文書中のパターン間の文書類似度による関連分析

Analyzing Relationship between Patterns in Documents based on Document Similarity

久保 淳人♥ 鷲崎 弘宜♦
高須 淳宏♦ 深澤 良彰♠

Atsuto KUBO Hironori WASHIZAKI
Atsuhiro TAKASU Yoshiaki FUKAZAWA

業務活動において繰り返し現れる問題とその解法をパターンとして記述することにより，実証済みの知識やノウハウを効率的に共有し再利用することができる．パターンは組み合わせて利用することで単一のパターンが扱うよりも大きな問題を扱うことができるが，有用なパターンの組み合わせを人手で分析するのは人的コストが高い．本稿では，計算機を用いて，パターン文書からパターンの情報を抽出し，パターン間の関連を自動的に分析する手法を提案する．提案手法を GoF デザインパターンに対して適用した結果，提案手法の実装システムは対象パターン文書に明示的に記述されたパターン間の関連を自動的に識別し，さらには，明示されていないパターン間の関連を示唆した．

In business activity, we can share and reuse proven knowledge and know-how effectively by describing problems, that we face repeatedly, and solutions as patterns. Teamed patterns treat bigger problem than that each the patterns treat. However, it costs a lot that the analysis of useful pattern pairs by human power. In this paper, we propose a technique for extracting information of patterns from pattern documents. We also propose a technique for identifying the appropriate relations between patterns automatically. As a result of applying our technique to GoF's design patterns, the implemented system has analyzed relations between patterns without explicit information, and has implied the new relations.

1. はじめに

業務活動において発生する問題の一部は，類似の状況において繰り返し発生する．業務の熟練者は，そのような問題を対応する解法との対として記述し，類似の状況においてその対を再適用することで業務の効率化を試みることがある．解決に際しては，制約条件(例えばソフトウェア開発であれば保守性や時間効率，空間効率など)が考慮される．このような問題・制約条件・解法の組を明示的に記述することで，問題解決に関する知識を効率的に共有および再利用できる．

♥ 学生会員　早稲田大学大学院理工学研究科修士課程
a.kubo@fuka.info.waseda.ac.jp
♦ 正会員　情報・システム研究機構 国立情報学研究所
washizaki@nii.ac.jp, takasu@nii.ac.jp
♠ 正会員　早稲田大学理工学部 fukazawa@waseda.jp

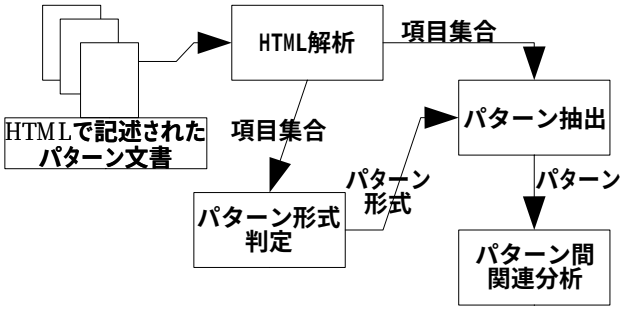

図 1 提案手法のブロック図
Figure 1: Block Diagram of Proposed Technique

パターンとは，こういった問題解決に関する知識を効率的に共有および再利用することを目的として，業務活動で繰り返し現れる状況について，解決すべき問題と制約条件(フォース; Force)，問題に対する実証済みの解法を明示的に記述したものである．これまでに，都市環境の建築・設計[1]，開発組織のマネジメント[2]，ソフトウェアの設計[3]といった様々な分野において，パターンによる知識の記述が試みられている．本稿では，問題と問題が発生する状況を合わせて文脈(Context)と呼ぶ．パターンを記述した文書がパターン文書であり，一連のパターン文書をパターンカタログと呼ぶ．パターン形式は，パターン文書の書式，および，パターン文書に記述される情報を規定する．

同一の問題について，いくつかのパターンが異なる解法を与えることがある．パターンの利用者は，直面した問題に適用可能なパターン間でフォースを比較し，最も適切と判断したパターンを用いる．さらに，複数のパターンを組み合わせ，また連続して適用することで，より大きな問題を解決する．このとき，複数のパターン間の関連があらかじめ明らかになっていれば，パターンの組み合わせや連続適用を円滑に行うことができるので，パターン間関連の分析はパターン利用に有用であると考えられる．これまでに，パターン間関連分析の試みはいくつか報告されている[5][6]．

パターン利用促進のために，より多数のパターン間の関連分析や，パターンカタログをまたいだ関連分析が望まれる．しかし，パターン間関連分析は人的コストが高く，パターン数の増加に伴って組み合わせの個数が爆発的に増加するため，多数のパターン間関連をすべて人手で分析するのは非常に困難である．例えば，WWW における代表的なパターンリポジトリである[4]には，2004 年 12 月時点で 700 個以上のパターンが公開されているが，それら全ての組み合わせを人手で分析することは現実的でない．そこで，本稿では，HTML を用いて英語で記述されたパターン文書の解析とパターン間関連の分析を，計算機を用いて自動的に行う手法を提案し，大規模なパターン間関連分析における問題の解決を図る．提案手法を用いることで，パターン間関連分析が容易になる．また，人手による分析では気付かなかったパターン間関連に関する示唆を得ることが期待される．

2. 提案手法

提案手法の全体を図 1 に示す．提案手法は，大きくパターン抽出とパターン間関連分析に分かれる．

2.1 パターンのモデル化

計算機でパターンを扱うために，モデル化が必要である．モデルに含める情報は，間関連分析の目的に依存する．本稿では，パターン間関連分析の目的を「組み合わせや連続的な

適用によって，個々のパターンが扱う問題よりも大きな問題を扱うことが可能となるパターンの組を得ること」とする．

同じ問題に対するパターンは組み合わせて利用できる可能性があるので，文脈(問題や問題が現れる状況)を，本稿におけるパターンモデルに含める．パターン適用前後の文脈をそれぞれ開始文脈(Starting Context)，結果文脈(Resulting Context)とし，パターンをパターン適用による文脈の遷移と考える(図 2)ことで，連続的なパターン適用を扱うことができる．さらに，パターン利用者はフォースを検討してパターン利用を決定するので，フォースもモデルに含めるのが妥当である．対して，解法はパターン適用の際に必要であるが，パターンの利用を検討する段階においては不要な情報であるので，モデルに含めない．以上から，本稿ではパターンモデルの構成要素を，開始分脈，フォース，結果文脈とする．計算機上では，実際にはこれらの各要素は文書片として表現される．文脈を表す文書片集合を C，フォースを表す文書片集合を Λ として，パターン p を，$p=(c_s,c_r,\lambda)$（$c_s,c_r\in C$，$\lambda\in\Lambda$）と定義する．

2.2 パターン文書の解析

World Wide Web(WWW)には HTML を用いて記述された多数のパターン文書が存在する．これらのパターン文書からパターンモデルに沿って文書片を抽出したい．代表的なパターン形式に従ったパターン文書は項目のリスト構造を持つ[7]．項目は見出しと本文の組である．そこで，見出しの集合 H と本文の集合 B を用いて，パターン文書 d を $d=\{(h,b)\mid h\in H,b\in B\}$ と定義する．図 3 は，[3]に含まれるパターンの一つを，HTML を用いて記述したパターン文書である．この HTML 文書では，見出しを h2 タグ，本文を p タグや li タグでマークアップしていることが分かる．このように，パターン文書においては見出しや本文に対して規則的なタグ付けがされていることが多い．そこで，HTML 解析器を考える．HTML 解析器は見出し状態，本文状態，空状態の 3 状態をとり，タグの出現によって状態遷移する．各状態では，それぞれ出現したテキストを見出しや本文とみなし，空状態では破棄する．例のパターン文書において，HTML 解析器に例で挙げた見出しタグ，本文タグを設定すれば，Name，Motivation といった見出しを持つ項目集合を得られる．

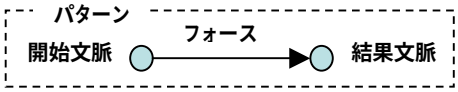

図 2 パターンモデル
Figure2: The Pattern Model

```
 1: <html><head>
 2: <title>Command Pattern [GoF]</title>
 3: </head><body>
 4:  <h1>Command Pattern [GoF]</h1>
 5:  <h2>Name</h2>
 6:  <p>Command</p>
 5:  <h2>Classification</h2>
 6:  <p>Behavior, Object</p>
 7:  <h2>Motivation</h2>
 8:  <p>Encapsulate a request as a parameterized …</p>
 9:  <h2>Applicability</h2>
10:  <p>More flexibility on managing an execution…</p>
11:  <h2>Collaborations</h2>
12:  <ol>
13:     <li>Client creates commands as needed, …</li>
14:  </ol>
15:  <h2>Consequences</h2>
16:  <p>Decouples an object from the …</p>
17:  <h2>Related Patterns</h2>
18:  <p>Commands may be assembled using …</p>
19:</body></html>
```

図 3 Command パターンの HTML 文書
Figure3: a HTML Document of Command Pattern

代表的なパターン形式は文書構造については共通するが，項目見出しはそれぞれ異なる．そこで，m 個の見出しからなるパターン形式 f を $f=\{h_1,h_2,\Lambda,h_m\}$ とする．パターン文書 d の見出し集合を $h(d)$ とするとき，パターン文書 d のパターン形式 f への適合度を次の式で算出する．

$$ad(d,f)=\frac{|h(d)\cap f|}{|h(d)\cup f|}$$

図 3 のパターン文書 d について，表 1 の各パターン形式に対する適合度を算出し，例えば，GoF 形式，Coad 形式，PoSA 形式 [7] でそれぞれ適合度 $ad(d,f_{GoF})=0.538$，$ad(d,f_{Coad})=0.083$，$ad(d,f_{PoSA})=0.105$ を得たならば，このパターン文書は GoF 形式に従うと判断できる．

表 1 代表的なパターン形式の見出し
Table1: Headings of Famous Pattern Forms

パターン形式	見出し集合
GoF 形式	f_{GoF} = { Pattern Name, Classification, Also Known As, Motivation, Applicability, Structure, Particiants, Collaborations, Consequences, Implementation, Sample Code, Known Uses, Related Pattern }
Coad 形式	f_{Coad} = { Name, Family, Typical Object Interactions, Examples, Combinations, Note }
PoSA 形式	f_{PoSA} = { Name, Also Known As, Examples, Context, Problem, Solution, Structure, Dynamics, Implementation, Example Resolved, Variants, Known Uses, Consequence, See Also }

代表的なパターン形式について，各項目とパターンの各要素との対応表をあらかじめ作成し，項目集合からパターンを得る．GoF 形式ならば，Motivation の項目が開始文脈に，Applicability がフォースに，Consequences が結果文脈にそれぞれ対応する．

2.3 パターン間関連

人手によるパターン間関連分析では，分析者がパターン間の関連を深く理解して分析を行うが，計算機を用いて同等の処理を行うのは困難である．提案手法では，パターンを構成する文書片間の類似度を利用したパターン間関連分析を行う．以下の文書片の組み合わせが考えられる．

開始文脈同士の類似: 開始文脈が類似する 2 つのパターンは，同じ問題に対する異なった解法を与える．

結果文脈同士の類似: 結果文脈が類似する 2 つのパターンは，類似した適用結果をもたらす．

結果文脈と開始文脈の類似: あるパターン p1 の結果文脈と，別のパターン p2 の開始文脈との類似度が高い場合，p1 の適用後に連続して p2 を適用できる．

2.4 パターン間関連の分析

文書解析の結果得られたパターンから，文書片を取り出し，類似度を算出して関連の強さとする．分析に先立って，文書中の各単語に対して[8]の不要語リストによる不要語削除処理，および，Paice/Husk の方法による接辞処理[9]を行う．

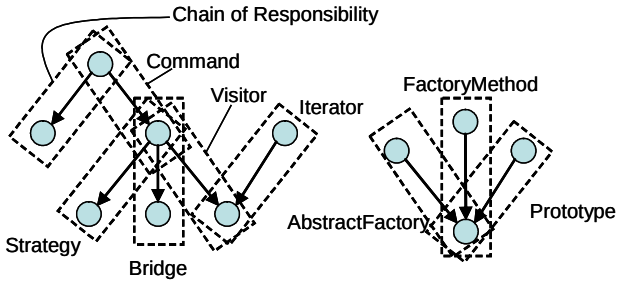

図 4 PRG の例
Figure 4: PRG Examples

文書間の類似度は，TF-IDF 法による単語重み値の文書ベクトル間の cosine similarity[10]を用いて算出する．

TF-IDF 法はある文書片 s 中の単語 t の重みを算出する手法である．文書片 s 中の単語 t の出現回数を $tf(s,t)$，単語 t を含む文書片数を $df(t)$ とする． N 個のパターンからなるパターン集合 P は $3N$ 個の文書片を含む．このとき，単語重みを以下の式で定義する．

$$tfidf(s,t) = tf(s,t)\left(\log\frac{3N}{df(t)}+1\right)$$

パターン集合中の全文書片，全単語について単語重みを算出すれば，その全体は，文書片の総数を m，単語の総数を n とすると m 行 n 列の行列となる．行ベクトルはそれぞれ文書片に対応する．文書片 s_1,s_2 に対応する行ベクトルを v_1,v_2 とするとき，文書片間の類似度を以下の式で定義する．$|v|$ は v のユークリッドノルムである．

$$\mathrm{sim}(s_1,s_2) = \frac{v_1 \cdot v_2}{|v_1||v_2|} \quad (1)$$

分析によって，全てのパターン間の全ての組み合わせについて関連の強さが得られる．得られた関連を関連の強さの大きい順に並べ，ある特定の順位までの関連を実際に関連があるものと見なしてパターンの文脈に対応する頂点を繋げていけば，図 4 のようなパターン関連グラフ(Pattern Relation Graph; PRG)[11]が得られる．

3. 実験

提案手法を実行するシステムを Java 言語により実装し，実験を行った．実験結果の評価には再現率および精度[10]を用いる．再現率と精度はトレードオフの関係にあるので，再現率-精度グラフおよび 11 点平均精度[10]を用いて比較する．

入力データとして，GoF デザインパターン[3]を記述した 2 つのパターン文書群 A，B[12][13]を用いた． A，B は，それぞれ 22,23 個のパターン文書を含む．パターン文書には Related Patterns の項目に関連パターンが明示されているが，実験では同項目の内容を破棄した．

実験 1,2 では，パターン文書内に明示されたパターン間関連の情報を用いずにパターン間関連分析を行い，性能を評価する．パターン文書群 A(実験 1)および B(実験 2)について，それぞれ全ての組み合わせについて関連の強さを算出し，強い順に並べる．入力パターン文書において Related Patterns の項で明示されている関連を正解として，閾値を最上位から最下位まで変化させて，それぞれ再現率および精度を求める．提案手法では 3 種のパターン間関連が得られるが，最も強い関連をシステムの出力とする．

実験 3,4 では単純な文書間類似度による関係分析に対する提案手法の優位性を評価する．パターン文書群 A,B をあわせたパターン文書群を入力し，同一の GoF デザインパターンを記述したパターン文書の組を正解として，実験 1,2 と同じ方法で再現率・精度を求める．実験 3 では提案手法による関連，実験 4 では，2.2 の処理を行わず，単に式(1)を用いた関連を用いる．

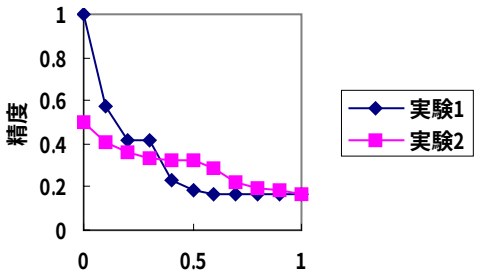

図 5 実験 1,2 の再現率-精度グラフ
Figure5: Recall-Precision Graph for Ex. 1 and 2

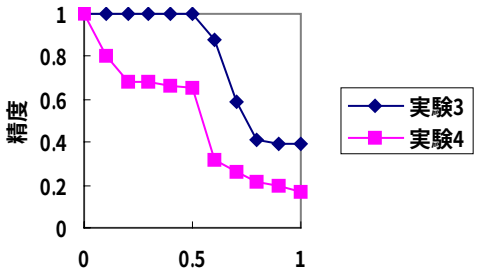

図 6 実験 3,4 の再現率-精度グラフ
Figure6: Recall-Precision Graph for Ex. 3 and 4

実験 1,2 の結果を，再現率-精度グラフとして図 5 に示す．11 点平均精度はそれぞれ 0.333 および 0.301 となった．実験 1 で本システムが判定した関連を，関連の強さの大きい順に表 2 に示し，PRG を用いて視覚的に表現した(図 4)．表 2 には，得られた関連がパターン文書に明示されていたかどうか，および，関連の種類も合わせて示した．例えば AbstractFactory，FactoryMethod，Prototype パターンは類似した適用結果をもたらすこと，および Command パターンの適用後に Visitor パターンが適用可能であることが分かる．正解に含まれていない Chain of Responsibility パターン・Command パターン間に何らかの関係があることを示唆している．これらの関係は，もともとのパターンの作者によって明示されていないものの，以下の理由から妥当なものであると考えられる．

Chain of Responsibility パターンの目的は，同じインタフェースを持つオブジェクトにそれぞれ責任を割り当て，委譲を繰り返すことで全体として責任を果たすことである．Command パターンは，処理をコマンドオブジェクトにカプセル化することで，1 つのオブジェクトの責任範囲を小さくし，動的な処理順序の変更を可能にする．これらのパターンは，オブジェクトに割り当てる責任を小さくするという目的において共通している．

実験 3,4 の結果を，再現率-精度グラフとして図 6 に示す．11 点平均精度は，それぞれ 0.789 および 0.514 を得た．実験 3，4 の結果，2.2 節で示した処理が有効であることが分かった．これは，提案手法を適用することで，文書のヘッダやフッタ等の重要でない情報を効率的に除去できたためである

と考えられる. また, 本システムはパターン文書群 A,B が GoF 形式に従うことを正しく検出した.

4. 関連研究

Borchers は, パターンを項目のリストとしてモデル化し, 関連するパターンの集合をパターンランゲージとして捉えている[14]. パターンを項目集合として捉える点は提案手法と共通している. Borchers のモデルはより多くの情報を保持することができるが, 提案手法のモデルは, パターン適用の流れを得るために一部の情報を破棄してモデルを特化させている.

Zimmer らは, GoF デザインパターン間の関連として, 新たに連続した適用関係および類似関係を考案し, それらを用いて GoF デザインパターンの体系化を試みている[6]. 提案手法では, パターン関連グラフを用いて, それらの関係をより詳細に表すことができる.

Ong らは, フォース間の協調や排他といった関連に基づくパターンの体系化を試みている[5]. 今後, 提案手法に意味解析処理を追加して, そのようなフォース間の関連分析を行う予定である.

5. おわりに

本論文では, HTML を用いて記述されたパターン文書からパターンを抽出し, パターンを文脈の変換として捉える新たなモデルに基づいて, パターン間の関連を自動的に分析する手法を提案した. 実験の結果, パターン文書に明示された情報を用いずにパターン間の関連を分析し, 互いに関連するパターン集合を得た. また, パターンの作者によって明示されていないものの, 実際には関係のあるパターン間関連を自動的に分析することについても, 提案手法が有効であることが分かった.

正解データ作成の限界により, 本稿では小規模な実験にとどまったが, 今後は, 多数のパターンにまたがったパターン間の関連を分析することを予定している. また, フォースを利用したパターン関連分析の提案, テキスト処理技術を用いた分析精度の向上を目指す.

分析結果をパターン文書リポジトリとして用いることで, 汎用的な検索エンジンに比べてより高度なパターン検索が期待される. 例えば, 直面した問題について検索すると適用可能な複数のパターンを提示するといった利用方法が考えられる. また, 提案手法を実装したシステムは, 人間が気付かないパターン間関連について示唆を提供するので, パターン間の関連についての研究を支援できると考えられる.

表 2 実験 1 で得られたパターン間関連
Table 2: Pattern Relations in Ex.1

順位	パターンp1	パターンp2	強さ	種類	明示
1	ObjectAdapter	ClassAdapter	1	S&S	○
2	ChainofResponsibility	Command	0.44	S&S	×
3	Visitor	Iterator	0.43	R&R	○
4	FactoryMethod	Prototype	0.42	R&R	○
5	State	Memento	0.42	R&S	×
6	Command	Visitor	0.38	R&S	×
7	AbstractFactory	FactoryMethod	0.37	R&R	○

S&S 開始文脈同士　　**R&R** 結果文脈同士
R&S 一方の結果文脈と他方の開始文脈

[文献]

[1] Alexander, C., et al.: A Pattern Language, Oxford University Press (1977).
[2] Coplien, J. O. : A Development Process Generative Pattern Language, Pattern Languages of Program Design, Vol.1, pp. 183–237, Addison-Wesley (1995).
[3] Gamma, E., Helm, R., Johnson, R., and Vlissides, J. : Design Patterns : Elements of Reusable Object-Oriented Software, Addison-Wesley (1995).
[4] Portland Pattern Repository, http://c2.com/ppr/
[5] Ong, H., Weiss, M., and Araujo, I. : Rewriting a Pattern Language to Make it More Expressive, Proceedings of the 6th Annual Southwestern Conference on Pattern Languages of Programs (2003).
[6] Zimmer, W. : Relationships between Design Patterns, Pattern Languages of Program Design, Vol.1, pp. 345–364, Addison-Wesley (1995).
[7] 鈴木純一, 田中祐, 長瀬嘉秀, 松田亮一: ソフトウェアパターン再考, 日科技連 (2000).
[8] freeWAIS-sf 2.x & SFgate 5.x User Guide, http://www-fog.bio.unipd.it/waishelp/waishlp3.html
[9] Paice, C. D. : Another Stemmer, SIGIR Forum, Vol. 24, No. 3, pp. 56–61 (1990).
[10] 徳永健伸:情報検索と言語処理, 東京大学出版会(1999).
[11] 久保淳人, 鷲崎弘宜, 高須淳宏, 深澤良彰: 組み込みシステムパターンに対するパターン間関連分析手法の適用, 組込みソフトウェアシンポジウム 2004 論文集 (2004).
[12] Camp, D. V. : An Object-Oriented Pattern Digest, http://patterndigest.com/.
[13] Huston, V. : Huston Design Patterns , http://home.earthlink.net/ huston2/dp/patterns.html
[14] Borchers, J. O. : A Pattern Approach to Interaction Design, AI & Society: Journal of Human-Centred Systems and Machine Intelligence, Vol. 15, No. 4, pp. 359–376 (2001).

久保 淳人　Atsuto KUBO
早稲田大学大学院理工学研究科修士課程在学中. 2004 年早稲田大学理工学部コンピュータ・ネットワーク工学科卒業. パターン文書研究に従事. 日本データベース学会学生会員.

鷲崎 弘宜　Hironori WASHIZAKI
情報・システム研究機構 国立情報学研究所助手. 2003 年早稲田大学大学院理工学研究科博士課程修了, 博士(情報科学). ソフトウェアパターン, コンポーネントベース開発の研究に従事. 日本データベース学会, 情報処理学会, 電子情報通信学会, 日本ソフトウェア科学会, ACM, IEEE 各会員.

高須 淳宏　Atsuhiro TAKASU
情報・システム研究機構 国立情報学研究所教授. 1989 年東京大学大学院工学系研究科博士課程修了, 工学博士. データベースシステム, 文書画像処理, 機械学習の研究に従事. 日本データベース学会, 電子情報通信学会, 人工知能学会, ACM, IEEE 各会員.

深澤 良彰　Yoshiaki FUKAZAWA
早稲田大学理工学部教授. 1983 年早稲田大学大学院理工学研究科博士課程修了. 工学博士. ソフトウェア工学, 計算機アーキテクチャの研究に従事. 日本データベース学会, 情報処理学会, 電子情報通信学会, 日本ソフトウェア科学会, ACM, IEEE 各会員.