

オブジェクト指向開発における事例ベース推論の適用性に関する一考察

On the Case Based Reasoning about Object Oriented Development

吉村 晋^{*1}
Susumu Yoshimura

^{*1} サレジオ工業高等専門学校 情報工学科
Salesian Polytechnic

In the development of Object-Oriented Software, one of the effective methods for advancing the productivity of Software is Software Reuse. To support the software development, Framework Method and Design Pattern Method are used effectively. These Methods match to the CBR techniques of searching software reuse testbed. I discuss about this discussion and application to the Education of Software Development towards undergraduate students.

1. はじめに

オブジェクト指向言語によるソフトウェア開発には多くのソフトウェア部品が再利用される。また設計段階においても設計部品に相当する仕組みが用意され、過去の部品をもとに再利用する多くの仕組みがある^[佐藤 1998]。その一つが継承を利用した再利用の仕組みで、アプリケーションフレームワークと呼ばれる。もう一つは複数部品構成と利用法に関する仕組みで、デザインパターンと呼ばれる。再利用の仕組みに関しては、過去にも多くの試みがなされてきた。著者らは通信ソフトウェア開発の場でソフトウェア仕様の再利用を事例ベース推論で行う試みを行った。特に通信プロトコルの対応の類似性に着目し、仕様部品再利用の試みを行った^[吉村 1993]、^[Chang-Fa 1995]、^[唐橋 1995]。これに対しフローダイアグラム(パターン)の類似性のみではプログラムの振る舞いの等価性(Behavior Equivalence)が保証されないとの批判がなされた。ところがオブジェクト指向ソフトウェア開発の手法は、まさに凡例を用いて再利用による開発を支援する手法を取る。ソフトウェア教育の場でも、再利用の考え方を自然に受け入れる^[吉村 2009]。そこで事例ベース推論の適用が比較的容易に受け入れられる^[Calero 2004]。

本報告は、その適用性について考察するとともに高専(大学)のソフトウェア教育の場での事例ベース推論^[Watson 1997]の適用性について考察する。

2. フレームワーク

簡単なフレームワークは、クラス継承によりなされる。たとえば抽象クラスがフレームとして存在する。複雑なフレームワークの例は、Struts, Struts2, JSF, Searser2, Spring, Hibernate などのように多くのクラス群からなるフレームワークも存在する。抽象クラスのフレームワークの場合、フレームワーク側が、サブクラスにホットスポットを持ち、実際に実装するサブクラスで(フックメソッド)を作れば良い。

3. デザインパターン

デザインパターン「インターフェースを中心とした仕組み」は、クラスライブラリより一般的なパターンの類似性をパターンとしており、当然再利用を前提として考案されている^[結城 2004]。デザインパターンに関しては、

①生成に関するデザインパターン(5パターン): Abstractfactory,

Builder, FactoryMethod, Prototype, Singleton.

②構造に関するパターン(7パターン): Adaptor, Bridge,

Composite, Decorator, Façade, Flyweight, Proxy

振る舞いに関するパターン(11): Chain of Responsibility,

Command, Interpreter, Iterator, Mediator, Memento, Observer, State, Strategy, Template Method, Visitor

これらのパターンは、構成要素、目的、協調関係、別名、結果、動機、実装、適用可能性、サンプルコード、構造、使用例、関連するパターンの項目で、カタログ的にまとめられ目的のパターンが検索できる。この項目は、そのまま事例ベースの辞書項目と考えることができる。たとえば振る舞いに関する凡例パターン Iterator は 集合体の一つ一つ数え上げて振る舞いをパターン化する。

4. 類似性と適用性

事例ベース推論は、類似性をよりどころに適用するメカニズムで、ソフトウェア開発では上流工程に適用しやすいと考えられる。過去の試みも仕様レベルでの適用を試みている^[Chang-Fa 1995]。

しかしフレームワークやデザインパターンは、構造や振る舞いの類似性に対応するため抽象度が高い。この仕組みの類似性は便利であるが、サンプルパターンへの適用は一定の経験が必要で部品再利用を抽象的に捉える凡例となる。

これに対し具体的プログラム段階のクラス(サンプルコード)は目的に依存した類似性を持ち、それだけ具体性が高く、連想や類推が容易である。学生(低学年から)のソフトウェア教育・研究の場では、あまりに抽象度の高いメカニズムは効果的でない。

5. 教育環境での適用性

オブジェクト指向ソフトウェアの学習過程(例えば Java プログラミング環境)が、事例学習過程と非常にマッチする。

学生にいきなり設計を教えるのは、動機付けの点で問題が多い。しかも比較的容易に開発を体験させることが重要になる。教育現場では、ソフトウェア教育が言語や文法の学習の場や、一方的な設計のみの場になる傾向が強く、学生の達成感を満たさない。この要求を満たすために良好な手段が事例から学ばせることである。そこで最初の段階では、事例からの改良を積極的に試みている。事例(サンプル)からの変更で、求めるものに近いプロダクトを作らせ、評価させることで学生の技術スキル向上を目指す。このような試みの中で簡単なCBR利用のテストベッドを検討中である。

学生に合わせた事例ベース推論。

・見かけ上の類似性をよりどころソフトウェア構築を行う

連絡先: 吉村 晋, サレジオ工業高等専門学校, 東京都町田市小山ヶ丘, Tel.042-775-3020, yosimura@salesio-sp.ac.jp

・具体的事例に適した類似性

対象: 高等専門学校 3年から5年(18歳から20歳)

事例: 抽象度を低く、

抽象度高い事例は、ほとんど受け入れられない。ソフトウェア開発経験が少ないため、具体的なプロダクト前提のアプローチが有効である。

事例から容易に受け入れられる範疇

1. ウィンドウ周りの部品 (GUI) によるサンプル事例 (ボタン, メニュー, スレッドのサンプルパターン, 画面がちらつかないように表示する事例)。
2. 学生の好むサンプル部品事例, たとえばシューティング, ゲーム事例, モグラたたきゲーム事例
3. 時計, カレンダー, タイマー, 電卓など簡単なガジェット上の事例などを検討中である。
4. 辞典的なサンプル事例^[山下 2003]

これから機能的な類似性を取り出し, CBR手法を取り入れる。

まだ検討途中であるが, 初学者にとってかなりのレベルで索引を付きの事例パターンと類似辞書を用意したCBRで良好なサポートデスクが可能になると考えられる。

ただ見かけ上の類似性が多く, 多相性(ポリモーフィズム)による齟齬が生じる。事例として

1. Applet 部品を使って画像を表示するサンプルコード事例パターンと Frame や JFrame 部品上で画像を表示するパターンは大きく異なる。
2. AU 部品で音を出力するサンプルコード事例パターンと Midi で同様なことを行わせるバタ菅原一は異なる。

Applet 部品と Swing や AWT 部品の違いを想定していない。クラスのメソッド利用条件や画像データの制限でアクセスメカニズムが部品レベルで違うにも関わらず, 類似レベルと判断する。これが例外条件となる。

6. 簡易パイロットモデルの検討

現在パイロットモデルの検討中である。

・フレームワークは知識フレームが対応する。

ホットスポット: 知識フレームのスロットが該当する。これとマッチするフックメソッドは, デモンメカニズムで対応する。そこで類似度で事例候補を選択し, 修正利用する。

・デザインパターンは, 一種のスクリプトメカニズムで対応する。

[システム構成]

本来ならばソフトウェアワークベンチ(SWB)に組み込む形で構築するのが良いのであるが, 現時点では困難と考え, サポートデスクとしての構築を検討している。図1にシステム構成図を示す。

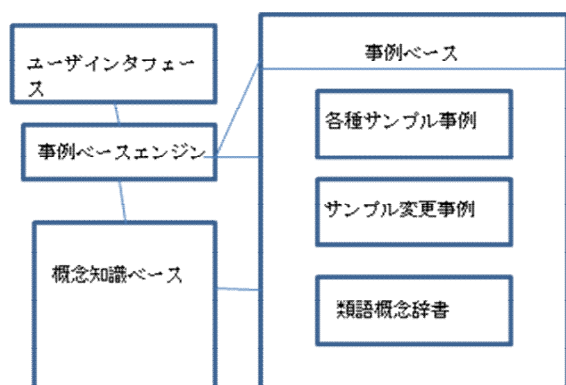


図. 1 事例ベース推論システム構成図

7. 考察

オブジェクト指向ソフトウェア開発では比較的低学年の学生からの教育でサンプル事例からの類推による教育支援が有効なケースが多い。事例ベース推論の適用が比較的に有効な環境が整っている。ただ事例ベース適用には, 具体的サンプル例といっても, 抽象度の高いレベルの概念と具体的のコードレベルの知識の協調メカニズムが必要になる。

類似度に関しては, 類語辞書による直接的類似度評価で, かなりの部分が類似例で対応可能であるが, 3段階論法的な類似性「間接的な類似性導出メカニズム」の検証も必要である。

ポリモーフィズムへの対応も鍵となる。例外事例に対する対策も一定の規則性があり, これに焦点をあてた事例ベースが有効になる。デバッグの際の学生が起こす典型的な誤り事例と対策についても事例ベース推論が有効な局面が多い。

ソフトウェア開発環境で開発ツールのバージョンによる変更でトラブルとなるケースが多いが, 事例ベースのリトリバルシステムで変換事例による対処が重要になる。

ケーススタディ段階でも適切な結果が誘導されるケースだけでなく, 適切でない結果が誘導されるケースも存在する。このケースの対処法についてさらなる検討が必要である。

8. 結論

本考察では, オブジェクト指向開発における事例ベース推論の適用性を検討した。現在, 一部のプロトタイプの開発を試行している。事例ベース推論に関して, ご助言, ご指導いただいた東北大学白鳥教授に感謝する。

9. 参考文献

- [佐藤 1998] 佐藤英人, オブジェクト指向がわかる本, オーム社, 1998
- [吉村 1993] 吉村晋, 黄金法, 白鳥則郎; 事例ベース推論を適用した通信ソフトウェア開発支援環境, 人工知能学会誌, Vol.8, No.6, 1993
- [Chang-Fa 1995] Ching-Fa Hung, Susumu Yoshimura, Takuji Karahashi, Norio Shiratori: A New Specification Environment for Communication Systems Based on Specification Reuse by the Application of Case Based Reasoning, IEICE Trns. INF & SYST, Vol.E78-D, NO.10, 1995
- [唐橋 1995] 唐橋拓史, 吉村晋, 白鳥則郎: 事例ベース推論に基づく仕様記述環境における類語辞書を用いた利用者要求獲得, 情報処理学会, マルチメディア通信と分散処理研究, 1993-5
- [吉村 2009] 吉村晋, 山本昇志, 鈴木弘, 斎藤敏治, ソフトウェア創造実習におけるグループ学習の推進, 情報処理学会第71回全国大会, 6A-2, 2009
- [Calero 2004] Petro A. Gonzalez Calero, Arcano: Case Based Framework Documentation and their Example-Based Learning TIC2002-01961, Jornada de Seguimiento de Proyectos, 2004, Programa Nacional de Tecnologias Informaticas.
- [Watson 1997] Iwan Watson, Applying Case-Based Reasoning Morgan Kaufman Publisher Inc., 1997
- [結城 2004] 結城浩, Java 言語で学ぶデザインパターン入門, ソフトバンク出版, 2004
- [山下 2003] 山下浩一, 戸室悦子, 門屋浩一, Java アドバンスドテクニック 逆引き大全 500の極意, 秀和システム, 2003