The University of Osaka
Institutional Knowledge Archive

Title	協調動作するオブジェクト群に着目したJavaプログラムの実行履歴分割手法の提案
Author(s)	大平, 直宏; 谷口, 考治; 石尾, 隆 他
Citation	電子情報通信学会総合大会講演論文集. 2005, 2005(1), p. 25-25
Version Type	VoR
URL	https://hdl.handle.net/11094/51100
rights	copyright©2005 IEICE
Note	

The University of Osaka Institutional Knowledge Archive : OUKA

https://ir.library.osaka-u.ac.jp/

The University of Osaka

D-3-5

協調動作するオブジェクト群に着目した Javaプログラムの実行履歴分割手法の提案

Division of Java Execution Trace Based on Collaboration of Objects

大平 直宏† 谷口 考治† 石尾 隆† 神谷 年洋‡ 楠本 真二† 井上 克郎†
Naohiro Ohira† Koji Taniguchi† Takashi Ishio† Toshihiro Kamiya‡ Shinji Kusumoto† Katsuro Inoue†

†大阪大学大学院情報科学研究科
†Graduate School of Information Science and Technology, Osaka University

‡科学技術振興機構さきがけ
‡Presto, Japan Science and Technology Agency

1．まえがき

我々は，Java プログラムの振る舞い理解を支援するために，実行プログラムのメソッド呼び出し履歴を取得し，この実行履歴を基にシーケンス図を生成する手法を提案している[1]．しかし，生成されるシーケンス図は実行履歴のサイズに比例して巨大なものになっている．そこで，実行中に協調動作するオブジェクト群に注目し，実行履歴を意味のある複数ブロックに分割した上で，各ブロック単位にシーケンス図を表示することを考えている．本稿では，実行履歴を分割する手法の概略と手法の適用事例について述べる．

2．協調動作するオブジェクトの時間変化例

一般にプログラムの 1 つの実行履歴は，初期化部やユーザ入力部，内部計算部，GUI 出力部など，システムの複数のフェイズを含んでおり，各フェイズではそのフェイズ固有のオブジェクト群が協調動作していると考えられる(例えば図 1 は，ユーザ入力部と GUI 出力部では，そのフェイズで協調動作するオブジェクト群が異なることを示す)．

そこで，実際のプログラム(著者らの 1 人が作成した，入力された Java プログラムを解析してプログラム依存グラフを出力するツール)の実行履歴を用いて，時間の経過とともにオブジェクトの動作状況が変化する様子をグラフ化した．図 2 は，時刻 t (実行履歴中の t 番目)のメソッド呼び出しに関して，呼び出し元と呼び出し先のオブジェクト ID をそれぞれプロットしたものである．この図より，約 500 番目，2600 番目のメソッド呼び出しを境にメッセージ通信に関わるオブジェクト群が急激に変化しているのが分かる．例えば，500 番目以前のメソッド呼び出しで動作したオブジェクトはその後ほとんど動作していない．実際に，1～500 番目，500～2600 番目までの部分は 2 個の入力データをそれぞれ個別に処理する部分であり，さらに 2600 番目以降の部分はファイル出力部に対応していることが確認できた．

3．実行履歴の分割

2．で調べたように，プログラムの実行履歴を時間順に追跡すると，プログラムの異なるフェイズに入った部分で協調動作しているオブジェクト群が大きく変化すると予想される．そこで，メッセージ通信に関わるオブジェクト群の変化に注目することで実行履歴を複数のブロックに分割する方法を検討した．具体的には，メソッド呼び出しに関係していたオブジェクトをキャッシュに記憶しておき，キャッシュ中のオブジェクトの入れ替わり頻度が大きくなる部分で実行履歴を分割するというアプローチである．

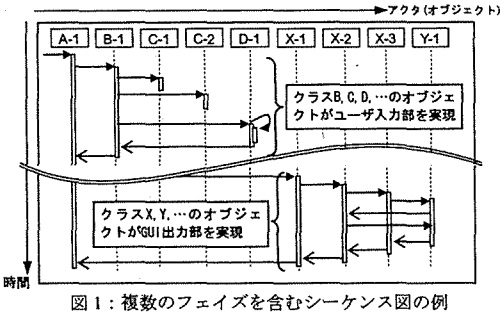

図 1：複数のフェイズを含むシーケンス図の例

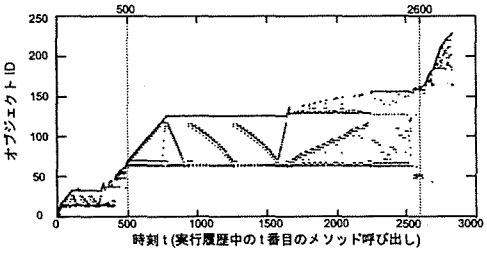

図 2：協調動作するオブジェクトの時間による変化

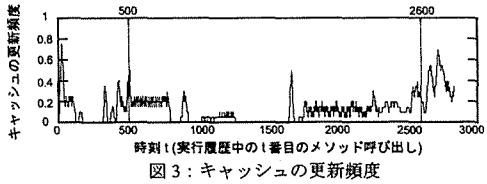

図 3：キャッシュの更新頻度

図 2 で使用した実行履歴に対し，実際に実行履歴の分割を試みた．その一例として，キャッシュサイズを 50 とし，各時刻でのメソッド呼び出し毎に過去 20 回分のオブジェクトの入れ替わり頻度(0～1)をグラフ化した結果を図 3 の折れ線グラフに示す．この例からは，システムの異なるフェイズに処理が移行したと考えられる部分で実際にキャッシュの更新頻度が高くなっていることが確認できた．

今後，様々な種類のプログラムの実行履歴に対して適用を行い，提案手法の詳細化・一般化を行う予定である．

参考文献

[1] 谷口, 石尾, 神谷, 楠本, 井上: “Java プログラムの実行履歴に基づくシーケンス図の作成”, FOSE2004 予稿集, pp.5-15 (Nov. 2004).