

タンジブルUIを使った Sound Coupling と Cohesion Coupling によるソフトウェアの再構性

岡崎 知也[†] 吉田 誠[†] 岩根 典之[‡]

岡山理科大学工学部[†] 広島市立大学情報科学部[‡]

1 研究目的

近年ソフトウェアは益々高度化・複雑化し、仕様変更などに柔軟に対応するためのソフトウェアの再構性が重要になっている。ソフトウェア再構性には、様々な指標が考案されている[2]。しかし、これらの指標を利用して短時間でプログラムの評価・修正を行うのは容易ではない。著者らは音・色・感覚といった属性を利用して簡単にソフトウェア再構性を行うことができる、ソフトウェアリエンジニアリングツールキットを開発した[3]。ツールキットは以下の方針で開発された。

- ・ソフトウェア再構性の操作が簡単である
- ・指標を理解しやすい
- ・(再構性後の) 評価の仕方が簡単である
- ・何度も繰り返し再構性を行える
- ・簡単に元に戻せる

本論文ではツールキットを使用した Sound Coupling と Cohesion Coupling によるソフトウェアの再構性について記述する。

2 研究概要

2.1 システム概要

TUI(Tangible User Interface)は、感触という概念をとり入れたインタフェースである。図1に本研究で使用した感触装置 NOVINT FALCON を示す。図2に、感触装置を使用したソフトウェア改善ツールキットのシステム構成を示す。

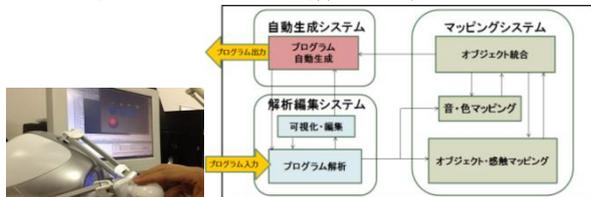


図1 NOVINT FALCON 図2 ツールキットのシステム構成

以下にプログラムが再構築されるまでの処理の流れを示す。

- ① プログラムを実行してプログラム構造

を把握する (AspectJ を使って実行履歴からアクセス構造を作成する)

②解析編集システムでプログラムの特性 (クラス・ライン数など) を分析し、表示する

③マッピングシステムでプログラム特性を感触装置属性 (音, 色, 感覚など) にマッピングしてオブジェクトとして配置する。

④目標の指標値に近づくようにオブジェクトを統合する。

⑤統合されたオブジェクトのデータを自動生成システムに渡し、再構成後のプログラムを自動生成する。

2.2 Sound Coupling と Cohesion Coupling

マッピングシステムでの2つの統合方法について説明する。

Sound Coupling では、ソフトウェア再構性の指標を音程にマッピングし、音程を頼りに統合を行う。図2にマッピングシステムで Sound Coupling を行う様子を示す。図の各球体はクラスを表している。

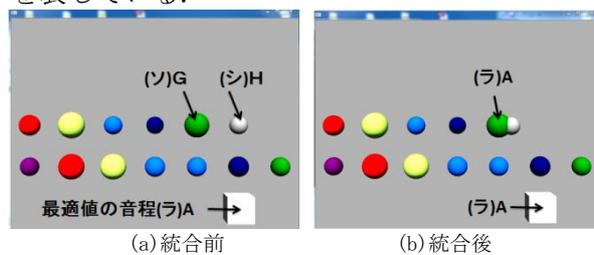


図2 Sound Coupling

それぞれのクラスの音程を鳴らして確認し、目標値の音程より低い音程のクラスは高い音程のクラスと統合することで音程を高くし、目標値の音程に近づける。この作業を繰り返すことで指標を目標値に近づけ、ソフトウェアの再構性を行う。

Cohesion Coupling ではアクセス関係をオブジェクト間の距離で表現し、アクセス関係の密なクラス同士を統合する。式(1)は、距離を基にした凝集度測定[1]による、類似性に基づいた2つの実体の距離を表す式である。

$$dis(x,y) = 1 - \frac{|b(x) \cap b(y)|}{|b(x) \cup b(y)|} \dots (1)$$

Software Reengineering by Sound Coupling and Cohesion Coupling through Tangible User Interface

Tamoya Okazaki[†] Makoto Yoshida[†] and Noriyuki Iwane [‡]
[†]Okayama University of Science Okayama, Japan
[‡]HiroshimCity University Hiroshima, Japan

式中の x, y は対象とするクラス, b はアクセス関係のあるメソッドを示している. 式(1)でクラス間の距離を求めた後, 多次元尺度構成法[2]で2次元上に配置することにより, ソフトウェアの再構性を行なっている. これにより近い距離のクラス同士を統合することができる.

3 実験

3.1 指標

本実験では, ライン数/メソッドの平均値を指標1(目標値10), メソッド数/クラスの平均値を指標2(目標値7)としている.(指標の目標値は文献[2]に記載されているデータを参考にして定めた)

3.2 実験概要

Sound Coupling と Cohesion Coupling を行い, 両指標を比較した. 図2に実験対象ソフトウェアのプログラムの構造(実行履歴から作成)を示す.

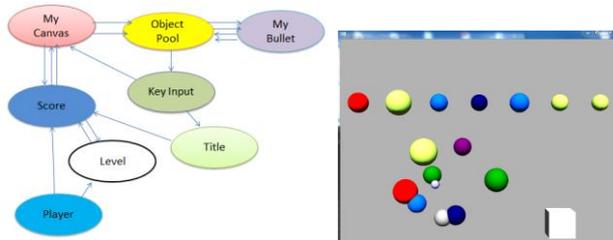


図2 プログラム構造

図3 クラスのアクセス関係の可視化 (マッピングシステム)

図の矢印は, アクセスがあることを表している. 図3はプログラムのアクセス関係を可視化している. 図の各球体はクラスを表している. アクセス関係が密なクラス間は距離が近く, 疎なクラス間は距離が遠く配置されている.

以下に, 実験対象ソフトウェアにアクセス関係の可視化を行うまでの流れを示す.

- (1)ソフトウェアの実行履歴を AspectJ を用いて解析する.
- (2)式(1)を用いて各クラス間の距離を求める.
- (3)求めた距離をもとに多次元尺度構成法によってオブジェクトの配置を行う.

3.3 実験結果

表3に実験対象ソフトウェアの解析データを示す. 表4に Sound Coupling によって再構性した後のデータを示す. 表5にクラス分割を行った場合と行わなかった場合の二通りの Cohesion Coupling の結果を示す.

Sound Coupling では各クラスが統合され, クラス数が 13→10 となった. 指標1は 7.2→7.2 となり指標2は 5.2→6.8 となった. また, Cohesion Coupling1 では指標1が 7.2→7.3,

表3 解析データ

クラス	メソッド数	ライン数	ライン数/メソッド
Bullet	6	48	6
Enemy	7	109	13
Game	2	21	9
GameObject	2	9	1
KeyInput	6	91	13
Level	3	18	4
MyBullet	4	23	4
MyCanvas	7	101	11
Objectpool	12	149	10
Particle	4	41	7
Player	4	33	6
Score	8	43	3
Title	3	29	7
平均	5.2	55.0	7.2

表4 Sound Coupling

クラス	メソッド数	ライン数	ライン数/メソッド
Bullet	6	48	6.0
Enemy	11	132	8.5
MyBullet	3	28	7.0
GameObject	8	100	7.0
Level	4	28	6.0
Objectpool\$1	15	144	7.0
MyCanvas	10	134	11
Objectpool\$0	4	41	7.0
Particle	4	33	6.0
Player	3	29	7.0
Title	3	29	7.0
平均	6.8	71.5	7.2

表5 Cohesion Coupling 1

クラス	メソッド数	ライン数	ライン数/メソッド
Bullet	6	48	6
Enemy	7	109	13
Game	2	21	9
GameObject	2	9	1
KeyInput	6	91	13
MyCanvas	11	134	8.5
Player	17	186	6.5
Objectpool\$0	1	10	8
Title	1	7	5
Objectpool\$1	4	41	7
Objectpool\$2	4	41	7
Particle	4	41	7
Score	11	61	3.5
Level	11	61	3.5
平均	6.1	55.0	7.3

Cohesion Coupling2

クラス	メソッド数	ライン数	ライン数/メソッド
Bullet	6	48	6
Enemy	7	109	13
Game	2	21	9
GameObject	2	9	1
KeyInput	6	91	13
Title	9	120	10
Level	11	61	3.5
MyBullet	16	172	7
Objectpool	7	101	11
Objectpool	12	149	10
Particle	4	41	7
Player	4	33	6
平均	7.2	78.5	7.8

指標2は 5.2→6.1 となった. CohesionCoupling2 では指標1は 7.2→7.6 となり指標2は 5.2→7.2 となった. 各指標とも目標値に近づいているのが観測された.

4 まとめ

音による統合とアクセス関係による統合を行い, 比較評価した. 今回の実験では各統合方法とも指標が目標値に近づいた. また, アクセス関係による統合は距離の近いオブジェクト同士を統合するというより単純な操作であるため, 短い時間で簡単に統合作業を終えることができた. Texture などへのプログラム属性のマッピングと統合は今後の課題である.

参考文献

- [1]Frank Simon, Silvio Loffer, Claus Lewerentz, Distance Based Cohesion Measuring, FESMA99.
- [2] Michel Lanza/Radu Marinescu Lanza, M., Marinescu, R., 2006. Object oriented metrics in practice, New York.
- [3]Makoto Yoshida, Shinnta Okumura and Noriyuki Iwane, Software Reengineering Toolkit with Tangible interface by Haptics, ICSOFT2014, pp351-356