

6K-01

オブジェクト指向プログラムにおける実行時ログの3次元表現とその操作

3D Representation and Manipulation of Runtime Logs in Object-Oriented Programs

鮎川拓実*¹ 深澤良彰*¹
Takumi Ayukawa Yoshiaki Fukazawa

*¹ 早稲田大学基幹理工学部情報理工学科
Department of Computer Science and Engineering, Waseda University

1.はじめに

大規模ソフトウェア開発におけるテスト、デバッグ、保守等において、ソフトウェア理解は大変重要である。ソースコードの動的な振る舞いを理解する時に、実行時ログは有用なデータとなる。しかし、大量の変数の変遷を表す場合、ログが長大になりその理解が困難な点がある。そこで本研究では、実行時ログの理解容易性の向上を目的とする。

特に本発表では、使用頻度の多い条件分岐、ログが長くなる大きな要因の繰り返し処理に焦点を絞る。

このために、より多くの情報を直感的に表現できるVR技術を利用する。3次元による可視化は特に複雑なモデルを扱う場合に非常に有効であるとされている[1]。

本研究ではVR機器として、Meta Quest2を使用している。また、後述する3次元表示システムはUnity(2020.3.16f1)を使用し、言語はC#を使用している。さらに、VRの機能へと拡張するために、Mixed Reality Toolkit[2]を利用した。

入力のプログラミング言語としてはC#を用いているが、C#に限らず他のプログラミング言語でも同様に考えることができる。

2.本研究の特徴

本研究の特徴として、2次元空間内で表されるソースコード、ログデータに時間軸を追加して考え、3次元空間で表現する。そして、その3次元空間で変数の変遷を表現できるようにした。

2.1.条件分岐におけるプログラムとログの対応関係の明確化

プログラムとログの対応関係を明確化するために、条件分岐によって実行される部分と実行されない部分を彩色により区別することとした。例えば、if文では図1のように表す。

具体的にはif文の条件を満たさず、実行されない部分のコードは灰色にすることで実行される部分と実行されない部分を差別化している。

また、図1のプログラムにおいて、for文の最後の繰り返し以外ではreturn文も実行されていない。このように実行されていない部分も同様である。

3次元で表現された各コードの側面には、プログラムの実行によって書き換わった変数の値を表示するようにしている。これによって、それぞれの変数の変遷を容易に理解できる。

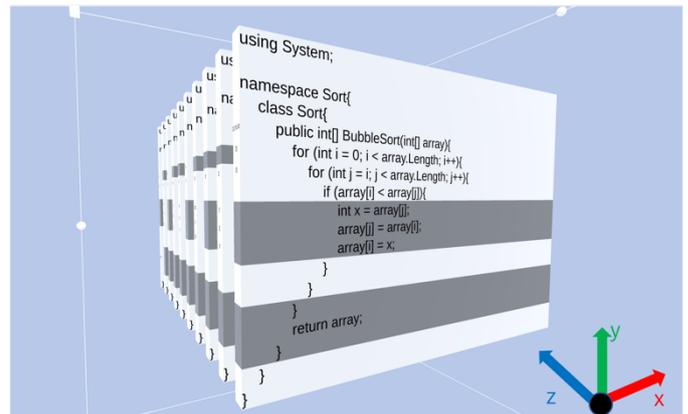


図1 if文の表現

2.2.繰り返し処理におけるログデータの3次元表現

ログデータを大量にする大きな原因の1つは繰り返し回数の多いfor文である。for文の理解においては大きく以下の2つの方法が役立つ。

- (1)ある繰り返しとその前後の繰り返しにまたがる値の変遷を理解すること
- (2)ある繰り返しにおけるプログラムの振る舞いを理解すること

(1),(2)を実現するために、例えばfor文では以下の図2のように表す。

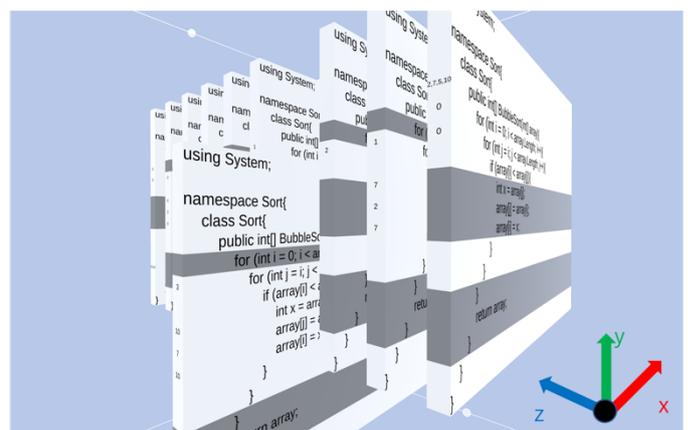


図2 for文の表現

for文の繰り返される数だけz軸正の方向にコードを表現し、それぞれの層の変数の変遷や前述の条件分岐の実行を比較できるようにした。例えば、図2

では最初の繰り返しではjの値が0であるが、次の繰り返しではjが1となることが分かる。また、同様に最初の繰り返しではif文の内部のコード灰色になっているため、実行されていないことが分かるが、次のコードでは値が代入されていることがわかる。これによって(1)の「ある繰り返しとその前後の繰り返しにまたがって値の変遷を理解すること」を視覚的に実現できるようにした。

また、(2)の「ある繰り返しにおけるプログラムの振る舞いを理解すること」を実現するために、図2のように、コントローラーで操作することで層ごとにコードを取り出せるようにした。図2においては、4回目の繰り返しに注力するために、その層を取り出した場面を示している。

コードを取り出す際にはz軸方向の移動および回転はできなくしているため、前後の層と干渉することはないようにしている。このようにすることで、前後にある層によってコードが見えなくなっている場合でも、コードを見ながらその繰り返しにおける変数の変遷を理解できるようにした。これによって(2)の実現をした。また、断面に代入される値の変遷を表現した。

3.本システムの構成

本システムの構成を以下の図3に示す。

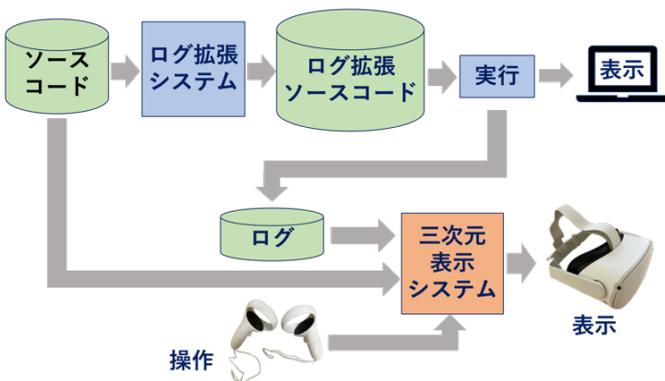


図3 システムの構成図

全体としては、ソースコードを入力とし、実行、標準出力という流れとなる。ただし、実行する際に、変数の変遷のログを出力することを想定している。そして、ソースコードとログを入力とし、本研究の三次元表示システムを実行し、VRヘッドセットで表示、コントローラーでの操作を可能とする。

以下では図3の中の三次元表示システムの部分の詳細について述べる。ソースコードを構文解析することで、クラス名、メソッド名、クラス間の関連、継承関係などを取得する。構文解析にはオープンソースのツールであるRoslyn[3]を利用している。そして、構文解析で得られた各種情報、ソースコード、ログをデータベースへ格納する。そして、これらから3Dオブジェクトを生成する。クラス図やクラス図の中にあるソースコード、そして変数の変遷などを表現するようにする。そして、最終的に3次元空間内での出力をし、ユーザーはコントローラーを利用することで操作可能とする。

4.実行例

ここではバブルソートを例に説明する。

図1及び図2はバブルソートのプログラムと配列{2,7,5,10}と入力した時のログデータを与えた場合の出力結果である。このソースコードを見たい場合には、正面から見ると図1のようになる。一方、値の変遷を見たい場合には、斜めから見ると図2のようになる。

z軸方向から見て、最も手前に存在する層がここでは $i=0, j=0$ のループを表しており、取り出されている4層目が $i=0, j=3$ を表している。

$i=0, j=0$ の時は値の入れ替えのコードは実行されないため、1層目ではその行は灰色となっている。

$i=0, j=3$ の時の4層目は値10,7の入れ替えが発生する。xに10が代入され、その後array[3]に7、array[0]に10が代入されることが分かる。

その他の行も同様にログデータから値の変遷の情報を取得し、3次元上で表示している。

必要な部分のコードをより詳細に見る場合には、図2のようにコントローラーを利用して動かすことができる。

5.おわりに

本研究では実行時ログの理解容易性の向上という目的を達成するために、三次元表示システムの開発を行なった。実現したこととして以下の2点が挙げられる。

- ・条件分岐におけるプログラムとログの対応関係の明確化
- ・繰り返し処理におけるログデータの3次元表現

今後の展望は大きく3点挙げられる。

1点目は、評価についてである。2次元の場合と今回の3次元のシステムの場合で正確性や効率性の比較を行うことで、有用性を検証する必要がある。

2点目は現時点对応していない文におけるソースコードの対応である。例えば、再帰呼び出しを含む関数呼び出し文、1行の中で複数の変数の値が変化する文などが挙げられる。

3点目は、クラス図などオブジェクト指向に関する文法と、ソースコード、実行時ログの一体的表現の実現が挙げられる。具体的には、クラス図の表現やクラスをインスタンス化した時の表現などが挙げられる。

6.参考文献

[1] Joacim Alvergren, Jonatan Granqvist, “3D Visualization of Complex Software Models: Practical use and Limitations”, IEEE Trans. on Visualization and Computer Graphics, Vol15, 87-105, 2019.
 [2] Microsoft, “microsoft/MixedRealityToolkit-Unity”.
<https://github.com/microsoft/MixedRealityToolkit-Unity>. (参照 2022-12-26).
 [3] Microsoft, “.NET Compiler Platform SDK”.
<https://learn.microsoft.com/ja-jp/dotnet/csharp/roslyn-sdk/>. (参照 2022-12-26).