

初学者を対象とした段階的 UML 設計手法の提案

影山 智一[†], 上田 賀一[†]

[†] 茨城大学

オブジェクト指向設計において、初心者が詳細で品質の良い設計を行うことは難しい。本研究では、段階的に設計することで初心者が設計を学びつつ、設計の品質を高めて行く手法を提案する。あらかじめ指導者が学習者にユースケースといくつかに分かれた設計工程を指定する。そして、学習者はそれらを元に設計し、設計の欠陥検出、各設計工程で書く UML 図要素判定および設計の品質評価を行う。測定結果を元に学習者は設計を修正する作業を繰り返して設計の品質を高めて行く手法を考案した。

A Proposal of Phased UML Design Approach for the Beginners

Tomokazu KAGEYAMA[†] and Yoshikazu UEDA[†]

[†]Ibaraki University

In the object-oriented design, it is difficult for the beginners to perform a detailed design with good quality. This study proposes the UML design approach in which the beginners learn the design technique and improve the quality of design by gradually designing. In this approach, the teacher gives the learner the usecase and specifies the design steps divided into some to the learner. The learner designs UML diagrams based on them. And, the learner detects the defects of the design, judges the elements of UML diagram written in each design process, and evaluates the quality of UML design. We designed the approach to which the learner improved the quality of UML design by repeating work to correct the design based on the result of the judgement and the evaluation.

1 はじめに

近年、ソフトウェアの複雑化・大規模化が進んでいる。大規模なソフトウェアの開発にはコストがかかってしまうため、ソフトウェアの再利用・部品化を意識した開発が行われている。オブジェクト指向の概念はソフトウェアの再利用・部品化に有効であり、オブジェクト指向による開発や UML での設計が増えてきている。そのような中、オブジェクト指向による開発や UML での設計ができる技術者を育てるために大学等の情報系教育機関ではカリキュラムに取り入れ、IT 系企業では新入社員教育で指導したりしている。しかし、オブジェクト指向ソフトウェア設計は容易に習得できるものではない。品質の良い設計を行うためには学習の上に設計の経験を積む必要がある。そのため初学者が品質の良い設計

を行うことは難しい。

上記の背景のもと、本研究では、初学者を対象とした段階的な UML 設計手法を提案する。複数の設計メトリクスを用い、段階的に設計することで設計の仕方を学び、かつ、品質の良い UML 設計になるよう支援する手法である。ソフトウェアの特性を設計の観点から定量的に評価するため、設計メトリクスを用いることでソフトウェア設計の状態や問題点を客観的に把握することが可能となる。

本報告では、提案手法である段階的 UML 設計手法について述べ、本手法の評価のための例題と実験の設定について説明したのち、本手法の有効性の検討を行う。そして、本研究に関連する他研究について述べ、最後に本研究のまとめと今後の課題について述べる。

2 提案手法

ここでは、本研究の提案手法について、その手順を示し、支援する要点を説明する。

2.1 概要

段階を踏んで設計することで、詳細で品質の良い UML 設計を目指す。まず指導者がいくつかに分かれた設計工程の内、初学者に学んでもらいたい工程を指定する。そして学習者は指導者の指定した工程を段階的にこなしていくことで UML 設計を学んでもらう。それぞれの工程で UML 設計の欠陥を検出と UML 図要素判定を行い、その結果から設計を修正するという作業を繰り返す。欠陥検出と同時に設計の品質を測定し、品質測定の結果から品質の良い UML 設計になっていくことを確認する。本研究では UML 図の中でも基本的なクラス図とシーケンス図に着目する。指導者が設計初学者にユースケースを与え、設計段階を指定することで UML 設計を学習する。

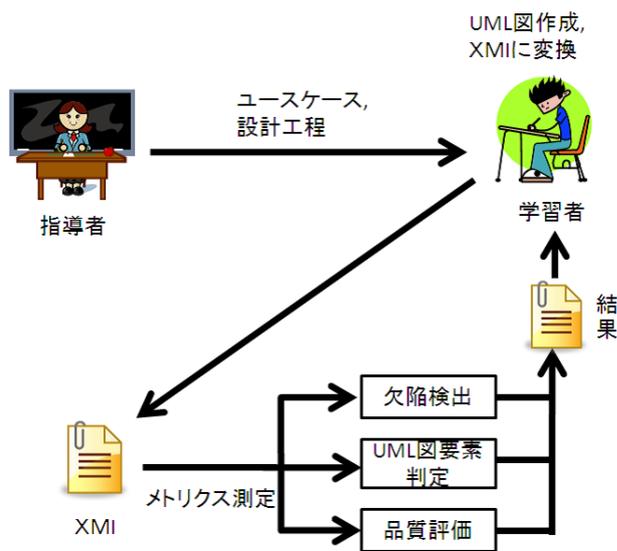


図 1: 手法の手順

本手法の流れを図 1 に示すとともに、その手順を以下に記す。

1. 指導者は学習者にユースケースを与え、ユースケースに合わせてメトリクスの閾値と設計段階を決める。

2. 学習者はユースケースと設計工程を参考に UML 描画ツールでクラス図とシーケンス図を作成する。
3. UML ドローツールより出力された XMI からメトリクスを測定する。
4. メトリクス値から現設計工程における必要な図要素の判定と品質測定、欠陥検出を行う。
5. 学習者に結果を通知する。
6. 学習者は結果より設計を修正する。
7. 現設計工程を終えたら次の工程へ進む。
8. 最終設計工程を終えるまで 2~7 を繰り返す。

2.2 段階的な UML 設計支援

本手法はいくつかに分かれた設計工程を順にこなすことで UML 設計を学んでもらうものである。さらにそれぞれの設計工程では設計図からメトリクスを測定し、設計の欠陥検出、UML の図要素の存在判定、品質評価を行う。

2.2.1 設計工程

本手法では学習者に学んでもらう上で基本的で重要である UML 設計要素を 7 つの設計工程 (以降では単に工程と称す) に分けた。それぞれの工程には書かなければならない UML 図の要素が存在し、必要な図要素が書かれ、設計に欠陥が存在しない、または少ない状態ならばその工程は終了とする。欠陥の多寡は指導者がユースケースに合わせて設定する。以下に 7 つの工程の詳細を示す。

- 概要クラス工程
 - クラス図: クラス名, 属性名, メソッド名を書く
 - シーケンス図: オブジェクト, メッセージ名を書く
- 詳細クラス工程
 - クラス図: 属性の型, メソッドの戻り値の型, メソッドの引数を書く
 - シーケンス図: メッセージの引数, リターンを書く
- カプセル化工程
 - クラス図: 属性/メソッドの可視性を書く
 - シーケンス図: メッセージの送り先と可視性の対応を確認する。

表 1: 欠陥リスト (一部)

| # | 欠陥名 | 概要 | 対応するメトリクス | *の閾値 |
|-----|--------------|-----------------------------------------------------|----------------|-------------------------|
| D1 | 大規模なクラス | クラスのインスタンス変数が多すぎる, もしくはメソッドが多すぎる | NOA NOM | |
| D2 | 大規模なメソッド | クラスのメッセージ送信数が多い, もしくはパラメータが多すぎる | NMSM NOP | |
| D3 | 小規模なクラス | クラスの属性が少ない/無い, もしくはメソッドが少ない/無い | NOA* NOM* | 1 : Bad 0 : Very Bad |
| D4 | 公開インタフェースの過多 | クラスの public インタフェースが多すぎる | NPubM | |
| D5 | データの貧弱なカプセル化 | クラスが public 属性を持つ | NPA | |
| D6 | 貧弱なインタフェース | クラスが public メソッドを持たない | NPubM* | 0 : Bad |
| D7 | 隠すべきメソッド | 自クラスでしか用いないメソッドを公開している | NACUM | |
| D8 | 非公開メソッドの過多 | クラスの private/protected メソッドが多すぎる | NPM | |
| D9 | 不十分な抽象化 | クラス抽象クラスが非常に少ない | RAC | |
| D10 | 複雑な抽象化 | 祖先クラスが多い | NOAC | |
| D11 | 余分なサブクラス | スーパークラスが1つのサブクラスしか持たない | NOS | |
| D12 | 関連しないデータと振舞い | クラスの属性とメソッド間に関連がない | LCOM | |
| D13 | 多面的な抽象化 | スーパークラスがいくつかの抽象測面を持つ | LCOM, NOA, NOM | |
| D14 | データクラス | クラスがデータだけをもつ | NOM* | 0 : Bad |
| D15 | 引数のないメソッド | クラスの大抵のメソッドがパラメータを持たず, 処理を行うために public/クラス変数を用いる | RMNP, NSV | |

● 関連工程

- クラス図: 関連線 (誘導可能性を含む), 関連名, 多重度, ロール名を書く
- シーケンス図: 関連の向きとメッセージの向きを確認する

● 集約/コンポジション工程

- クラス図: 集約, コンポジションを書く
- シーケンス図: メッセージ送信元のオブジェクトがメッセージ送信先のオブジェクトを属性値として持っているか確認する

● 継承工程

- クラス図: 汎化を書く
- シーケンス図: なし

● 実現工程

- クラス図: 抽象クラスの作成・継承とインタフェースの定義とその実装との間の関係を書く
- シーケンス図: なし

2.2.2 欠陥検出

各工程でメトリクスを計測し, その値から欠陥検出を行う. 欠陥検出モデルは佐藤らのモデル [2] を一部改良する. 佐藤らの欠陥検出モデルは完成された詳細な設計に対して適用するものであるが, 本研究では作成途中の設計図に対して用いる. そのため, まだ書かれていない UML 図要素に関する欠陥は検

出されないようにする. 例えば, 概要クラス工程, 詳細クラス工程, カプセル化工程の順で設計すると仮定する. 概要クラス工程の段階では詳細クラス工程とカプセル化工程で書く UML 図要素に関する欠陥は検出されない. 詳細クラス工程の段階に移ると, カプセル化工程で書く UML 図要素に関する欠陥は検出されないようになる. 欠陥検出の結果から学習者に設計の問題点を指摘し, 学習者は設計を修正する. 欠陥の重大さは「little bad」「bad」「very bad」の3段階存在する. 表1に検出される欠陥の一部を示す.

2.2.3 UML 図要素判定

各工程でクラス図とシーケンス図それぞれに書く必要がある図要素が存在する. 図要素が存在するかどうかはメトリクスによって判定する. この図要素の存在の判定に用いたメトリクスは川崎らの報告 [3] を参考にした.

2.2.4 品質評価

欠陥検出の際に同時に設計の品質を評価する. 品質評価は表2の観点から100点満点で示す. 品質評価モデルも佐藤らのモデル [2] を使用するが, 欠陥検出モデルと同様に改良する. 具体的には, まだ要求されていないために書かれていない UML 図要素

に関するメトリクスの得点は評価できないことから50点とする。同様に、書かなければならないのに、書かれていないUML図要素に関するメトリクスの得点は0点とする。品質評価の結果自体は設計を修正するための情報として用いないが、本手法の有効性を確認するために品質を評価する。

表 2: 品質特性

| 品質特性 | 定義 |
|------|--------------------------------|
| 再利用性 | 設計の一部が新しい設計に再適用しやすいか |
| 拡張性 | 新しい機能を追加しやすい設計であるか |
| 有効性 | オブジェクト指向概念と技術を設計に適切にとりいれられているか |
| 柔軟性 | 変更しやすい設計であるか |
| 機能性 | クラスに割り当てられた責務を行っているか |
| 理解性 | 設計を見て簡単に構造を理解することができるか |

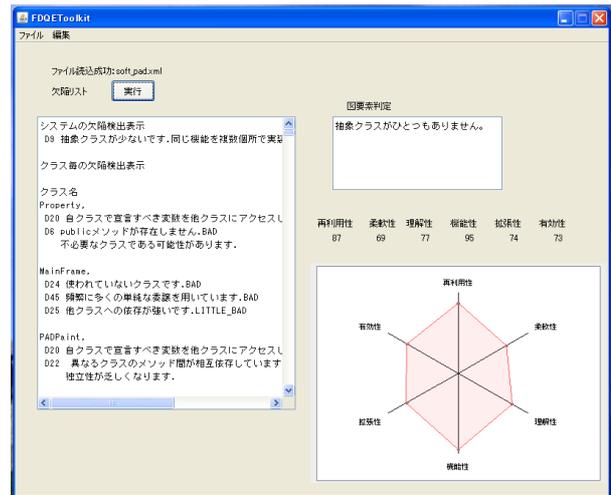


図 2: 支援ツールの画面

2.2.5 支援ツール

本手法を支援するため、佐藤らの欠陥検出ツール[2]を拡張した。本ツールはUML描画ツールで作成した設計図のXMIファイルからメトリクスを測定し、メトリクス値より欠陥検出、UML図要素判定、品質評価を行う。以下に本ツールの実行手順を、図2にツールの画面を示す。

- (1) UMLモデリングツールから出力されたモデル情報であるXMIファイルを本ツールの入力情報として読み込む。
- (2) 設計図の欠陥を検出し、UML図要素を判定し、品質評価を行う。
- (3) 欠陥検出、UML図要素の判定、品質評価の結果を画面に出力する。

以下にツールの機能について説明する。

欠陥検出 欠陥検出の結果はクラス毎、メソッド毎など欠陥の検出単位で表示される。図3に欠陥検出の表示部分を示す。

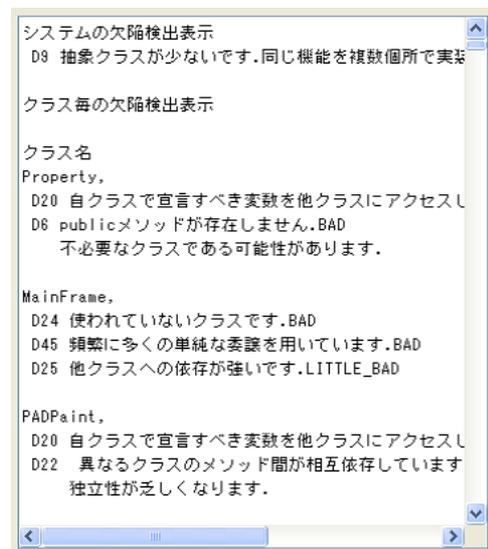


図 3: 欠陥検出の結果

UML図要素判定 UML図の要素判定結果は各工程に必要な図要素が書かれていなければ指摘する。図4にUML図要素判定の結果表示部分を示す。

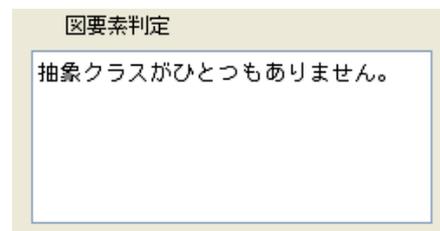


図 4: UML図要素判定結果

品質評価 品質評価の結果は再利用性，拡張性，有効性，柔軟性，機能性，理解性の6つの品質特性を100点満点で数値とグラフで表示する．図5に品質評価の結果表示部分を示す．

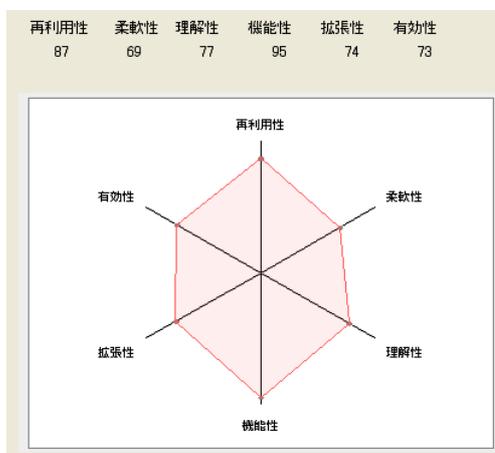


図 5: 品質評価の結果

3 評価実験

2章では提案手法について説明を行った．本章では，適用例として書籍「初めて学ぶUML」[4]8章のケーススタディを用いた．2.2.5節で説明したツールの入力には，UMLモデリングツールである Rational XDE によって XMI 形式に変換されたファイルを使用した．

3.1 適用例：コンビニエンスストアの販売システム

例題はコンビニエンスストア（以下コンビニ）の販売システムの設計であり，図6にユースケース図を示す．コンビニにはスタッフ，チーフ，店長があり，それぞれ以下の業務を行う．

- スタッフ：販売処理のみを行う
- チーフ：販売処理および商品在庫管理を行う
- 店長：販売処理，商品在庫管理，売上管理，商品管理を行う

実験は3人の情報系学生を対象として実施した．3人はオブジェクト指向の概念と Java プログラミ

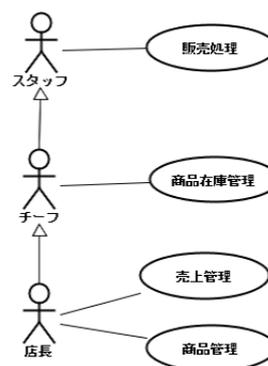


図 6: コンビニの販売システムユースケース図

ングについての講義を受けたことがあり，UML についてはどのような図があるか知っている程度の知識がある．

3.2 評価実験

3.2.1 選択した工程

本手法の有効性を確認するため 2.2.1 節において定義した7つの工程すべてを概要クラス工程から紹介した順にこなすよう指定した．欠陥検出の結果に「bad」「very bad」が出た場合には設計を修正するように指示した．

3.2.2 実験結果

収集したデータは各工程が終了するまでの時間，工程ごとのメトリクス測定回数，各工程終了時の品質特性の得点である．ここでは1人の学生の実験結果のみを示す．各工程が終了するまでの時間，工程ごとのメトリクス測定回数，1測定当りの所要時間，各工程終了時の品質特性の得点を図7～10に示す．

設計にかかった時間 図7より設計にかかった時間は概要クラス工程で70%以上占めていることがわかる．これはユースケースからクラスを抽出するために試行錯誤して，手間取っていたためであると考えられる．

計測回数 図8より工程が進んでいくと測定回数が少なくなった．概要クラス工程の段階では必要と考えるクラスを全て抽出するものと考え，それらの調整に時間をかけ，確認を繰り返したためである．こ

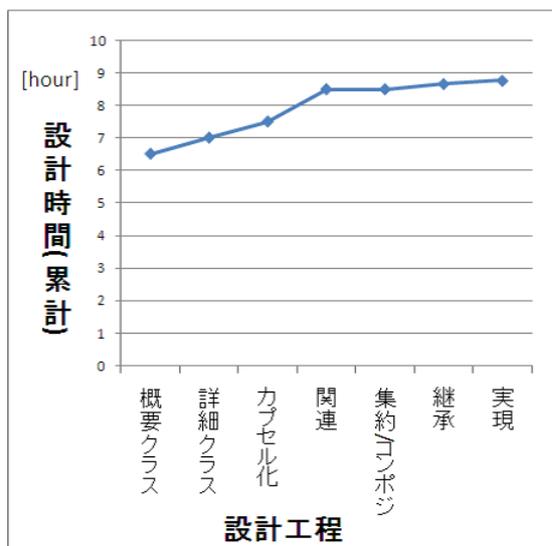


図 7: 各工程終了までの時間

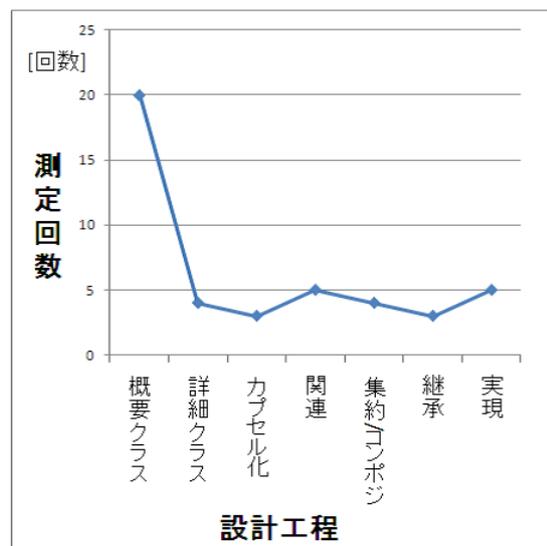


図 8: 各工程における測定回数

の時点で実際には詳細に設計していたので、後の工程で測定する回数が少なく済んだと考えられる。設計にかかった時間と合わせて見てみると概要クラス工程で時間をかけ、詳細に設計していたことが読み取れる。また、図 9 を見ると、概要クラス工程ではクラスの抽出に時間がかかっているため、1 測定当りの時間が長くなっている。詳細クラス、カプセル化、関連の 3 工程では、抽出したクラスそれぞれに対して、図要素を付け加えているので 1 測定当りの時間が長くなっていると考えられる。集約/コンポジション、継承、実現の 3 工程は始めの 4 工程で詳細に設計されており、大きく設計の変更がなかったため、1 回の計測に要する時間が短くなっていると思われる。

設計品質 図 10 を見ると設計の品質は総合的に高まっていることがわかる。しかし、再利用性、機能性、理解性の 3 つの品質特性は実現工程で得点が下がっている。これは実現工程を終了させるために無理に抽象クラスを用いたためであると考えられる。また、柔軟性、拡張性、有効性の 3 つの品質特性は継承工程で大きく得点が上がっている。これらの品質特性は設計特性の継承と関連が強く、継承の得点を求めるメトリクス数は他の設計特性よりも多い。そのためクラス図中に汎化関係を書くことで大きく得点が上がったと考えられる。

4 関連研究

ここでは、UML 設計の教育・学習するという観点と UML 設計の評価という観点で、関連研究について述べる。

4.1 オブジェクト指向、UML に関する教育の視点と実践

オブジェクト指向、UML に関する教育に関する研究報告 [6] がある。この報告ではオブジェクト指向開発理解のためのパターン指向の利用やグループ作業による UML の理解について述べられている。本手法をグループで行い、グループで議論することで、他人との設計の差異を理解し、UML の理解を深めることができると考えられる。

4.2 創造性教育における学習成果の評価に関する提案

創造性教育における学習成果の評価への利用を目的としたモデリングテンプレートの提案 [7] がある。これは UML でモデリングを行う前の中間言語としてモデリングテンプレートを用い、段階を踏んで UML 学習を支援するものである。しかし、この報告ではモデルの検証は行われていないため、有効性が確認されていない。本手法も同じく UML 学習を支援するものだが、こちらでは UML による設計そ

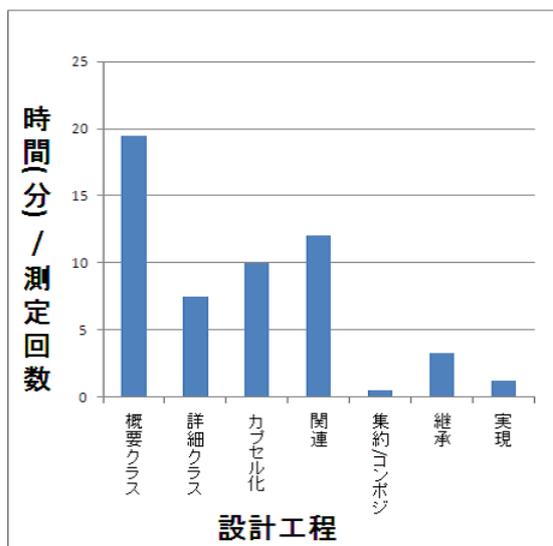


図 9: 1 測定当りの所要時間

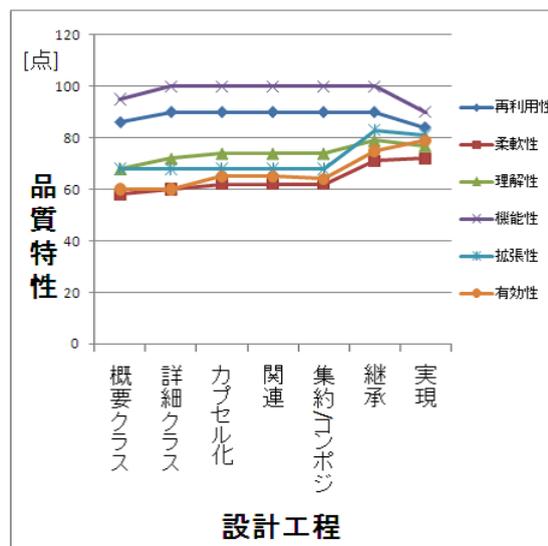


図 10: 各工程終了時の品質特性の得点

のものを段階的に分けて行うことで UML 学習を支援している。

4.3 UML 設計の欠陥検出および品質評価モデル

佐藤らが複数のソフトウェアメトリクスを使用して UML 設計図からソフトウェア設計の欠陥検出と品質評価の手法 [1][2] を提案している。欠陥検出手法は UML 設計図よりメトリクスを測定し、測定結果からその設計に含まれる欠陥を検出するモデルである。品質評価手法は UML 設計図よりメトリクスを測定し、メトリクス値から設計特性を算出し、設計特性から品質特性を算出するモデルにより実現している。

これらの欠陥検出モデルと品質評価モデルは完成された詳細な設計に対して適用するものであるが、本研究では作成途中の設計図に対して用いる。そのため本研究の目的に適するようにこれらのモデル改良した、欠陥検出の結果を設計修正のための情報として用い、品質評価モデルを使用して設計の品質が向上していくことを確認した。

4.4 UML 設計の詳細度測定

川崎らが複数のメトリクスを用いて、オブジェクト指向設計演習において、設計の詳細レベルを段階的に達成していくことを支援する手法 [3] を提案している。

本研究では川崎らの手法で用いられている UML 図要素の判定のためのメトリクスを使用している。また川崎らの手法では設計の詳細さと設計するステップの数を指導者が細かく指定しなければならない。本研究では設計する工程を定義することで、初学者の設計品質を高めつつ、指導者の手間を減らした。

5 終わりに

5.1 まとめ

本研究では、段階的な UML 設計支援法を提案した。UML 設計初学者が詳細で品質の良い設計を行うことは難しい。この手法を適用することで品質の良い設計を行うことが可能になる。本手法の有効性の確認のため、専門書籍の例題を元に評価実験を行った。実験の結果、段階を踏んで行くにつれ設計品質が高まっていくことが確認できた。

5.2 今後の課題

実例への適用数 本研究では 3 人の学生を対象に本手法を用いた UML 設計を実施してもらう実験を行った。3 人ではサンプル数としては不十分である。多くのサンプルで本手法を適用し統計学的解析を行い有効性を示す必要がある。また 3 人の初学者がモデリングツールの使用に慣れていないなどの理由で、設計時間がかかってしまっていた。同じ人物に同程

度の難度の別のユースケースで再び手法を適用することによって有効性を確認する必要がある。

工程の見直し 評価実験の結果、3人の学生からこの工程の分け方では設計しにくいと意見が出た。例えば、クラス図は一般的には関連線が引かれる。わざわざ関連線を引くために関連工程として分ける必要があったのかといった意見があった。書籍「ダイアグラム別 UML 徹底活用」[5]を参考に UML 図要素を選定したが、図要素の分け方が不適切と考えられる部分があった。よって工程の分け方を見直す必要がある。さらに見直した工程で評価実験を行う必要がある。

支援ツールの改良 本研究では佐藤らの欠陥検出ツール [2] に UML 図要素判定と品質評価の機能を追加し、結果を画面に表示させるよう改良した。元となった欠陥検出ツールは UML1.4 のみに対応しており、UML2.x は扱えない。UML1.4 と UML2.x では図の書き方に違いがあるので、現状では本研究で拡張した最新の UML を扱えない。よって UML バージョンに合わせてツールを改良する必要がある。また、実験でツールが使いにくかったという感想も出ているので、ツールのユーザビリティの向上を図る必要がある。

参考文献

- [1] 佐藤美穂, 田村真吾, 上田賀一: UML 設計を対象とした品質評価モデルの検討, 特集: ソフトウェア工学の効果と価値 分析・設計技術, 情報処理学会論文誌, Vol.49, No.7, pp.2319-2327, (2008).
- [2] 佐藤美穂, チョインゾン ムンフナサン, 上田賀一: オブジェクト指向設計の構造的欠陥検出モデルの検討, 情報処理学会研究報告, No.2009-SE-163, pp.185-192 20090311, (2009).
- [3] 川崎結花, 上田賀一: 演習教育を対象とした UML 設計の詳細度判定手法の提案, 情報処理学会研究報告, No.2009-SE-163, pp.145-152, (2009).
- [4] 竹政昭利 著: 初めて学ぶ UML, 株式会社ナツメ社, 2003.
- [5] 井上樹 著: ダイアグラム別 UML 徹底活用, 株式会社翔泳社, 2005.
- [6] 中尾 信明: オブジェクト指向, UML に関する教育の視点と実践, 情報処理学会研究報告, No.2004-CE-074, pp.9-16, (2004).
- [7] 長尾 祐樹, 鈴木 裕利, 藤吉 弘亘, 藤井 隆司, 石井 成郎: 創造性教育における学習成果の評価に関する提案, 報処理学会研究報告, No.2008-SE-160, pp.131-138, (2008).