオブジェクトが生成されるが、タスクの出力となるオブジェクトは一部である [1]。タスク間でのオブジェクトの利用関係を調べることでタスク間の関係を知ることができると言われているが [2]、本研究では一つのタスクにおける入出力オブジェクトの関係が分かれば、そのタスクの概要が理解できると考えた。

本研究では開発者が指定した実行履歴上の区間における入出力オブジェクトとオブジェクトの生成関係の要約を提示する手法を提案する。これにより、開発者がデバッグ等の作業において、興味のある実行区間の概要を理解する作業を支援する。

2 提案手法

本手法は開発者により指定された実行履歴上の区間における入出力オブジェクトとオブジェクトの生成関係を要約したグラフを求める。

まず指定された区間における入力オブジェクト、出力オブジェクト、内部オブジェクトを特定する。オブジェクトの区分について図 1 に示す。

入力 区間より前の実行で生成され、区間内の実行で利用されるオブジェクト

Extraction of Summarized Program Execution Based on Observation of Object Creation

Yuki NAKANO, Yui WATANABE, Takashi ISHIO,
Katsuro INOUE
Graduate School of Information Science and Technology,
Osaka University

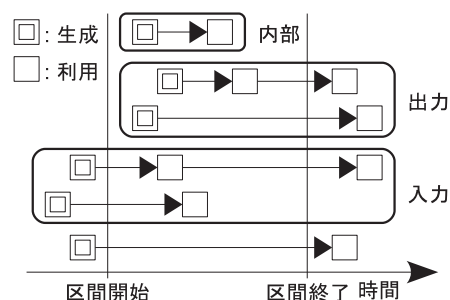


図 1: オブジェクトの区分

```
class RegisterDialog {
    ...
    TRIGGER
    newSchedule = new Schedule(date, time, memo);
    ...
    BASE
}
```

図 2: オブジェクトの生成関係の特定例

出力 区間内の実行で生成され、区間終了後の実行で利用されるオブジェクト

内部 区間内の実行で生成され、区間終了後の実行で利用されないオブジェクト

次にオブジェクトの生成関係を特定するため、出力オブジェクトと内部オブジェクトについて、それらを生成したオブジェクト TRIGGER、生成時に利用されたオブジェクト BASE を求める。Java におけるオブジェクトの生成関係の場合、図 2 に示すようにコンストラクタ呼び出しを行うオブジェクトが TRIGGER、コンストラクタの引数になるオブジェクトが BASE となる。

特定したオブジェクトの生成関係に対し、オブジェクトの集約、省略を行うことにより要約を作成する。集約では同じ生成パターンを持つオブジェクト群、すなわち、対応するオブジェクトのパッケージ名、クラス名、区分が等しく、かつ対応するオブジェクト間の生成関係が等しいオブジェクト群を一つのグループにまとめる。省略では入出力オブジェクトと関係をもたないオブジェクトを取り除く。

3 ツール

Java プログラムを対象とし、提案手法を実現するツールを試作した。ツールへの入力の実行履歴と、興味の

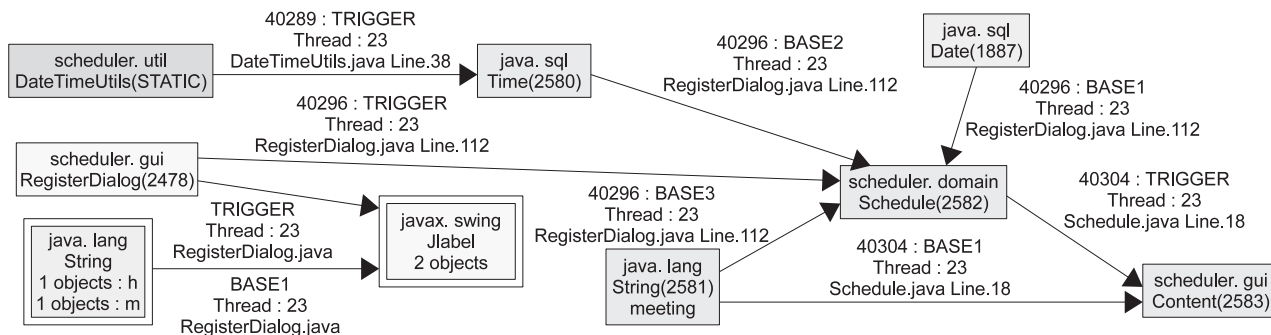


図 3: オブジェクトの生成関係を示すグラフの表示例

表 1: 入出力オブジェクトの一覧の表示例

I/O	ID	Name	Value
I	6	scheduler.domain.Scheduler	
I	2029	java.lang.Boolean	true
O	2581	java.lang.String	meeting

ある実行区間の開始時刻, 終了時刻である. 実行履歴の取得には Amida^{*1}に含まれるトレーサを用いた. ツールの出力は入出力オブジェクトの一覧, 要約したオブジェクトの生成関係を示したグラフである.

入出力オブジェクトの一覧の表示例を表 1 に示す. 入出力オブジェクトの一覧では特定した入力オブジェクトと出力オブジェクトについて, 入出力の区分, オブジェクト ID, クラスの完全修飾名を表示する. また, 実行履歴から値が取得可能なクラス (String, Boolean, Integer 等) についてはその値を表示する.

オブジェクトの生成関係を示すグラフではノードはオブジェクトを表し, エッジは生成関係を表す. グラフの表示例を図 3 に示す.

ノードにはオブジェクトの情報としてパッケージ名, クラス名, オブジェクト ID を表示する. 実行履歴から値が取得可能なクラスについてはその値を表示する. オブジェクトのグループを表すノードの場合, 枠を二重にして, オブジェクト ID を示さずに個数を表示する. 実行履歴から値が取得可能なクラスについては値ごとの個数を表示する. ノードは入力, 出力, 内部, STATIC の区分により色分けして表示する.

エッジには対応する関係のほかに生成時の情報としてタイムスタンプとスレッド ID を, 対応するソースコードの位置情報としてファイル名と行番号を表示する. 関係が BASE の場合は後ろに何番目の引数であるかの数字を表示する. 集約された関係を表すエッジの場合, 集約される前の情報のうち, 共通する情報のみを表示する.

予定管理プログラムで予定の追加を行う処理に対応

する区間に対し, ツールを適用したところ, 要約前のノード数が 247, 要約後のノード数が 64 となった. 図 3 はこの結果から一部抜粋したものである. この図から追加された予定を表す Schedule オブジェクトは予定入力ダイアログを表す RegisterDialog オブジェクトにより作成され, 日付を表す Date オブジェクト, 時刻を表す Time オブジェクト, 予定内容の文字列 meeting を表す String オブジェクトを作成時に利用していることが分かる.

4 まとめ, 今後の課題

開発者が興味のある実行区間の概要を理解する作業を支援するために, 指定された実行履歴上の区間における入出力オブジェクトの一覧, オブジェクトの生成関係を要約したグラフを提示する手法の提案を行い, ツールを試作した.

今後の課題としては, 出力される生成関係は重要なオブジェクトを残し, できるだけ簡潔なものにする必要があるため, 適切な要約基準を決定する必要がある. また, 要約を行った結果で重要な情報が省略されていないか, 実際に実行区間の概要の理解支援を行うことが可能かを評価することが必要である.

謝辞

本研究は, 文部科学省科学研究費補助金若手研究 (B)(課題番号:21700030) の助成を得た.

参考文献

- [1] Lieberman, H. and Hewitt, C.: A Real-Time Garbage Collector Based on the Lifetimes of Objects. *Comm. ACM*, Vol.26, No.6, pp.419-429 (1983).
- [2] Lienhard, A., Greevy, O. and Nierstras, O.: Tracking Objects to Detect Feature Dependencies. *Proc. ICPC'07*, pp.59-68 (2007).

*1Amida. <http://sel.ist.osaka-u.ac.jp/~ishio/amida/>