

# Portland Pattern Repository におけるソフトウェアパターンに対するネットワーク分析

鷺崎 弘宜<sup>†</sup> 川村 健<sup>‡</sup> 角谷 将司<sup>†</sup> 深澤 良彰<sup>†</sup>

<sup>†</sup> 早稲田大学基幹理工学部情報理工学科  
早稲田大学グローバルソフトウェアエンジニアリング研究所  
〒169-8555 東京都新宿区大久保 3-4-1

<sup>‡</sup> USOL 東京株式会社 〒135-8560 東京都江東区豊洲 1-1-1

E-mail: <sup>†</sup> washizaki@waseda.jp kage\_purinman@ruri.waseda.jp fukazawa@waseda.jp

**あらまし** ソフトウェアパターンの収集サイト Portland Pattern Repository (以下 PPR) にネットワーク分析の手法を適用し、パターンの参照関係の位置をしめす指標である中心性を算出し、パターン間の参照関係の構造的特徴を明らかにした。また、PPR に記載されている各パターンの履歴情報についてデータ分析を行った。分析の結果、使用頻度の高いと思われるパターンは次数中心性が高い傾向がみられた。このことは適用するパターンの検討や新たなパターンの抽出に有用であると考えられる。

**キーワード** ソフトウェアパターン, ネットワーク分析, パターンマイニング

## Network analysis for Software Patterns in Portland Pattern Repository

Hironori Washizaki<sup>†</sup> Takeshi Kawamura<sup>‡</sup> Masashi Kadoya<sup>†</sup> and Yoshiaki Fukazawa<sup>‡</sup>

<sup>†</sup> Waseda University, Dept Computer Science and Engineering, Global Software Engineering Laboratory  
3-4-1, Okubo, Shinjuku-ku, Tokyo, 169-8555 Japan

<sup>‡</sup> USOL Tokyo Co., Ltd.  
1-1-1 Toyosu, Koto-ku, Tokyo, 135-8560 Japan

E-mail: <sup>†</sup> washizaki@waseda.jp s.murakami-620@fuji.waseda.jp fukazawa@waseda.jp

**Abstract** We applied the technique of network analysis to collection site Portland Pattern Repository (henceforth, PPR) of the software pattern, and centrality was calculated. And the feature of the relation between software patterns was clarified. Moreover, metrics analysis was conducted about the history information of each software pattern. As a result, The degree centrality of the frequently-used software pattern was high value. This is useful to select a software pattern and the extraction of a new software pattern.

**Keyword** Software patterns, Network analysis, Pattern mining

### 1. はじめに

ソフトウェアパターン (以下、パターンと表記) とはソフトウェア開発のさまざまな局面において繰り返し現れる出来事や問題から得られる知識を、再利用できるように抽象化・形式化してまとめたソフトウェア開発に関する問題に対する解法である。

ソフトウェア開発者は開発時に発見する問題に対し、それを解決できるパターンを選択し適用する。また、別のソフトウェア開発時に再利用できそうな知識を抽象化し、新たなパターンを抽出することもある。

パターンを選択する際の参考情報として、パターンの収集・整理を行っている Web サイト Portland Pattern

Repository[1] (以下、PPR と表記) があげられる。参照関係にあるパターンには、ページ間にハイパーリンクを設定し、参照関係を持たせている。PPR のような要素と要素の関係を分析する手法としてネットワーク分析[2]があげられる。要素をノード、関係を線として定義し、要素間の構造的な特徴を分析する。ネットワークの構造的な特徴を示す指標として中心性があげられる。

ソフトウェア開発者は、単独のパターンのみを適用することは少なく、参照関係にあるパターンも適用することが多い[3]。パターンの参照関係において、そのパターンが参照関係の末端付近にある場合、そのパタ

ーンは抽象度が低い特定領域の問題を解決するパターンであることが知られている[4]。そのため、ソフトウェア開発者がパターンを選択する時、選択したパターンが参照関係の末端付近にあった場合、より汎用性の高い抽象的なパターンの適用も検討する必要がある。

そのため、ソフトウェア開発者がパターンを選択する時、選択したパターンが参照関係の中心付近にあるのか、それとも参照関係の末端付近にあるのかを把握する必要がある。しかし、パターン間の参照関係の位置を把握する指標はあまりわかっていない。

また、パターンマイニング時には、パターンの再利用を考慮し、抽出したパターンから参照するパターンも抽出しておく必要がある。参照パターンが参照関係の末端付近にあるパターンの場合、特定の問題領域を解決するパターンである可能性があり、抽出したパターンからの参照関係を設定すべきか妥当性を検討する必要がある。よって、参照パターンについて、パターン間の参照関係の位置を把握する必要があるが、そのための参考となる指標はあまりよくわかっていない。

そこで本稿では、パターン間の参照関係の構造的特徴について明らかにした。また、得られた構造的特徴がパターン利用者にとって有用かどうかを分析検討した。

その背景と分析方法および結果を以下に述べる。

## 2. 背景

### 2.1. Portland Pattern Repository (PPR)

PPRとは、パターン収集・整理を目的とし、「Wiki」の仕組みのもとになったウェブサイトである。PPRはネットワークを介して、複数の利用者がコンピュータ上の文章（ハイパーテキスト）を共同作業で追加・修正・更新できるようにした仕組みである。PPRは新規の書き込みに加えて、すでに公開済みのコンテンツを随時（別の参加者が）書き換えていくという利用スタイルに特徴がある。

PPRでは学術研究機関や企業研究機関などの様々なパターン利用者が抽出した新たなパターンを登録する。または、登録済みパターンに対して追記・修正を行っている。PPRには1パターンの情報を1ページにまとめて登録されており、ソフトウェアパターンの内容以外にも、そのパターンが属するパターンカテゴリーや更新回数、更新内容、最終更新日といった履歴情報も併せて登録されている。

パターンの登録・編集時に、利用者がパターンを他のパターンと参照関係を持たせたい場合、編集時のページのテキストに、参照パターンのページへのハイパーリンクを設定することで参照関係を設定している。ただし、ハイパーリンクの先はかならずパターンとは限らず、技術者・研究者やパターンカテゴリー、パタ

ーンに関する書籍など、パターン以外へのリンクも含まれている。

### 2.2. ネットワーク分析

ネットワーク分析は、社会学や通信ネットワークなどの分野で多く用いられている分析手法で、数学のグラフ(Graph)理論に基礎を置いている。ここでいうネットワークとは、頂点(N: Node)と線(E: Edge)を基本構成要素としている。ネットワークは、線で点と点の関係を示しており、各頂点の持つ線の数を次数と呼ぶ。

ネットワーク分析によって明らかになっているネットワークの特徴には、スモールワールド性、クラスター性、スケールフリー性などが知られている。

スケールフリー性とは、ごく少数のノードが膨大な次数(エッジ)を持つ一方、大多数のノードはごく少数の次数を持つという特徴である。

またノードがネットワーク内でどこに位置するかを表す指標として中心性がある。次数中心性、近接中心性、媒介中心性の3つ中心性が知られている。

次数中心性とは「線が集まる点ほど中心性が高い」とする最もシンプルな中心性の計算法である。点の次数がそのまま中心性となる。次数中心性が高いほどハブに近い役割を果たしている頂点であるといえる。頂点  $v_i$ 、頂点  $v_i$  と隣接する頂点の次数  $\text{deg}(v_i)$ 、ネットワーク内の総頂点数  $n$  の時、次数中心性  $C_d$  は以下のように表される。

$$C_d(v_i) = \frac{\text{deg}(v_i)}{n-1}$$

近接中心性とは「他の点と距離が近いほど中心性が高い」とする中心性の計算法である。近接中心性が高いほど、他者からの影響を受けにくい頂点になる。頂点  $v_i$ 、他の頂点までの距離の総和  $s(v_i)$  とすると、近接性  $C_c$  は以下のように表される

$$C_c(v_i) = \frac{n-1}{s(v_i)}$$

媒介中心性とは「その点を通る経路が多いほど、中心性が高い」とする計算法である。媒介中心性が高いほどネットワーク内での情報や資源の流れに頻繁に関与している頂点になる。媒介中心性  $C_b(v_k)$  は、頂点の媒介値が理論的に最大となる場合(スターグラフ)の媒介値  $BC(v_k)$  の比で表現される

$$C_b(v_k) = \frac{2BC(v_k)}{(n-1)(n-2)}$$

### 2.3. パターン利用時の問題点

パターン利用者がパターンを適用する場合、特定領域の問題を解決するパターンのみを適用することは少なく、より全体的な問題を解決する汎用性のあるパタ

ーンを組み合わせることで適用することが多い。

ある問題を解決するパターンを選択したとき、そのパターンと参照関係にある汎用性の高いパターンと組み合わせる必要があるかその妥当性をパターン利用者は検討しなければならない。そのため、選択したパターンがパターン間の参照関係のどこに位置するのかを検討する必要がある。しかし、パターン間の参照関係の位置を把握する指標というのはあまりわかっていない。

パターン抽出時においては、パターンの再利用性を高めることが重要である。ソフトウェア開発時パターンを単独で適用することが一般的ではない。そのため、パターン利用者が抽出したパターンと参照関係にあるパターンも抽出しておく必要がある。

参照パターンがパターン間の参照関係の末端付近にあるパターンの場合、特定の問題領域を解決するパターンの可能性がある。よって、抽出したパターンが解決する問題と参照パターンが解決する問題がずれていないか妥当性を検討する必要がある。そのために、参照パターンについて、パターン間の参照関係の位置を把握する必要があるが、そのための参考となる指標はあまりよくわかっていない。

そこで本稿では、各パターンがパターン間の参照関係のネットワーク内のどこに位置するのか構造的な特徴について明らかにすることにした。また、得られた構造的な特徴がパターン利用者にとって有用かどうか検討することにした。

### 3. Portland Pattern Repository の分析

パターン間の参照関係の構造的な特徴について明らかにするために、次のような分析を行った (図 1)。

- Step1 web クローラーwebSPHINX を用いたツールを作成し、PPR に登録されているパターン間の参照関係と履歴情報 (パターンのページの更新回数及び最終更新日) を各ページのリンクをクロウリングすることで収集した。
- Step2 その参照関係に対して、ネットワーク分析ツール Pajeck 使用して、各パターンの中心性を算出した。
- Step3 中心性と履歴情報のメトリクス間の関連を、統計ソフト R を使用して分析した。

Step1 において、PPR に登録されているパターンは次のように抽出した。PPR の「Pattern Index」および「Category Pattern」のページに記載されているパターン名を今回の調査対象のソフトウェアパターンとして抽出した。

ただし、下記のパターン名は対象から除外した。

- 明らかにパターンでないと思われる文字列を含んでいるパターン (GuideLine, Discussion, Patterns

etc)

- Not a Pattern カテゴリーうち、明らかにパターンでないパターン (RoleModels, OakTreeDiagram, SpeculationInPatternFormat, SuccessStory)
- 大きな開発プロセスに関するパターン (TestDrivenDevelopment, TestFirst etc)

また、最終更新日は、2013 年 2 月 6 日を基準とし、Week へ単位を変換した。

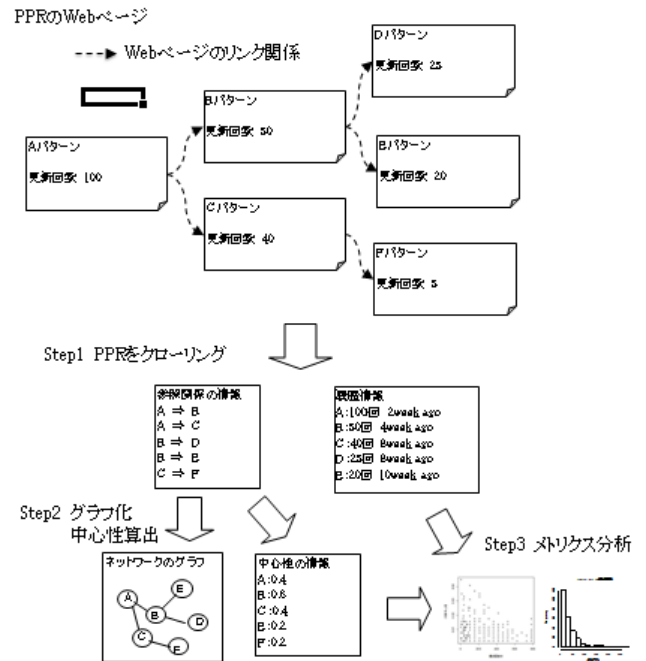


図 1 分析手順

#### 3.1. 分析内容

(1) メトリクス単独での分析

算出した 3 つの中心性、更新回数、最終更新日に対して、トリクス単独でヒストグラムを作成し、パターン間の参照関係図をグラフ化した。

(2) 各パターンカテゴリーのメトリクスの分類

各パターンの所属するカテゴリーごとにメトリクスの平均値を計算し、パターンのカテゴリー間で特徴がみられるかについて箱髭図を作成し、分析した。また、パターン間の参照関係図において、特徴のあるカテゴリーを色分けして表示し、分布状況を分析した。

(3) メトリクス間の関連

中心性と更新回数、中心性と最終更新日のメトリクス間の関連を分析した。

#### 3.2. 結果と考察

(1) メトリクス単独での分析

各パターンの中心性、更新回数、最終更新日のヒストグラムを図 2 から図 6 に示す。また、ノードのサイズを大きくした度数中心性の高い上位 10 パターンを表 1 に示した。

各パターンの参照関係を図7に示す. 各ノードがパターンを表しており, パターン間の参照関係を線で表示している. 度数中心性の高い上位10パターンのノードのサイズを大きくしてグラフ化した.

図4から媒介中心性に関しては, スケールフリー性がみられ, ごく一部の限られたパターンがパターン間の関係を媒介しているという特徴がみられた.

表1から ModelViewController, AdapterPattern, StrategyPattern といった, 実際によく利用されるとおもわれるパターンは度数中心性が高い傾向があるということがわかった.

度数中心性の定義から度数中心性の高いパターンはネットワーク内で中心部の近くに位置するはずである. 図7から実際に度数中心性の高いパターンがネットワークの中心部の近くに位置していることが確認できた.

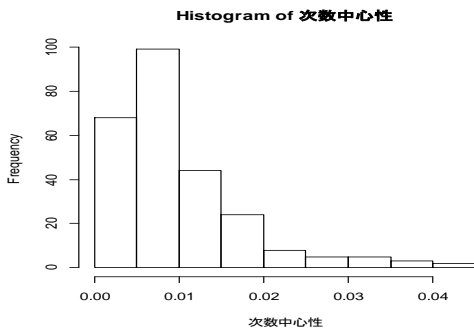


図2 ヒストグラム 度数中心性

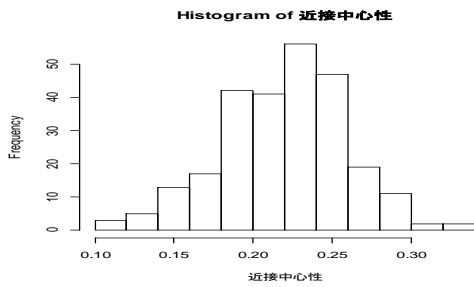


図3 ヒストグラム 近接中心性

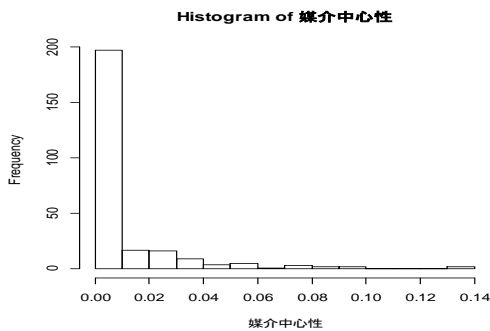


図4 ヒストグラム 媒介中心性

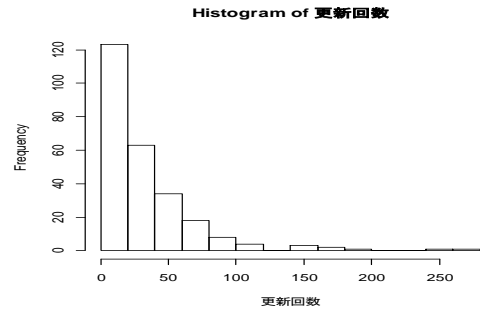


図5 ヒストグラム 更新回数

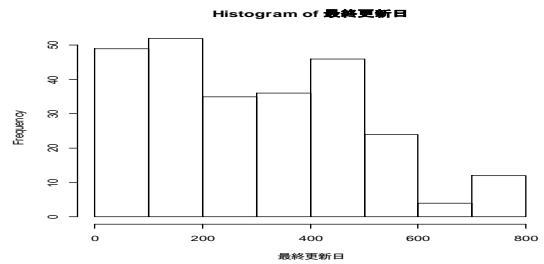


図6 ヒストグラム 最終更新日

表1 度数中心性の高い10パターン

パターン名	度数中心性
ModelViewController	0.0447
AdapterPattern	0.0408
HandleBodyPattern	0.0369
VisitorPattern	0.0350
SynchronizationStrategies	0.0350
ValueObject	0.0330
SceneGraph	0.0330
StrategyPattern	0.0311
CompositePattern	0.0311
ScapeGoat	0.0311

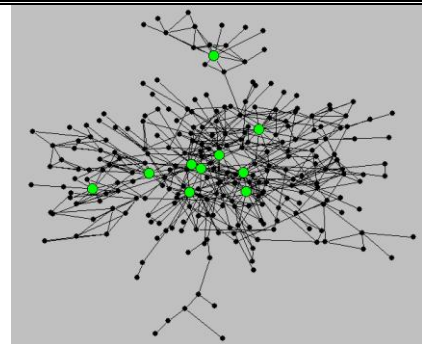


図7 パターン間の参照関係

(2) 各パターンカテゴリーのメトリクスの分類  
 パターンカテゴリーごとの度数中心性, 更新回数, 最終更新日の平均値を算出し, その箱髭図を図7から図9に示す. またカテゴリーごとのカテゴリー名, 度数中心性の平均値, 所属するパターン数を表2に示す. カテゴリーによって, 所属するパターン数, 度数中心性, 更新回数, 最終更新日にはばらつきがあることが

わかった。

次数中心性の平均値が高かったカテゴリ Functional Pattern System For Object Oriented Design と次数中心性の高いパターンを含む Software Design Patterns のカテゴリのパターンを大きくし参照関係をそれぞれ図 10 と図 11 にグラフ化した。また、次数中心性の低いカテゴリ Testing Patterns について大きくし、参照関係を図 12 にグラフ化した。

次数中心性の定義は、ノードがネットワーク内で中心に近いかどうかを示す指標であって、カテゴリ内でパターン同士が参照関係しているかという指標ではない。しかし、図 10-12 からカテゴリ内のパターン同士がネットワークにおいて近くに位置し、参照していることが確認できた。よって同一カテゴリ内のパターンはネットワーク内において比較的近くに位置しており、カテゴリ内のパターンと参照関係を持つ傾向があることが分かった。

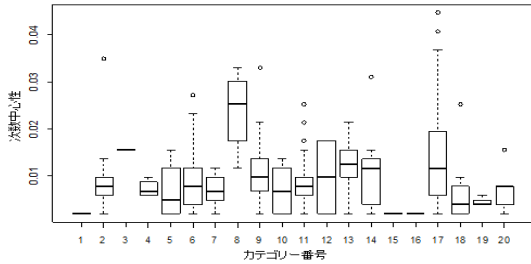


図 7 箱髭図 次数中心性とカテゴリ番号

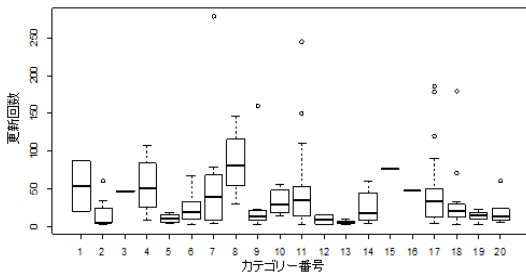


図 8 箱髭図 更新回数とカテゴリ番号

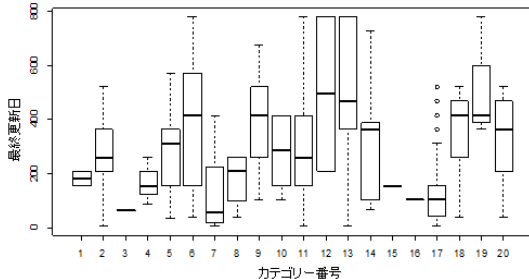


図 9 箱髭図 最終更新日とカテゴリ番号

### (3) メトリクス間の関連

次数中心性と更新回数の散布図を図 13 に示す。また、次数中心性と更新回数の関係について回帰分析を行った。その結果、次数中心性と更新回数の相関係数の値は 0.328 となり、あまり相関性は見られなかった。

更新回数の多いパターンは、 ModelViewController, MockObject, SingletonPattern などよく利用されると思われるパターンだった。

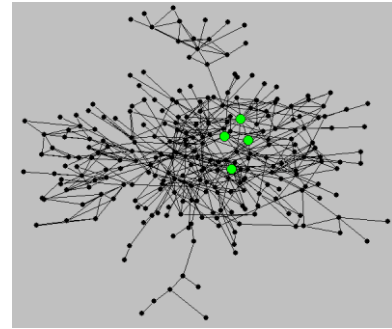


図 10 パターン参照関係 カテゴリ8

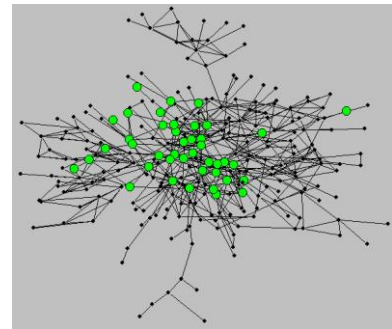


図 11 パターン参照関係 カテゴリ17

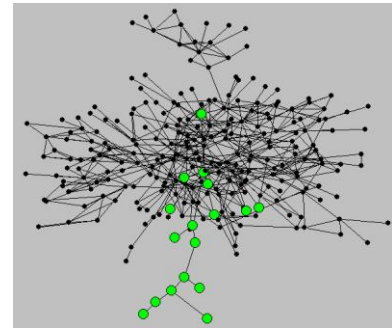


図 12 パターン参照関係 カテゴリ18

次に次数中心性と最終更新日の関連を示す散布図を図 14 に示す。次数中心性と最終更新日に関して、次数中心性 0.2 を境界として 2 グループに分け、最終更新日の平均値に差異があるか t 検定を行ったところ、P 値は 2.861e-07 となり、有意に差異があることが分かった。登録したパターンから参照する他パターンの登録数が増えているため、更新タイミングが最近であるほど、次数中心性が高いパターンを作成できることが理由として考えられる。

## 4. おわりに

本稿では PPR 上に登録されているパターン間の参照関係を洗い出し、データ分析を行った。また、算出したパターンのメトリクスの分析を行った。得られた知見を以下にまとめる。

表 2 カテゴリー名一覧

No	カテゴリー名	平均度数 中心性	所属パタ ーン数
1	カテゴリーなし	0.002	2
2	Category Concurrency Patterns	0.010	9
3	Category Creational Patterns	0.016	1
4	Category Security Patterns	0.007	4
5	CategoryStructuralPatterns	0.007	6
6	Component Design Patterns	0.009	25
7	Derivations & extensions to MVC	0.007	8
8	Functional Pattern System For Object Oriented Design	0.024	4
9	Graphics Patterns	0.011	23
10	Individual patterns	0.007	6
11	Java Idioms	0.008	69
12	Not a pattern	0.010	2
13	Object Based Programming	0.012	10
14	Organizational Patterns	0.010	15
15	Patterns For Effective Meetings	0.002	1
16	Process Anti Patterns	0.002	1
17	Software Design Patterns	0.015	42
18	Testing Patterns	0.006	17
19	User Interface Patterns	0.005	3
20	Website Patterns	0.008	10

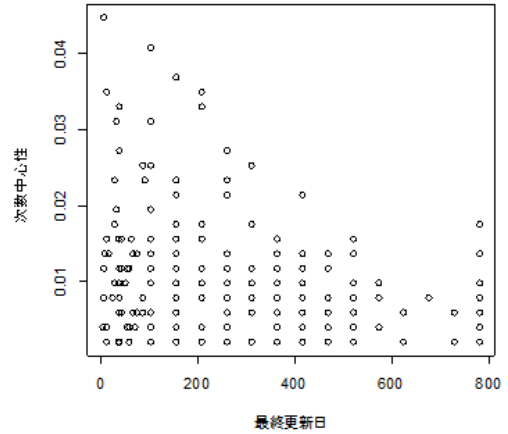


図 14 度数中心性と最終更新日

パターン選択時において、度数中心性の高いパターンは、パターンの参照関係の中心付近に存在する傾向があり、汎用性が高いと思われるため、度数中心性の高いパターンから適用を検討したほうが効率がよいと考えられる。また適用するパターンの度数が非常に小さい場合、特定の領域の問題に適用するパターンである可能性が高いため、パターンが扱う問題と直面している問題がずれていないか妥当性を検討すべきである。

パターン抽出時においては、抽出したパターンの参照パターンの度数中心性が低い場合、参照パターンが参照関係の中心からは外れており、特定領域の問題を解決するパターンである可能性が高い。よって、抽出したパターンとの参照関係の妥当性を検討する対象になると考えられる。

今後の展望として、OSSのコミットログやパターン間の参照関係に基づく重要度[5]などを利用し、各パターンのソフトウェアへの適用回数、再利用回数、重要度などのメトリクスが得られれば、度数中心性の高いパターンがよく利用されるのかといった裏付けをとることが可能になると考えられる。

## 文 献

- [1] <http://c2.com/cgi/wiki?PortlandPatternRepository>, Portland Pattern Repository, 2013年2月6日
- [2] 大平雅雄, ソーシャルネットワーク分析の基礎, [http://se.naist.jp/~masao/lecture2008/sna/socialnetwork\\_analysis\\_1.pdf](http://se.naist.jp/~masao/lecture2008/sna/socialnetwork_analysis_1.pdf), 2013年1月8日
- [3] 井庭崇, 湯村 洋平, 若松 孝次, 古市 奏文 2007年, プロジェクト推進のパターン・ランゲージとその評価, 日本ソフトウェア科学会 ネットワークが創発する知能研究会&情報処理学会 数理モデル化と問題解決研究会合同ワークショップ
- [4] Kubo, A., Nakayama, H., Washizaki, H., Atsuhiko, T and Fukazawa, Y.: ソフトウェア設計パターンの抽象度測定方法, FOSE2006
- [5] Kubo, A., Nakayama, H., Washizaki, H. and Fukazawa, Y.: PatternRank: A Software-Pattern Search System Based on Mutual Reference Importance, 15th Pattern Languages of Programming (PLOP2008), Nashville (2008).

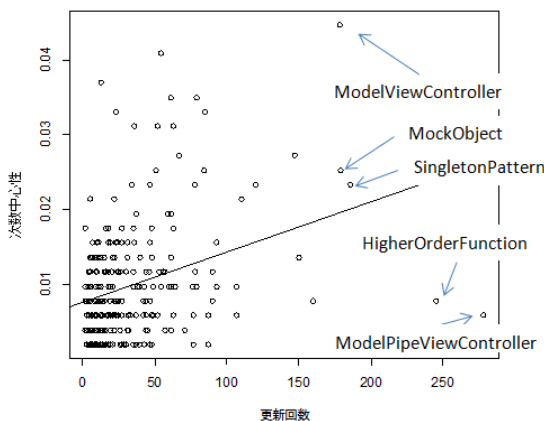


図 13 度数中心性と更新回数