UML設計情報を用いたデバッグ支援の提案
The proposals of support debug by using Unified Modeling Language

任浩元† 櫻井孝平‡ 古宮誠一‡
Haoyuan Ren† Kouhei Sakurai‡ Seiichi Komiya‡
† 芝浦工業大学　工学部
‡ 芝浦工業大学大学院　工学研究科
† Facultu of Engineering, Shibaura Institute of Technology.
‡ Graduate School of Engineering, Shibaura Institute of Technology

要旨
ソフトウェアデバッグの欠陥を修正する作業の中、一番の問題点は、プログラムの大規模化と複雑化により、欠陥の原因となるバグ位置の特定が困難である。デバッグ支援をするために、プログラムの UML 設計情報とプログラム実行のトレース情報を自動的に対応させる手法を提案する。具体的には設計工程で扱う UML のシーケンス図と逆戻りデバッガにより得られる実行履歴が一貫しているかを提示するツールを開発する。

1. はじめに

ソフトウェアデバッグとは、欠陥を修正のするための一連の作業である。欠陥とはプログラムの実行する時の障害である。デバッグプロセスは一般的に欠陥の再現、原因となるバグ位置の特定、そしてバグの修正から構成される。この中でも特に、バグ位置の特定は、プログラム上の原因となる場所を発見する作業で、プログラムが大規模化、複雑化するにつれて困難になる。本研究はデバッグ支援のために、プログラムの設計情報とプログラム実行のトレース情報を自動的に対応させる手法を提案する。具体的には、設計工程で扱う UML のシーケンス図と逆戻りデバッガにより得られる実行履歴が一貫しているかを利用者に提示するツールを開発する。

2. 逆戻りデバッガによるデバッグ問題

2.1. 逆戻りデバッガ

逆戻りデバッグとは、プログラムで正しくない結果が生まれた場所から遡って、バグ位置を特定する手法である。[1]

本研究で使う Traceglasses とは、トレース情報に基づく Java を対象となる逆戻りデバッガである。

Traceglasses は、プログラムの実行中のイベントを記録したトレース情報の対話的な表示と検索が可能で、利用者がプログラムの動作を遡って欠陥を発見することを支援する。Traceglasses は、トレース情報をメソッド呼出し関係により木構造に変換し、ブレークポイントデバッガに見られるようなステップ操作を行わずに、トレース情報の表示を行う。また，指定されたオブジェクトに対する操作のみを抽出したトレース情報を表示できる。これらの特徴により、利用者は、実行順に並んだ膨大なトレース情報に対して局所的および大域的に、必要な部分だけを理解し、欠陥を発見できる。[2]

2.2. 設計情報との食い違い

トレース情報として再現された欠陥は、オブジェクトの対話を定義した設計情報との食い違いを生む。このような設計情報は、UML のシーケンス図として記述できる。シーケンス図は、オブジェクト間のメソッド呼び出しやオブジェクトの生成などのメッセージ通信を、時系列に沿って示すことができる。

図 1 はログインシステムのシーケンス図の例である。

欠陥が発生すると、食い違いとして、シーケンス図のメッセージとトレース情報のメソッドに対応しないことになる。そのような食い違いを生むプログラムの箇所は、バグ位置と関係することが多い。

図 2 のように、ログインシステムにアカウントとパスワードを入力しないままログインしようとする

場合、設計通りだと Compare メソッドを呼び出すに対して、トレース情報に Compare メソッドの呼び出し履歴がない。これで doPost メソッド呼び出しの位置以降にバグがあることを分る。

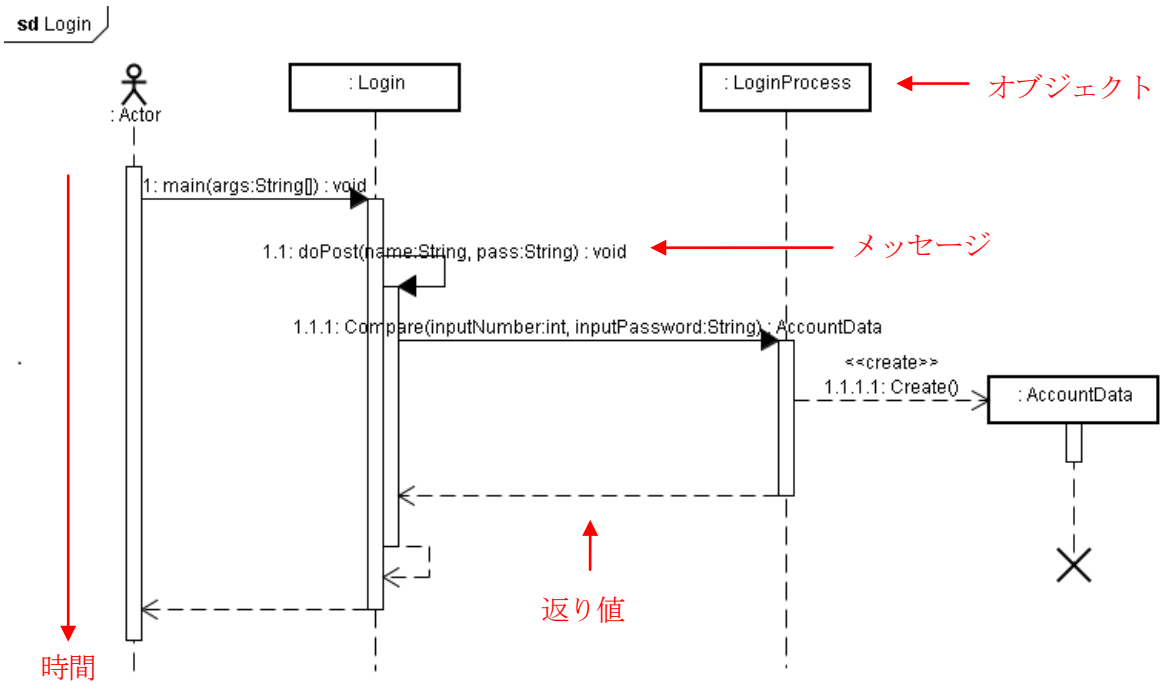

図 1　UML のシーケンス図の例

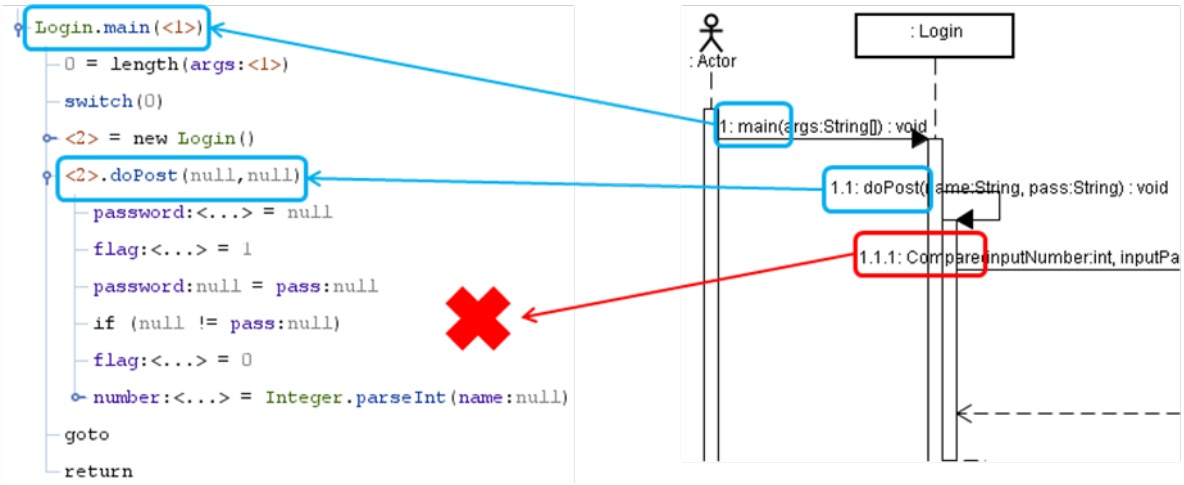

図 2　トレース情報とシーケンス図の食い違いの例

2.3.　トレース情報の問題点

プログラムの規模が大きいとそれに比例してトレース情報も膨大なものとなり扱いにくくなる。設計工程で扱うシーケンス図は人が扱える範囲内でないといけないため、プログラム全体の部分ごとに図がある。しかし一つのシーケンス図に対応するプログラムの範囲は自明でない場合があり、食い違いを発見するため、対応するトレース情報箇所の特定が困難になる。

図 3 のように、ログインシステムに間違ったアカウントとパスワードを入力する場合、実際に Compare メソッドは呼び出されるが、処理が増えると共にトレース情報も増えるため、Compare メソッドをすぐ見付からない。大規模のプログラムの場合、この様な作業はとても手間が掛かると予測できる。

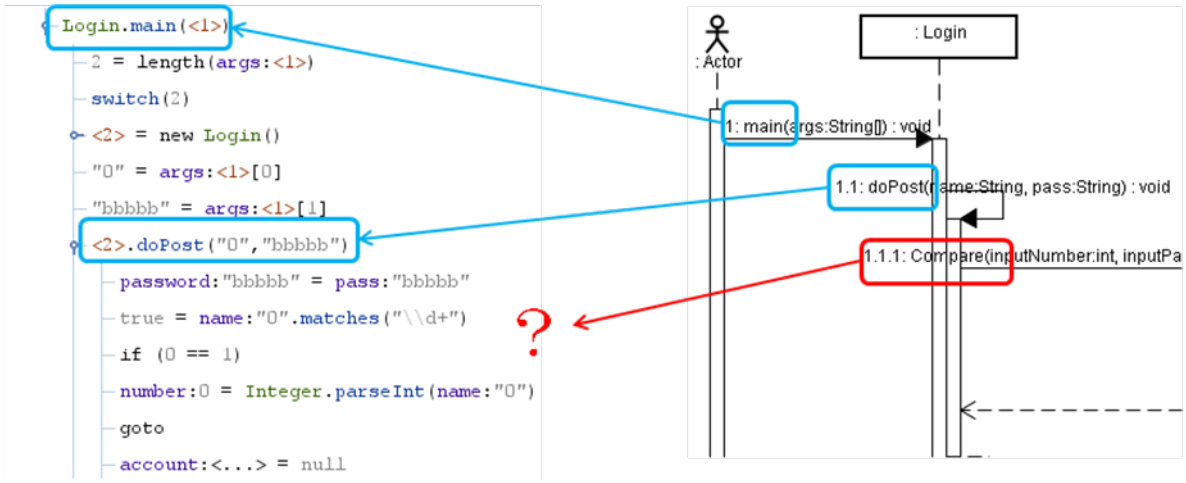

図 3　トレース情報の問題点

3. 提案手法

前述の問題を解決するために、設計工程で扱う UML のシーケンス図とトレース情報の自動対応により、プログラムの欠陥の発見を支援する手法を提案する。

本手法は大きく二つのステップからなる。

第一ステップはシーケンス図の情報とトレース情報を取得する。

第二ステップはシーケンス図の情報とトレース情報を解析する。シーケンス図にあるメッセージの名前は、トレース情報に記録される場合、名前を表示し、範囲を特定するためリンクをつける。メッセージの名前はトレース情報に記録されない場合、食い違いとして名前を表示する。

4. 実装

提案した手法を、UML モデリングツールの Jude / Community と逆戻りデバッガの Traceglasses を利用して、ツールとして実装する。

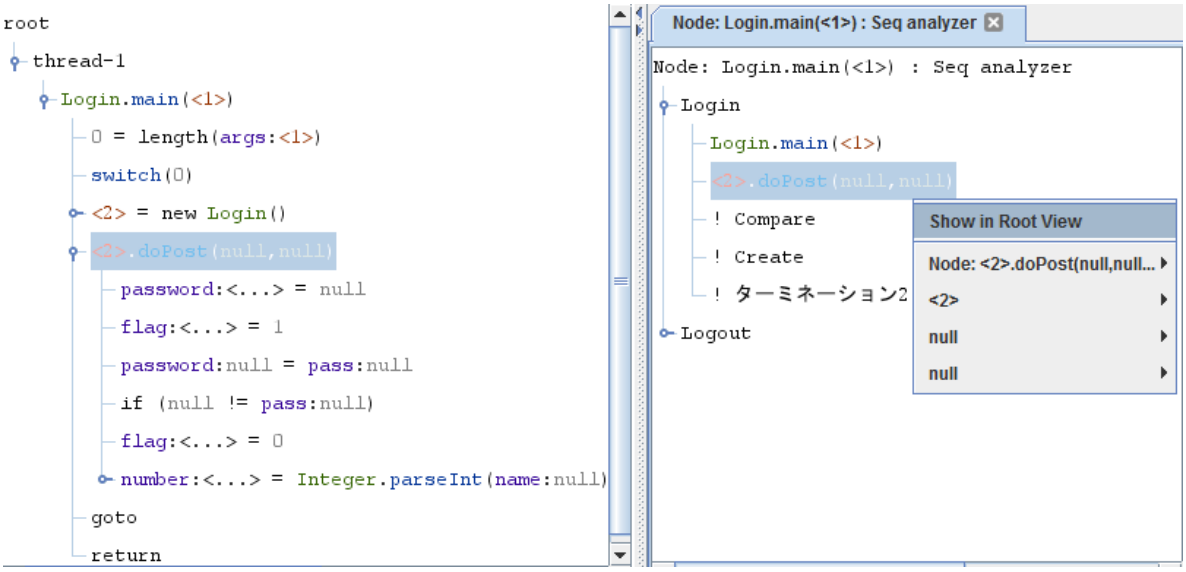

図 4　シーケンス図のプログラム中の範囲の特定

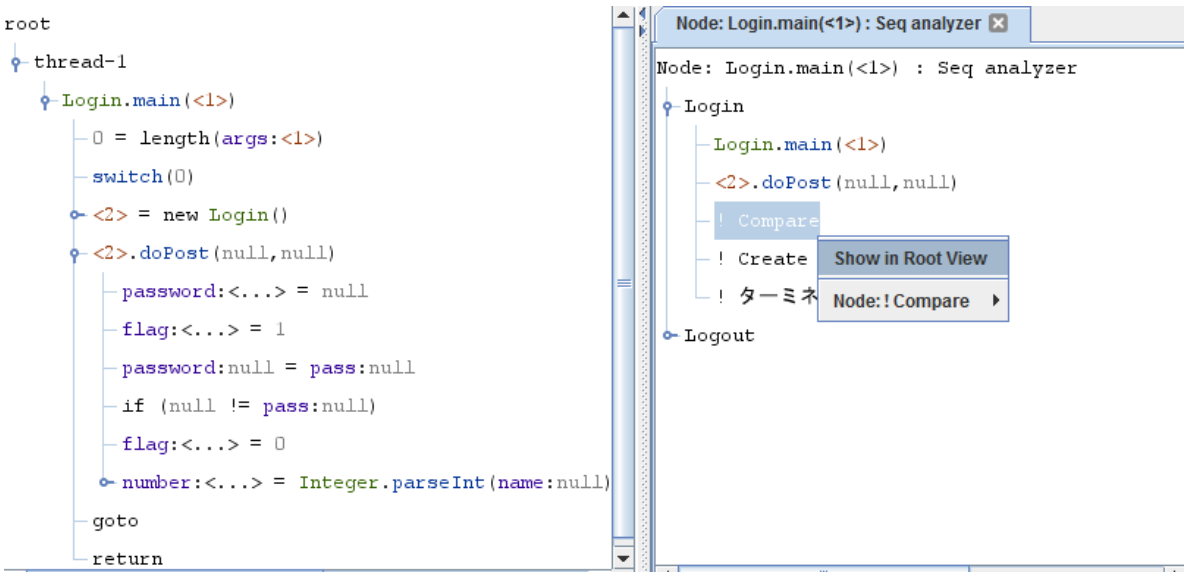

図 5　トレース情報とシーケンス図の食い違いの発見

図 4 と図 5 の左側画面はトレース情報の表示画面、右側画面は実装したシーケンス図との自動対応の結果表示画面。

図 4 のように、左側のトレース情報の中の doPost メソッド呼び出しの位置を知りたい場合、右側の自動対応の結果表示画面の中の doPost を選択して右クリックすると、操作メニューが表示される、操作メニューの中の「Show in Root View」を選択すると、左側のトレース情報表示画面に doPost の行の色が変わる。これでシーケンス図のプログラム中の位置を特定できる。

図 5 のように、シーケンス図の設計通りに Compare メソッド呼び出しが実行されない場合、右側の自動対応の結果表示画面に Compare の前に、「！」を付け加えて食い違いとして表示する。

5. まとめ

本研究ではソフトウェアデバッグの問題点を解決するため、設計工程で扱う UML のシーケンス図とトレース情報の自動対応により、プログラムの欠陥を発見する手法を提案した。

6. 今後の課題

今後の課題としては、メソッド呼び出しの順序を検証し、改善する。また実験を実施する予定である。

参考文献

[1]. Glenford J.Myers, Tom Badgett, Todd M.Tomas, Corey Sandler, 長尾真「監訳」、松尾正信「訳」「ソフトウェア・テストの技法」第 2 版

[2]. 櫻井孝平、増原英彦、古宮誠一「Traceglasses：欠陥の効率よい発見手法を実現するトレースに基づくデバッガ」情報処理学会論文誌プログラミング（PRO）,3(3),1-17 (2010-06-16) , 1882-7802