

## 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ついて On the Case Based Reasoning about Object Oriented Software Development

吉村 晋<sup>\*1</sup>  
Susumu Yoshimura

白鳥則郎<sup>\*2</sup>  
Norio Shiratori

### 1. はじめに

オブジェクト指向ソフトウェア開発では多くのソフトウェア部品を再利用する。設計段階も仕様部品等を再利用する仕組みが考案される<sup>[1]</sup>。その一つが継承を利用する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時の利用パターンに関する仕組みでデザインパターンと呼ばれる。仕様段階の部品再利用は過去にも多くの試みがなされ、著者らは通信ソフトウェア仕様設計を事例ベース推論で行う試みを行った。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<sup>[2,3,4]</sup>。当時は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は、パターン等の再利用によって開発支援する手法に適している。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<sup>[5,6,7]</sup>。本報告は、事例ベース推論の適用性を考察するとともに、高専(大学)のソフトウェア教育での事例ベース推論<sup>[8]</sup>の適用性について述べる。

### 2. フレームワークとデザインパターン

簡単なフレームワークはクラス継承でなされる。抽象クラスはフレームワーク例である。複雑なフレームワークは、Struts, Struts2, JSF, Searser2, Spring, Hibernateなどの多くのクラス群からなる。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。一方、デザインパターンはパターンの類似性に着目し、再利用を前提として考案される。デザインパターンでは、「GOF」のデザインパターンが知られている<sup>[9]</sup>。

①生成関連パターン(5): Abstract Factory 等

②構造関連パターン(7): Adapter, Bridge 等

③振る舞い関連パターン(11): Command, Interpreter 等

これらのパターンは、構成要素、目的、協調関係、動機、実装、適用可能性、サンプルコード、構造、使用例、関連するパターン項目でカタログ的にまとめられ、目的パターンを選ぶことができる。この項目は、事例ベースの辞書項目と考えてもよい。たとえば Interpreter パターンは言語から別の言語への書き換えをパターン化する。

### 3. 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、過去にも仕様レベルでの適用を試みている<sup>[3]</sup>。しかしフレームワークやデザインパターンで

は、抽象度が高い類似性を対象にする。この類似性は部品再利用を抽象的に捉えるため、サンプルパターンの適用には一定の知識や経験が必要になる。これに対し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、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、抽象度の高すぎるメカニズムは効果的でない。

### 4. 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(例:Java プログラミング学習)が事例学習とマッチする。ソフトウェア教育が言語や文法学習や設計学習になる傾向が強く、学生が達成感を得られないケースが多い。具体的に開発を体験させることが重要になる<sup>[5]</sup>。この対策として良好な手段が事例から学ばせることである。事例からの改良を積極的に試みる。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。

- ・学生の観点に立った事例ベース推論。

- ・類似性をよりどころにソフトウェア構築を行う  
具体的な事例に適した類似性

- ・対象:高専3年から5年(18歳から20歳)

- ・事例:抽象度を低く設定。

抽象度の高い事例は、ソフトウェア開発経験がほとんどないため、受け入れられない。具体的なプロダクトベースのアプローチが有効である。

「容易に受け入れられる事例の範疇」

1. ウィンドウ周りの部品のサンプル事例(ボタン、メニュー、スレッド、画面がちらつかない表示)。
2. 学生の好むサンプル事例(シューティング、ロボコード、ゲームプログラミング、モグラたたき等)
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等。
4. 辞典的なサンプル事例<sup>[10]</sup>

機能的な類似性を取り出し、CBR手法を取り入れる。検討段階だが、初学者でも索引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る。ただ見かけ上の類似性も多く、多相性(ポリモーフィズム)による齟齬も生じうる。またAppletとSwing部品の違いや、画像データ、音楽データに対する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。

### 5. 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。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。Fig.1に質問テンプレートの

\*1 サレジオ高専, Salesian Polytechnic

\*2 東北大学 電気通信研究所, RIEC Tohoku University

概要, Fig.2 にシステム構成図を示す. また類似性の計算メカニズムを以下に示す.

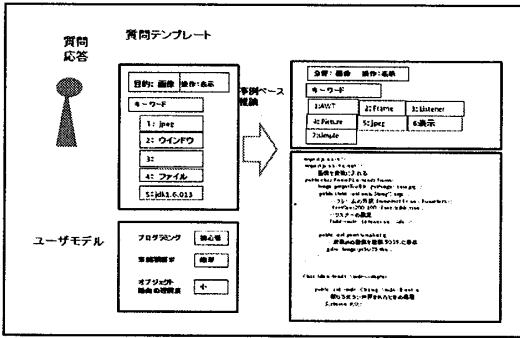


Fig.1 質問テンプレートの概要

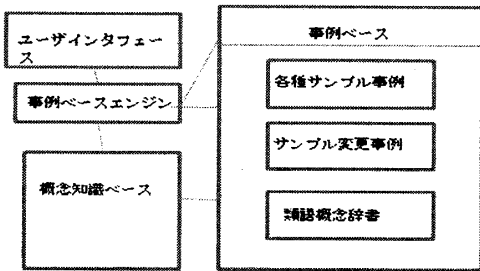


Fig.2 事例ベース推論システム構成図

キーワードと問い合わせ項目との類似性

1. 質問項目(キーワード集合)とパターンキーワード集合の類似性

- 3つの類似性 $g_1, g_2, g_3$  とその結合関数  $f$  で求める.

- 1). 分野の類似性(分野は独立とする)  
分野  $F_i$  と  $F_j$  の類似性

$$g_1(F_i, F_j) = \begin{cases} 1 & : F_i = F_j \\ 0 & : F_i \neq F_j \end{cases}$$

- 2). キーワード集合とキーワード集合の類似性

$$g_2(K_i, K_j) = \frac{(\frac{1}{n} \sum_{k=1}^n w(a_k, K_j) + \frac{1}{m} \sum_{k=1}^m w(b_k, K_i))}{2}$$

キーワード $a$ とキーワードバンク $B$ の類似性

$$w(a, B) = \max_{b \in B} S(a, b)$$

$S(a, b)$  の求め方

①  $a=b$  なら  $S(a, b)=1$

②  $a \neq b$  なら, 2つの方法で求める.

- $a$  と類似な  $b$  が存在して値が  $\alpha$  ならば,  $S(a, b)=\alpha$
- $a$  と類似なものが存在しないならば,  $S(a, b)=0$
- $a$  がキーワードバンクに存在なく,  $S(a, b)$  が  $a$  と  $b$  の文字列マッチングにより, 計算された値  $\beta$  のとき,  $S(a, b)=\beta$

- 3). 付加情報と付加情報の類似性

$A_i$  と  $A_j$  の類似性

$$g_3(A_i, A_j) = L(h_1(U_i, U_j), h_2(P_i, P_j))$$

ここで  $U$  は, ユーザーモデルに関する類似性の付加情報

$P$  は, プログラムに関する類似性に関する付加情報

$$f(g_1(F_i, F_j), g_2(K_i, K_j), g_3(A_i, A_j)) = \begin{cases} 0 & : g_1(F_i, F_j) = 0 \text{ または } g_3(A_i, A_j) = 0 \text{ のとき} \\ \alpha * g_2(K_i, K_j) + \beta * g_3(A_i, A_j) & : \text{その他のとき} \end{cases}$$

$$0 \leq \alpha, \beta \leq 1, \quad \alpha + \beta = 1$$

6. 考察

オブジェクト指向ソフトウェア開発では低学年の学生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非常に多い. 事例ベース推論の適用が有効な環境が整っている. ただ具体的な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対応可能であるが, 間接的な類似性導出メカニズムの検証も必要である. ポリモル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, 事例ベースの事例変更システムでの対処が重要になる.

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.

7. 結論

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ついて検討し, 部分的なプロトタイプの開発を試行している.

ご助言頂く東北大学 木下哲男教授, 千葉工業大学 菅原研次教授に感謝する.

8. 参考文献

[1] 佐藤英人:オブジェクト指向がわかる本, オーム社, 1998  
 [2] 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  
 [3] 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  
 [4] 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得, 情報処理学会, マルチメディア通信と分散処理研資, 1993-5  
 [5] 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  
 [6] 吉村晋: オブジェクト指向開発における事例ベース推論に関する一考察, 人工知能学会全国大会, 2009-6  
 [7] 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.  
 [8] Iwan Watson: Applying Case-Based Reasoning, Morgan Kaufman Publisher Inc., 1997  
 [9] 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  
 [10] 山下浩一, 戸室悦子, 門屋浩一: Java アドバンスドテクニック 逆引き他以前 500 の極意, 秀和システム, 2003