

GUIプログラムの理解支援のための可視化システムのマルチスレッド対応

佐藤 竜也† 志築 文太郎† 田中 二郎†

† 筑波大学 コンピュータサイエンス専攻

1 はじめに

昨今の GUI プログラムの多くはマルチスレッドによる並列プログラミングがなされている。例えば、アニメーションの描画処理はメイン処理とは別のスレッドで制御されることが多い。それらのプログラムに対して機能追加する場合などにはプログラムの実行の様子を理解する必要がある。その際に複数スレッド間にまたがった処理を理解するには、各スレッドの実行の様子を把握しなければならない。しかしながら、GUI プログラムにおけるマルチスレッドの実行の様子を監視する有効な手段は少なく、それらの挙動を理解することは難しい。

我々はこれまで、Java の GUI プログラムの理解支援を目的とした可視化システムを提案してきた [1]。このシステムでは GUI プログラムの特徴を活かし、GUI への操作に応じたソースコード実行の可視化を行う。本研究ではこのシステムを拡張し、スレッドごとの実行の色分け表示による Java のマルチスレッド GUI プログラムの理解支援を行う。

2 提案する可視化システム

2.1 ベースとなる GUI プログラム可視化手法

GUI プログラムはユーザからの操作に対して処理を行い、画面を更新するという特徴を有する。そのため、本研究では GUI プログラムの操作に対するプログラム実行部分を可視化する。ユーザからの操作がされる度に、操作前後の GUI 画面と操作によって実行されたソースコード情報を併せて可視化する。ソースコード情報としては、実行された部分とその関数呼び出し関係の強調、クラス階層を併せて表示する。

図 1 に以上で説明した可視化手法の模式図を示す。サムネイル表示部において、ユーザからの操作によって起こった GUI 画面変化が強調されている。ソースコード情報表示部では、これらの GUI 画面変化が起こった際に実行されたソースコードが強調表示される。今は簡単のためにソースコードの色付けによる強調のみが行われているが、実際には関数呼び出しとクラス階層

を併せて表示する。このような表示を行うことで、ユーザからの操作に対してどのソースコードが実行されたのかを把握することを可能にする。

また本手法では、各操作に応じて実行されたソースコードをトレースしやすくするためのズーム表示を行う。注目しているクラス、関数呼び出し、ソースコード行を魚眼表示によりズームすることで、ソースコードを実際に読みながらプログラム実行の様子を詳細に把握することが可能である。

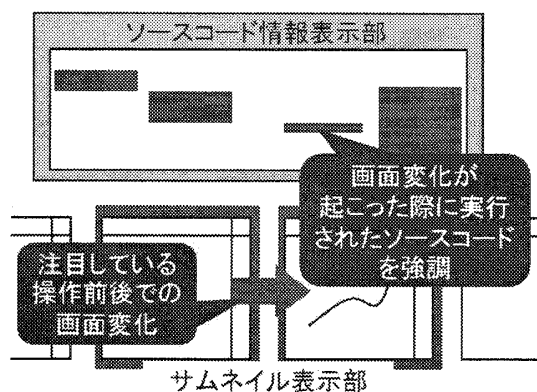


図 1: GUI 操作に対するプログラム実行部分の可視化

2.2 スレッド毎に色づけすることによるマルチスレッド GUI プログラムの理解支援

本研究では、マルチスレッドの実行情報を可視化するために、ソースコードの強調表示をスレッドごとに別々の色で表現する。色分けされたスレッドごとの強調表示を重ね合わせて表示することで、複数のスレッドによるプログラム実行の様子を一画面上で可視化する。

また、マルチスレッドプログラムを理解する目的によってはスレッドごとの実行の様子を見たい場合がある。例えば、アニメーションのフレームレートを調整する際には、アニメーション描画処理部分だけに興味がある。このような処理は複数の操作間にスレッドの実行がまたがることもあり、操作ごとに区切られた可視化結果を閲覧するだけでは把握しづらい。そのためスレッド別に表示非表示を指定可能にし、ユーザが閲覧したいスレッドの実行だけを強調できるようにする。その際には、ユーザによって選択されたスレッドが実

Supports for Understanding Multi-threaded GUI Programs with a Visualization of Execution

†Tatsuya Sato †Buntarou Shizuki †Jiro Tanaka

†Department of Computer Science, University of Tsukuba

行したソースコード行の強調と、そのスレッドが実行された期間の GUI 画面サムネイルを強調を行う。これにより、スレッドが実行されたタイミングを明らかにしながら、一つ一つのスレッドの実行の様子を把握することができる。

図2にシステムがマルチスレッド GUI プログラムの実行を可視化している様子を示す。ここで可視化の対象プログラムは図形を描画可能なドローツールである。このプログラムでは描画処理が別スレッドにて行われるようにマルチスレッド化されている。図2では、ドローツールがキャンバスの再描画イベントを処理している場面での実行の様子を表している。図中では、ユーザ操作イベントによる再描画要求とそれに対する描画処理の二つのスレッドが実行されており、それぞれのスレッドが別の色で表現されている。これらの表示に加えて、スレッドごとの表示非表示を切り替えることで、二つのスレッドのそれぞれの実行部分のみを閲覧することもできる。

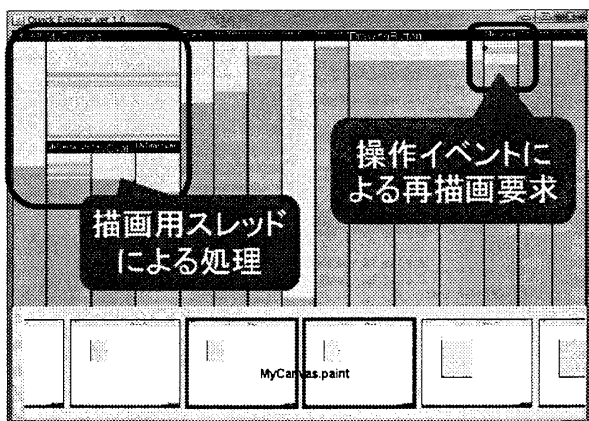


図2: マルチスレッドプログラムの実行情報の可視化

またソースコードをトレースしやすくするズーム表示もスレッドごとに行えるようにする。マルチスレッドプログラムでは、スレッド間の同期や実行のタイミングが重要な情報を含んでいる場合がある。例えば、スレッド同士でメッセージパッシングを行うような場合が挙げられる。そのため、一つ一つのスレッドに対して別々にズーム表示を行うとこれらの情報が欠落しまう危険がある。そこで本研究では、スレッドの数だけのズームのフォーカスを持たせることで、すべてのスレッドの実行の様子を同時にズームする。

図3にズームを行っている様子を示す。このように複数のスレッドの実行点が同時にズームされる。

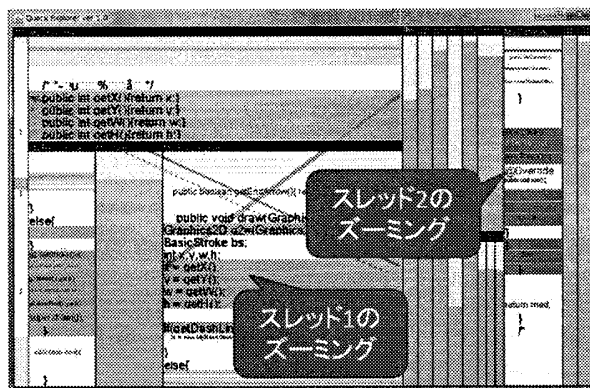


図3: マルチスレッドプログラムのズーム表示

3 関連研究

本研究と特に関連のある、動的解析に基づく Java マルチスレッドプログラム可視化システムについて述べる。[2]ではスレッドごとに複数のウィンドウを用いてプログラムの実行の様子を可視化している。それに対して、本手法では単一ウィンドウ上での可視化を行っている。[3]では動的情報取得時のオーバーヘッドの少なくすることを目標としており、クラスレベルでのプログラムの様子を可視化する。それに対して、本研究ではソースコードレベルでのより詳細な情報を可視化する。

4 まとめ

本研究では GUI プログラム理解支援のための可視化システムを拡張し、マルチスレッド GUI プログラムの理解支援のための可視化手法を提案した。本システムでは、スレッドごとの実行情報を色分けし、それらを重ね合わせて表示することで、一画面上でのマルチスレッド可視化を行う。本システムを利用することで、GUI との同期を取りながらスレッドごとの挙動を把握することが可能となる。

参考文献

- [1] 佐藤竜也, 志築文太郎, 田中二郎: 実行の可視化システムと連動した統合開発環境による GUI ベースプログラムの理解支援, WISS2007 予稿集, pp. 25–30.
- [2] 山岡裕司, 寺田実: Java マルチスレッドプログラムのためのアニメーションツール, インタラクション 2001 予稿集, pp. 63–64.
- [3] Steven P. Reiss: Efficient Monitoring and Display of Thread State in Java, Proceedings of IWPC2005, pp. 247–256.