

## GRASP を利用するオブジェクト指向設計学習支援システム Object oriented design learning support system using GRASP

杉浦 啓孝†  
Hirotaka Sugiura

酒井 三四郎‡  
Sanshiro Sakai

### 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なわれており、再利用しやすく、保守のしやすいソフトウェアを作ることができるようになった。開発には UML がソフトウェア設計で利用されている。しかし、UML を利用するだけで、オブジェクト指向に基づく設計を行なえるということではない。保守性、再利用性を高くするためには、経験と知識が必要であり、初心者が開発を行なうとき、これらが十分でない。そのため、初心者がオブジェクト指向に基づく設計を行なうには学習や支援が必要となる。

そこで、本研究は初心者を対象に、オブジェクト指向設計を学習することを目的とするシステムの設計を行なう。本システムでは、オブジェクト指向設計を学ぶために、オブジェクト指向設計の基本方針となる GRASP を理解してもらう。また、システムは、UML クラス図、コラボレーション図を描き、設計を行い、その過程で、GRASP を適用させる。

### 2. GRASP

GRASP とは、[1]で示されている 9 つのパターンである。オブジェクト指向設計では、オブジェクトに役割を持たせ、複数のオブジェクトの協調関係を構築することで 1 つのソフトウェアを作り上げる。その役割を責務といい、クラスに操作を割り当てることで、責務を実現する。オブジェクトに責務を割り当てる際の基本原則、方針をまとめたものが、GRASP である。

このパターンが示していることは、中級者以上の人が設計の際に無意識に使うような基礎的なことである。しかし、初心者には基礎が根付いていないので、設計するときに、どのように設計を行なえばよいのか解決できない。したがって、GRASP は初心者が最初に学ぶべき考えであり、この学習を支援する。表 1 に今回実装したパターンについて示す。

### 3. 関連研究と本研究の位置付け

オブジェクト指向学習を支援する研究[2]が行なわれている。この研究は初心者向け統合開発環境の設計とそれを用いた教育の方法を提案している。クラス図を用いてクラス同士の関連やオブジェクトの視覚化など初心者が理解しやすいように工夫されている。

また、デザインパターンを利用することでソフトウェアの品質を高めることができるが、この利用はパターンを熟知した設計者でなければ利用することが難しい。そのため、デザインパターンを適用させる研究[3]もある。[2]の研究では、オブジェクト指向を学ぶことが目的であり、設計に関する支援は行なわれていない。[3]の研究で

† 静岡大学大学院情報学研究科

‡ 静岡大学情報学部

表 1:実装したパターンとその内容

パターン名	問題と解決策
情報エクスパート	責務を配置するとき、その責務を実現するために必要な情報を持っているクラスに責務を配置する
疎結合性	オブジェクトの独立性、再利用性を高めるには、結合度が低くなるように責務を配置する
高凝集性	複雑さを低減するには、凝集度が高くなるように責務を配置する
クリエータ	オブジェクトを生成する責務は、そのオブジェクトを集約、包含するオブジェクト、またはそのオブジェクトを生成するために必要な情報を持っているオブジェクトに割り当てる
コントローラ	システムイベントを処理する責務は「コントローラ」となるクラスに割り当てる

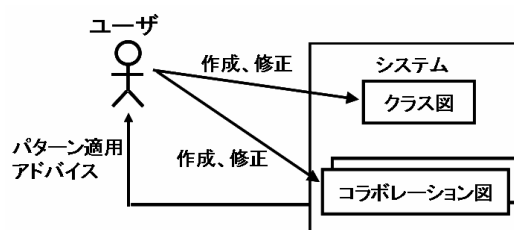


図 1:システムの概観

デザインパターンを適用させ、パターンの理解を目的にしている。しかし、初心者が学ぶべきことは基本的な設計の方法とその理解であると思う。

本研究では、GRASP を理解させ、オブジェクト指向設計に置ける基礎的な部分を学ばせることが目的である。

### 4. システムの概要

システムは統合開発環境 Eclipse のプラグインとして実装され、Eclipse 上で動作する。

#### 4.1. システムの構成

システムの構成を図 1 に示す。システムはクラス図とコラボレーション図を扱う。システムは 1 つの設計では、クラス図を 1 つ、コラボレーション図を複数描くことができる。ユーザは 2 種類の図を作成して、システムから GRASP を適用するアドバイスを受けることができる。

#### 4.2. 支援対象

支援対象はオブジェクト指向について学習しているが、実際に UML を用いた設計を行ったことがない初心者レベルである。

#### 4.3. 支援方法

ユーザはクラス図を用いて全体的な設計を行い、コラボレーション図で責務の配置を検討する。責務の配置を検討するにはシーケンス図を用いることもあるが、シー

ケンス図が時間経過に注目するのに対して、コラボレーション図はオブジェクト間の作用に注目する。オブジェクト指向の考えを身に付けるにはコラボレーション図の方が適していると思うのでこちらを利用する。作成された2種類の図からアドバイスを作成して、ユーザがGRASPを適用する支援を行い、オブジェクト指向設計を学ぶ。

アドバイスの提示方法を情報エキスパートパターンの場合に沿って説明する。1)操作名を入力する。2)描かれたクラス図から、クラス、属性の情報を取得し、ユーザは操作を実現するために必要な情報を選択する。3)選択された情報をどのクラスが多く持っているのか計算し、多い順にクラスを表示する。その順番がその操作を持つべき情報エキスパートなクラスとなる順番となる。また、パターンを適用したときの効果を説明する。これらのアドバイスからユーザはクラス図の修正を行なうことで、パターンを適用できるようになる。

パターンのアドバイスはユーザの描いたクラス図、コラボレーション図から作られる動的な部分と、パターンの説明を示した静的な部分から構成されている。上記の例では、情報エキスパートな順番を表示する部分が動的な部分となる。

#### 4.4. ユーザインタフェース

図2にシステムのユーザインタフェースを示す。ユーザはクラス図、コラボレーション図を描くエディタ、パレット、GRASP適用ツールバーを主に使用する。エディタでは、パレットで選んだ部品を作図し、クラス図、コラボレーション図を描く。学習者はGRASP適用ツールバーのボタンからアドバイスをもらうことができる。

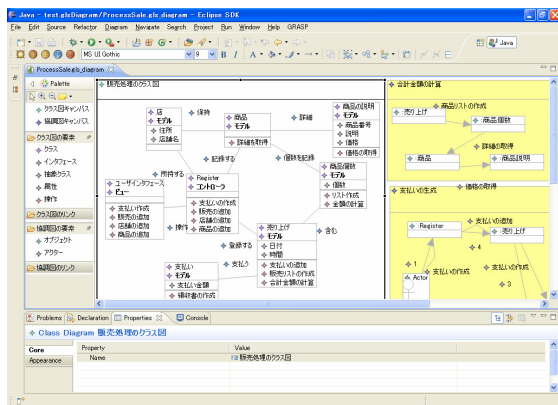


図 2: システムのユーザインタフェース

#### 5. 評価実験

オブジェクト指向初心者である大学生で評価実験を行なった。実験は2回に分けて実施した。1回目は情報エキスパート、疎結合性、高凝集性の3パターンの支援機能が完成した時に行い、4名で行なった。2回目はクリエータ、コントローラパターンの支援機能を追加したときに1回目とは異なる4名で行なった。

前者はその時点で完成した3パターンを、後者は情報エキスパートと追加した2パターンの計3パターンの評価を行なってもらった。実験方法は、クラス図に各パターンに適していない点を作り、システムからアドバイスを受け、クラス図の修正を行なうという方法である。実験後、

アンケート調査を行なった。アンケート内容として、各パターンの理解度とパターンを今後利用できるかを評価してもらった。4は十分理解できた/十分利用できる、3は少し理解できた/利用できる、2はあまり理解できなかった/あまり利用できない、1は理解できなかった/利用できないである。その平均評価を表2に示す。

表 2: 評価結果

パターン名	平均評価	被験者数
情報エキスパート	2.8	8
疎結合性	2.6	4
高凝集性	2.6	4
クリエータ	2.3	4
コントローラ	2.5	4
今後利用できるか	2.8	8

また、自由記述から否定的意見として、「アドバイスの意味は分かるが、どのように改善すればいいのかわからない」、「用意されているクラス図が分かりにくかった」とあり、肯定的意見として「情報エキスパート、クリエータパターンのアドバイスは適するものを見せてくれてよかった」という意見があった。

#### 6. 終わりに

実験ではシステムを利用して、各パターンの理解度の評価を行なってもらった。しかし、いくつかのアドバイスはユーザにとってどのように活用し、どのような修正を加えるのかという説明が不足していた。このために、パターンの理解度が低くなってしまった。

今後の課題として、ユーザの理解度を高めるためにアドバイスの提示方法を改善することである。また、パターンや語彙の説明を行なえる機能を提供できると理解が増すだろう。そして、現在のシステムではボタンを押すことでアドバイスを受ける仕組みになっているので、図を描きながらアドバイスを提示できると便利になる。また、2回の実験では、オブジェクト指向設計初心者の方に評価実験を行なってもらい、パターンを理解することができるかどうかの評価を目的に行った。この評価とは別に、オブジェクト指向上級者の方にパターン適用アドバイスの評価をしてもらいたい。アドバイスの良い点、悪い点を指摘してもらうことで、初心者にも理解できるアドバイス作りが行なえることと思う。

#### 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] Michael Kölling, Bruce Quig, Andrew Patterson, John Rosenberg: “The BlueJ System and its pedagogy”, Computer Science Education, vol. 13, issue 4, pp. 249-268 (2003)
- [3] 山下 純司, 林 雄二: “プログラミング段階のデザインパターン適用支援ツール”, 北海道大学紀要, 第13巻, 第2号, pp.67~74 (2002)