

凝集度と結合度に注目する OOD 学習支援システム Fourcs の提案

杉浦 啓孝[†]松澤 芳昭[‡]酒井 三四郎[‡]静岡大学大学院情報学研究科[†]静岡大学情報学部[‡]

1. はじめに

オブジェクト指向に基づいたソフトウェア開発が行われている中で、知識、経験が乏しい初学者がオブジェクト指向設計(OOD)を行うことは難しく、学習や支援環境が必要となる。

本研究では OOD 初学者を対象にした、オブジェクト指向に基づく設計を学習することを目的としたシステムの開発を行なう。

2. 関連研究と本研究の位置づけ

初学者が OOD を行うために必要な知識の 1 つに GRASP^[1]があり、このパターンを適用させ、学習するシステムの評価を行っている^[2]。作成したシステムを用いて、各パターンの学習を促すことはできたが、初学者にとってシステムが提示するアドバイスが難しく十分に学ぶことができなかつたという問題点が挙げられる。

そこで、本研究では初学者にも理解できるアドバイスとして、凝集度と結合度を示し、これらに着目して設計を行わせることで、OOD 学習を支援するシステムを提案する。

3. 本システムにおける凝集度と結合度

本システムで用いる凝集度、結合度を定義する。本システムにおける凝集度は LCOM*^[3]を用いる。LCOM*はクラスにおける操作と属性の関係の強さを測り、関係のある操作、属性がまとめられているかを示す。このため、操作、属性の配置を検討するために効果的である。

本システムにおける結合度はクラス図のクラス間関係と、コミュニケーション図の操作の呼び出し関係から求める。クラス間関係は関連の種類と参照先のクラスの種類によって分類し^[4]、表 1 のように結合の強さを決定する。ここで、本システムでは参照の方向を明確にするため、関連を利用しない。また、集約と合成を明確に区別して利用することは初学者に難しいので、合成を利用しない。

次に、コミュニケーション図における結合度を定義する。この図では、メッセージの呼び出し関係とそれに伴う引数、返り値におけるデータの交換によって結合度を求める。このとき、あるクラス間で、メッセージの呼び出し関係が明確になるほど、結合度が小さくなるよう定義する。ある 2

Fourcs:OOD Learning System Focused on Coupling and Cohesion

[†]Hirota Sugiura, Graduate school of informatics, Faculty of Informatics Shizuoka University

[‡]Yoshiaki Matsuzawa, Sanshiro Sakai, Faculty of Informatics, Shizuoka University

つのクラスに注目したとき、呼び出されるメッセージにおけるデータ交換によって重みを決定し、同クラス間で複数のメッセージが呼び出されている場合は、その重みを合計する。その和をクラスの持つ操作数で割り、この値を小さくする度合いとなる(式 1)。データ交換による重みは表 2 のように定義する。

以上で述べた、クラス図とコミュニケーション図から得られる値を掛け合わせ、2 クラス間の結合度とする。

表 1:クラス間関係の結合

	インタフェース	クラス
誘導可能性	1/6	3/6
集約	2/6	4/6
汎化・実現	5/6	6/6

$$1 - \frac{\sum(\text{データ交換による重み})}{\text{メソッド数}} \quad (\text{式 1})$$

表 2:データ交換の結合

	なし	プリミティブ	オブジェクト
単一	1/5	2/5	4/5
配列		3/5	5/5

4. OOD 学習支援システム Fourcs

本研究の目的は、UML を一通り学習した OOD 初学者に対して、凝集度と結合度に着目させ OOD 学習支援を行うことである。学習者は本システムで設計図としてクラス図とコミュニケーション図を作成する。システムはこの 2 つの図が書き換えられる度に結合度、凝集度を計算し、提示する。学習者は各クラスの凝集度、クラス間の結合度を手がかりに、OOD として欠点のある箇所を見つけ出し、「良い設計」へと修正する。これらのことを行うために、次の機能を提供する。

- (1) 操作属性参照関係エディタ
- (2) 変更履歴

4.1. 操作属性参照関係エディタ

本来、LCOM*はコーディングされてから計算できるメトリクスであり、設計図から求めることはできない。そこで、1 つのクラスに割り当てられた操作が、同じクラスのどの属性を参照するのかわ、学習者がグラフィカルに決定するエディタを提供する。LCOM*は、属性がどの操作に参照されているか決められていれば求めることができるので、このエディタで参照関係を定めることで、LCOM*を求めることができる。また、参照されていない、

複数の操作で共有されていない属性がある場合や、属性、操作の配置に問題がある場合も、参照関係を見ることで発見が容易になり、操作、属性の移動やクラスの分割などの修正が行いやすくなる。

4.2. 変更履歴

クラス図、コミュニケーション図でモデルの追加、削除、修正、凝集度や結合度が変化したときに、それ以前の設計を残しておく、設計の履歴を保存する。これによって、ある時点から分岐して設計を行うことや、ある時点とある時点での設計を見比べることができる。学習者は、複数の設計を行うことや比較することが容易になり、複数の設計から、良い設計がどれかを検討することができる。

以上の機能を持ったシステムを提案し、開発したシステムのユーザインタフェースは図 1 のとおりである。

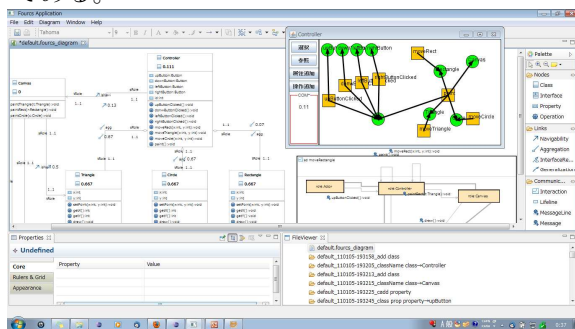


図 1:Fourcs ユーザインタフェース

5. 評価実験

5.1. 実験方法

OOD 初学者である大学生 6 名に評価実験を行った。評価実験の方法は、クラス数 3 から 5 個の設計図を 2 つ用意する。両方とも、改善可能な点がある。対象者をランダムにグループ A とグループ B に分け、グループ A は設計図 1 で Fourcs を用いて修正を行い、2 つ目の設計図 2 を提案した機能や凝集度、結合度が表示されないエディタを用いて修正を行う。グループ B は利用するエディタの順番を逆にし、設計図 1 を普通のエディタで、設計図 2 は Fourcs を用いて修正を行う。グループ A,B ともに 2 つの実験後アンケートを行う。そのアンケートと修正作業の様子から評価を行う。

5.2. 実験結果

グループ A では、Fourcs の方が改善すべき点を見つけやすく、操作属性参照関係エディタを用いて、操作と属性の参照関係の把握ができ、不適切な配置の操作、属性が分かるという結果であった。しかし、グループ B では Fourcs を用いても、改善すべき点を見つけることがそれほど容易でなく、操作属性参照関係エディタを用いて、参照関係の把握はできるが、不適切な配置の操作、属性は分かりにくいという結果になった。また、履歴を利用した人はグループ A に 2 人、グループ B に 3 人

おり、設計の比較、元の設計との変更点を確認するという作業を行っていた。

Fourcs と普通のエディタを比較したとき、どちらのエディタのほうが、凝集度と結合度に注目して改善を行いやすいかという質問では、6 人が Fourcs であると答えている。この理由として、値が表示されていることで目安にすることができるという意見や、設計の段階でこれらをチェックでき、改善しやすい、操作と属性の参照関係を決めることで、実際のプログラムの動きが想像しやすかったという意見が挙げられた。

元の設計と比較して、改善後の設計を良い設計として判断した理由は、Fourcs を用いた場合、凝集度、結合度の観点から判断した人が 5 人いた。一方で、普通のエディタの場合、凝集度、結合度の観点から判断した人は 1 人のみで、他 5 人はクラス数を少なくしたことや、多態性を用いたことを理由に挙げていた。

6. おわりに

実験結果から、グループ A は、操作属性参照関係エディタや履歴機能を効果的に利用して、OOD 学習を行っていたと判断できる。一方で、グループ B においては改善すべき箇所や、操作属性参照関係エディタを利用しても不適切な属性や操作を見つけることが難しいという結果であった。この原因として、実験で利用した設計図の理解が難しかったためと思われる。

しかし、両グループとも Fourcs を用いた場合、凝集度と結合度に注目させることができ、改善前と改善後でこれらの値を比較し、どちらが良い設計であるかを検討していたので、グループ B においても OOD 学習は行われていたと判断できる。

今回の研究では、凝集度に関する操作属性参照関係エディタを提供したことで、被験者の多くが、凝集度だけに注目してしまった。そのため、結合度に関しても、クラス図とコミュニケーション図から決まるのではなく、学習者自ら凝集度が決まる際と同じような操作を行うことで結合度が変化する機能も提供できていれば、結合度にも注目することができていたと思われる。

参考文献

- [1] クレーグ・ラーマン (依田 光江 訳, 今野 陸, 依田 智夫 監訳)“実践 UML パターンによる統一プロセスガイド第 2 版”, ピアソン・エデュケーション (2003)
- [2] 杉浦啓孝, 酒井三郎, “GRASP を利用するオブジェクト指向設計学習支援システム”, 第 8 回情報科学技術フォーラム講演論文集, pp.525-526 (2009)
- [3] Brian Henderson-Sellers, “Object-Oriented Metrics: Measures of Complexity”, Prentice-Hall, pp.142-147 (1996)
- [4] M.Hitz, B.Montazeri, “Measuring Coupling and Cohesion In Object-Oriented Systems” In Proc. Intl. Sym. On Applied Corporate Computing, pp.1-10 (1995)