

# ユースケース記述の解析結果を利用した デザインパターンの推薦に関する研究

中村健二<sup>†</sup> 田中成典<sup>‡</sup> 池辺正典<sup>†</sup> 細島啓史<sup>†</sup> 伊達浩典<sup>‡</sup>  
 関西大学大学院総合情報学研究科<sup>†</sup> 関西大学総合情報学部<sup>‡</sup>

## 1. はじめに

近年、オブジェクト指向によるソフトウェア設計開発が普及している。それに伴い、設計の品質向上のためにソフトウェアの設計時に頻繁に発生する問題とその解決策を記述したデザインパターン[1]の利用が増加[2]している。しかし、適切なデザインパターンを適用するためには、ソフトウェア設計開発に関する知識と経験が必要となる。そのため、既存研究[3]では、デザインパターンを推薦する手法として、ユースケース記述とデザインパターン判断規則とのパターンマッチを行い、デザインパターンの判定を行う手法が提案されている。しかし、この手法は、各デザインパターンに用意されているルールとのパターンマッチでデザインパターンを判別しているため、汎用性が低いという問題がある。そこで、本研究では、Web リソースから各デザインパターンの解説箇所を特定[4]し、デザインパターンに特化したコーパスを作成する。そして、そのコーパスと判定対象のユースケース記述を利用してベクトル空間モデル[5]でデザインパターンの判定を行うことで、精度の高いデザインパターンの適用支援を目指す。

## 2. 研究の概要

本研究では、Web リソースと GoF (Gang of Four) のデザインパターンカタログ[1]から作成したコーパスを利用して、ユースケース記述から適用可能なデザインパターンを判定することを目的とする。本システムは、図1に示すように、1) コーパス作成機能、2) デザインパターン判定機能の2つの機能により構成される。

### 2.1 コーパス作成機能

本機能では、Web リソースとデザインパター

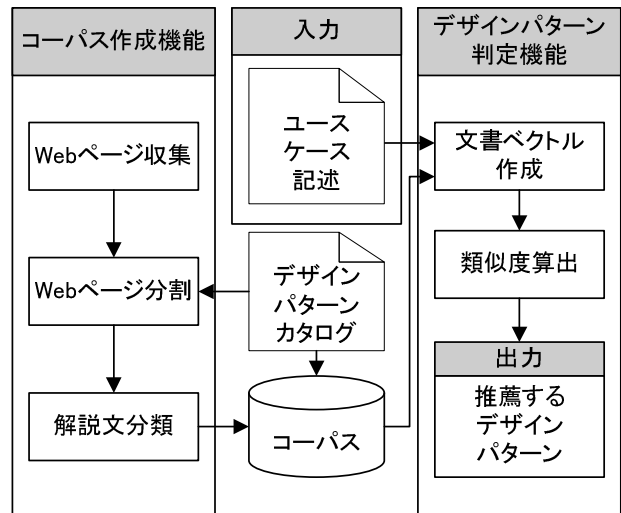


図1 システムの流れ

ンカタログからデザインパターンの解説文を収集し、デザインパターン毎に分類してコーパスに登録する。本機能は、Web ページ収集処理、Web ページ分割処理、解説文分類処理の3つの処理により構成される。

#### 2.1.1 Web ページ収集処理

本処理では、Google SOAP Search API を利用して、デザインパターンに関する Web ページを検索し、その Web ページを取得する。

#### 2.1.2 Web ページ分割処理

Web ページ内には、デザインパターンの解説文以外にもメニュー部分や広告部分などの不要な部分や、複数のデザインパターンの解説文が含まれる場合がある。これらの Web ページをそのまま登録すると、デザインパターンの推薦精度低下の原因となる。そこで、本処理では、VIPS (a VIsion-based Page Segmentation) アルゴリズム[4]を利用することで、Web ページの内容をブラウザ上の表示に従ってブロック単位に分割し Web ページから必要な箇所のみを収集可能にする。

#### 2.1.3 解説文分類処理

本処理では、デザインパターンの解説文を分類してコーパスに登録する。デザインパターン

Research on Recommending Design Pattern Analyzing Use Case Description

<sup>†</sup>Kenji Nakamura, Masanori Ikebe, Hirofumi Hosohata  
 Graduate School of Informatics, Kansai University, 2-1-1  
 Ryozenji-cho, Takatsuki-shi, OSAKA 569-1095, Japan

<sup>‡</sup>Shigenori Tanaka, Hironori Date  
 Faculty of Informatics, Kansai University, 2-1-1 Ryozenji-cho,  
 Takatsuki-shi, OSAKA 569-1095, Japan

表1 実験結果

	Decorator	Abstract Factory	Composite	Strategy	全体
ユースケース記述数	3	3	3	3	12
正解数	3	3	1	0	7
一致率 (%)	100	100	33	0	58

の定義であるデザインパターンカタログ内の解説文をコーパスに登録する。そして、ベクトル空間モデルを利用して、ブロック内の解説文とデザインパターンカタログ内の解説文の類似度を算出し、該当するデザインパターンに一致するコーパスに解説文を登録する。

## 2.2 デザインパターン判定機能

本機能では、入力されたユースケース記述とコーパスを利用して、本システムが推薦するデザインパターンを判定する。まず、コーパスからデザインパターン毎の文章情報を抽出する。次に、ベクトル空間モデルを利用しユースケース記述とコーパスに登録されている各デザインパターンの解説文の類似度を算出する。そして、類似度が最も高いデザインパターンを推薦結果として出力する。最後に、システムが推薦するデザインパターンをユーザに提示する。ここで、ユーザの理解やシステムへの適用を支援するためにデザインパターンカタログの内容もあわせて提示する。

## 3. システムの実証実験と考察

実証実験として、本システムの有効性を確認するために、経験者が実際に適用したデザインパターンと本システムが推薦したデザインパターンを比較した。今回実験対象は、参考文献[1]の第2章で紹介されているテキストエディタの事例とする。

### 3.1 実証実験

本実証実験では、まず、デザインパターン名23個を検索語としてWeb検索を行い1つのデザインパターンで100件ずつ合計2,300件の検索結果とデザインパターンカタログを利用してコーパスを作成した。次に、デザインパターン判定機能にユースケース記述を入力し、本システムが推薦するデザインパターンを決定した。最後に、ドキュメントエディタの事例で実際に適用されているデザインパターンと本システムが推薦したデザインパターンとの一致率を算出し、その結果について考察を行った。

### 3.2 結果と考察

実証実験に使用したドキュメントエディタに適用されている各デザインパターンと本システムが推薦したデザインパターンとの一致率を表1に示す。この結果から、システムに適用され

ているデザインパターンと本システムが推薦したデザインパターンの一致率は、全体では58%となった。各デザインパターンの推薦結果を確認したところ、Compositeパターンの誤判定2つはいずれもVisitorパターンであった。CompositeパターンとVisitorパターンはデザインパターンとして類似しているため、各コーパスの特徴語が重複してしまったと考えられる。また、Strategyパターンの結果を確認したところ、Template Methodパターンと誤判定していた。ここで、入力したユースケース記述とデザインパターンカタログのみを比較したところ正常にStrategyパターンと判定された。このことから、コーパス作成段階でStrategyパターンの分類精度が低かったと考えられる。DecoratorパターンとAbstract Factoryパターンについては、全ての推薦結果が一致した。

## 4. おわりに

本研究では、ユースケース記述から自動的にデザインパターンを推薦する手法を提案した。実証実験では、経験者が実際に適用したデザインパターンと本システムの推薦結果を比較して、本システムの有用性を確認した。今後は、ユースケース記述以外の仕様書情報も考慮して、より高精度にデザインパターンを推薦する手法へ発展する予定である。

## 参考文献

- [1] Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides : オブジェクト指向における再利用のためのデザインパターン改訂版, ソフトバンクパブリッシング, 1999.11.
- [2] Gary Moynihan, Abhijit Suki, Daniel Fonseca : An Expert System for the Selection of Software Design Patterns, Expert Systems, Blackwell Publishing, Vol.23, No.1, pp.39-52, 2006.2.
- [3] 木梨充高, 小飼敬, 上田賀一 : デザインパターン利用促進のためのモデリング支援ツールの開発, ソフトウェア工学研究会研究報告, 情報処理学会, Vol.2004, No.30, pp.67-74, 2004.3.
- [4] Deng Cai, Shipeng Yu, Ji-Rong Wen, Wei-Ying Ma : VIPS: a Vision-based Page Segmentation Algorithm, Microsoft Technical Report, Microsoft Research, MSR-TR-2003-79, 2003.11.
- [5] Gerard Salton, Michael McGill : Introduction to Modern Information Retrieval, McGraw-Hill, 1983.9.