

1. はじめに

本研究では、プログラムのソースコードの構成と変更の履歴を可視化する。情報可視化とはコンピュータグラフィックスを利用してデータを表現する視覚技術のことである。

大規模なソフトウェアの開発には、多数の開発者が関わっており、特にオープンソースソフトウェアの開発では、その仕様や構成が変化していくので全体の構造を把握するのが難しい。

本研究は、このようなソフトウェアでもその全体構造を把握しやすく表示し、時間経過の経過に伴って、どの部分が頻繁に改変されているかなどをわかりやすく表示することを目的とする。

2. 可視化手法の検討

本研究で可視化する手法はシムシティのような全体を見渡せることができ、時間的変化も対応できるようなソフトウェアにする。

シムシティ (図 1) は箱庭の中でリアルに再現される「社会」で、プレイヤーが「市長」として自分の都市を作る「都市開発シミュレーション」ゲームである。

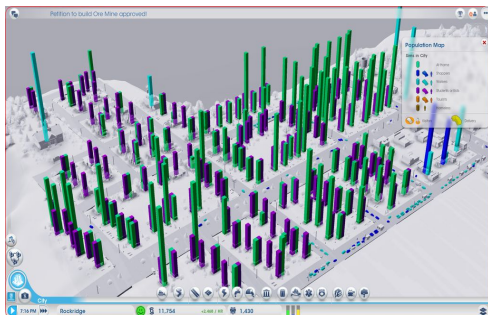


図 1 シムシティ

ツリーマップ (図 2) とは 2 次元平面上の領域を入れ子状に分割することによって、木構造の

データを可視化する手法である。

(<http://www.cs.umd.edu/hcil/treemap-history/>)

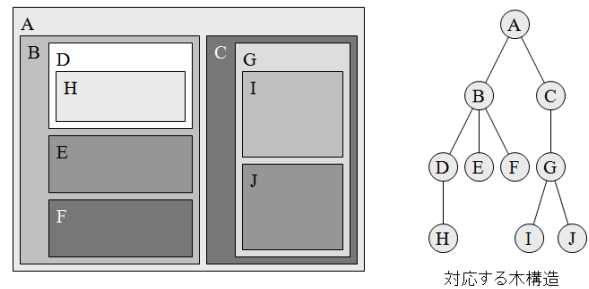


図 2 ツリーマップ

4. 提案する可視化手法

シムシティのようなオープンソースのソフトウェア全体をひとつの街と考え、ツリーマップで可視化する。測定したいプログラムまでのフォルダの流れを、ツリー構造を用いて実現させる。ツリー構造とはデータ構造の一つで、ひとつの要素が複数の子要素を持ち、更にその要素が子要素をもつという形で階層が深くなるほど枝分かれしていく構造のことである。親要素の中に子要素が入っているのが目で見てわかりやすく可視化できるため、木構造を使用したツリーマップが適していると考えられる。

本研究では 2 次元上ではなく、3 次元の立体の領域に分割して可視化を行う。

4. ソースコードの情報の取得方法

本研究では、ソースコードからプログラムの全体の構造と各モジュールの特徴、さらに更新履歴などを分析して可視化するソフトウェアを開発する。ソースコードを可視化するために、ソフトウェアメトリクス算出プログラムを用いてソースコードの特徴量を求める。

SourceMonitor (<http://www.campwoodsw.com>)

ではコードの複雑さやネストの深さ、コードの行数、ステートメント数、循環的複雑度などを測定することができる。

測定対象のプログラムは、オープンソースのソフトウェアを用いる。オープンソースソフトウェアは、ソースコードを閲覧することができ、多数の開発者が関与しており、バージョンの履歴を追跡することができるので、本研究の対象として適している。

5. 開発環境

本研究で使用した言語は Processing である。Processing とはオープンソースソフトウェアであり、Java を単純化して電子アートとビジュアルデザインのために使用されるプログラム言語である。今回の研究では可視化を目的としているのでほかのプログラム言語より、グラフィック機能に特化した Processing が有効である。

プログラムを数値化するソフトウェアは SourceMonitor を使用し、CSV ファイルで保存する。保存する内容は ID:ファイル名, PARENT:親のファイル名, COL:プログラムの行数, SIZE:サイクロマティック複雑度である。

7. ツリーマップのアルゴリズム

ツリーマップを作成するには、最初に全体の領域を決めるのではなく、末端の子要素の領域から決め、それらを足し上げていき、最後に全体の領域を決める。しかし、各要素の領域をそれぞれ計算するのは効率が良くないため、再帰的アルゴリズムを使用する。そうすることで自分が持つ子要素の計算だけを考えれば良いとなる。座標の計算式は以下の通りである。

$$\text{矩形の座標} = \text{隣の領域の座標} + \frac{\text{各領域}}{\text{全体の領域}}$$

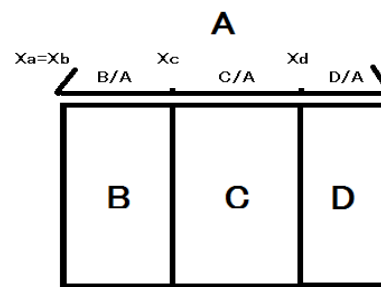


図3 領域の分割方法

図3より、始めのX座標が X_a とする。X軸の全体の領域がAとするとその子要素となっているBの右側のX座標は X_a+B となる。このように全体の領域から各子要素の領域を割ることでツリーマップができ上がる(図4)。

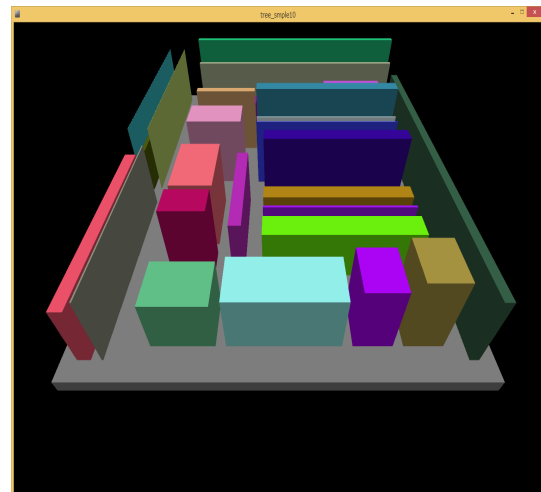


図4 実行結果

9. まとめ

本研究ではオープンソースのソフトウェアを用いてステートメント数やサイクロマティック複雑度などを測定し、3次元空間のツリーマップとして可視化した。現状では測定したいオープンソースのソフトウェアを、一旦別のソフトウェアで測定し、本研究のソフトウェアを用いて可視化を行っている。今後の展望は測定するソフトウェアと本研究を別々で使用するのではなく、ひとつのプログラムとして作成することと、更新履歴などから時間的に変化するツリーマップの作成である。