

# オブジェクトの動作の比較による実行履歴削減手法の提案

脇 阪 大 輝<sup>†1</sup> 石 尾 隆<sup>†1</sup> 井 上 克 郎<sup>†1</sup>

ソフトウェアの稼働中に障害が発生したとき、開発者は障害が生じた実行についての履歴を集め、デバッグ環境でその障害を再現しようとする。障害を再現するための実行履歴は、ロガーツールによってプログラムの実行を監視、保存することによって得られるが、すべての実行情報を保存しているとその情報の量が膨大になるため、履歴を常に保存し続けることは困難である。本研究では、プログラムの実行を事前の実行と比較し、未知の動作をしそうな実行についてのみ、その実行の履歴を保存することで、実行履歴の量を軽減する手法を提案する。具体的には、事前にプログラムを実行し、オブジェクトの動作を蓄積しておく。その後、本番環境でプログラムを実行するとき、あるメソッドの入力となるオブジェクトの動作を、事前の実行でそのメソッドの入力となったオブジェクトの動作と比較し、過去の記録にない動作をしているオブジェクトが入力となるメソッドの実行のみを記録する。

## A Method for Reducing Execution Traces by Comparing Object Behavior

HIROKI WAKISAKA,<sup>†1</sup> TAKASHI ISHIO<sup>†1</sup> and KATSURO INOUE<sup>†1</sup>

Software developers gather execution traces of a program and reproduce the execution in-house when software fails in the field. Developers obtain the execution traces from logger tools but the volume of that tends to be huge. It is therefore difficult that logger tools always record execution traces. In this paper, we propose an approach for reducing the volume of execution traces. The approach runs a program in advance and records behavior of objects in it. When the program runs in the field, the approach compares the behavior of objects in parameters of a method with the behavior in previous executions of the program and records traces only for method executions of which parameter objects behave differently from the previous executions.

### 1. はじめに

ソフトウェアのバグは開発中のテストにより、発見され修正されるが、すべてのバグをテストによって発見することができるとは限らない<sup>1)-2)</sup>。リリースされたソフトウェアに残存するバグは、そのソフトウェアが利用される現場で顕在化し、障害を引き起こす。

稼働中のソフトウェアに残存するバグによって障害が発生したとき、開発者は、障害が生じた際の実行履歴を用いて、デバッグ環境にてそれを再現することを試みる。この実行履歴には様々な形式があるが、スタックトレース等の軽量な形式では、障害の再現が困難なこともある。そのため、実行履歴をより詳細に取得することが望まれる。たとえば、各命令で使用したオブジェクトの ID やフィールドの値、条件分岐などを記録すると、オブジェクトの状態や制御フローの異常を分析することができる。

しかし、実行の詳細な履歴は、そのデータ量が膨大となる。したがって、いつ顕在化するかわからないバグに備えて、ソフトウェアの実行の詳細を常に監視、記録することは、ディスク容量を浪費することになる。

本研究では、未知の動作をしているオブジェクトが入力となるメソッドの実行の履歴のみを保存することで、実行履歴の量を削減する手法を提案する。提案手法は、あらかじめプログラムを事前に行う。そのとき出現したオブジェクトの動作を記録する。また、実行されたメソッドとその入力となったオブジェクトの組を記録する。その後、本番環境でプログラムを実行するとき、メソッドの実行が開始される度に、そのメソッドの入力となっているオブジェクトと、事前の実行でそのメソッドの入力となっているオブジェクトの動作を比較する。比較の結果、入力となっているオブジェクトの動作が、事前の実行による記録に含まれている場合、そのメソッドの実行を記録しない。提案手法によって、事前には実行されなかったメソッドの実行や、未知の動作をしているオブジェクトが入力と

<sup>†1</sup> 大阪大学大学院情報科学研究科  
Graduate School of Information Science and Technology

なるメソッドの実行のみについて、その履歴が保存されることになる。

## 2. 関連研究

本研究では、プログラム中のオブジェクトの動作をオートマトンとして抽出している。宗像ら<sup>3)</sup>の研究では、同一クラス内に多数のオブジェクトが存在する場合にも、それらの動作は少数のグループに分類することが可能であることが示されている。提案手法は、同じグループに分類されるようなオブジェクトが入力となるメソッドは同じ動作するという考えに基づいている。

実行履歴を削減する手法として、プログラムの外部入力などの非決定的な振舞いの情報のみを保存する手法<sup>4)</sup>がある。この手法は、実行履歴の量を削減できるが、デバッグに必要なプログラムの状態を再現するために、保存した情報を用いてプログラム全体を実行しなおす必要がある。提案手法では、メソッド単位での実行履歴を保存するので、バグ再現に関係のありそうな実行のみを部分的に再現することができる。

また、プログラムに障害が発生してから、その原因となるバグに関係しそうな箇所のみの実行履歴を保存する手法<sup>5)</sup>がある。この手法は、障害が発生してから、それを再現できるまでの間も、プログラムを稼働し続ける必要がある。提案手法は、障害が初めて発生したときに、その再現に必要な実行履歴が保存されていることを理想としている。

## 3. 実行履歴削減手法

提案手法は、事前にプログラムを実行するときの処理と、本番環境でプログラムを実行するときの処理に分けられる。

事前にプログラムを実行する段階では、出現したオブジェクトの動作を表すオートマトンと、実行されたメソッドとその入力となったオブジェクトの組を記録する。オートマトンは、入力がメソッドの呼出し位置で、同じメソッド呼出し位置を入力として与えると同じ状態へ遷移するように状態が定義される。また、メソッドの入力となるオブジェクトとは、メソッドを実行されるオブジェクトと、メソッドの引数に与えられるオブジェクトを指す。ここで得られるオートマトンと、メソッドの入力を、プログラムの既知の動作を表す記録として扱う。

本番環境でのプログラムの実行では、あるメソッドの実行が開始されたとき、その入力となっているオブジェクトの動作と、事前の実行でそのメソッドの入力

となったオブジェクトの動作を比較する。比較は、入力となっているオブジェクトに対するメソッド呼出し位置を、事前の実行で得られたオートマトンに入力として与えることで行う。1つ以上のオートマトンが状態遷移できているうちは、オブジェクトの動作を既知であるとみなす。すべてのオートマトンが状態遷移できなくなった場合、その時点から、オブジェクト動作は未知であるとみなす。メソッドの入力に、未知の動作をするオブジェクトが存在する場合にのみ、そのメソッドの実行を実行履歴取得の対象とする。

## 4. 今後の課題

提案手法を評価するために、実行履歴を削減できるか、保存されるメソッド実行単位の実行履歴はデバッグに有益であるか、そしてプログラムにどの程度のパフォーマンス低下を及ぼすかを調査することを考えている。

実行履歴の削減量については、ベンチマークソフトを用いた評価実験を行うことで調査することを予定している。その評価実験では、オプションによって実行の規模を指定しプログラムを実行することができる、DaCapo ベンチマーク<sup>\*1</sup>を対象に用いる。小規模な実行を事前の実行とし、大規模な実行を本番環境での実行とすることで、実行履歴の削減量を調査する。

他の2点については、実験方法を現在検討中である。

謝辞 本研究は、JSPS 科研費 23680001, 25220003 の助成を受けたものである。

## 参考文献

- 1) Clause, J. and Orso, A.: A Technique for Enabling and Supporting Debugging of Field Failures, *Proc. ICSE*, pp.261–270 (2007).
- 2) Jaygarl, H., Kim, S., Xie, T. and Chang, C.K.: OCAT: object capture-based automated testing, *Proc. ISSSTA*, pp.159–170 (2010).
- 3) 宗像 聡, 石尾 隆, 井上克郎: 類似した振舞いのオブジェクトのグループ化によるクラス動作シナリオの可視化, 情報処理学会研究報告, Vol.2009-SE-167, No.31, pp.225–232 (2009).
- 4) Bell, J., Sarda, N. and Kaiser, G.: Chroni- cles: lightweight recording to reproduce field failures, *Proc. ISSSTA*, pp.362–371 (2013).
- 5) Chilimbi, T.M., Liblit, B., Mehra, K., Nori, A.V. and Vaswani, K.: HOLMES: Effective statistical debugging via efficient path profiling, *Proc. ICSE*, pp.34–44 (2009).

\*1 DaCapo. <http://www.dacapobench.org/>