Portland Pattern Repository 上のソフトウェアパターン群に対するネットワーク分析

Network Analysis for Software Patterns in Portland Pattern Repository

角谷　将司[†]　　鷲崎　弘宜[†]　　川村　健[‡]　深澤　良彰[†]

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示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となった．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．

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様々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の問題に対し，それを解決できるパターンを選択し適用する．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．PPR では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．これらの要素間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．

ソフトウェア開発者は，単独のパターンだけでなく，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，パターンが末端付近にある場合，より抽象的なパターンの適用も検討する必要がある．

したがって，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．

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．

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を

† 早稲田大学基幹理工学部情報理工学科
‡ USOL 東京株式会社

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，既に公開済みの内容を随時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し，追記・修正行っている．PPR には 1 パターンの情報を 1 ページごとに登録されており，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登録されている．

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，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．

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，線で頂点と頂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，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．

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．

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．

以下に次数中心性 C_d，近接中心性 C_c，媒介中心性 $C_b(v_k)$の式を示す．

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

頂点 v_i，頂点 v_i と隣接する頂点の次数 $\deg(v_i)$，ネットワーク内の総頂点数 n

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

頂点 v_i，他の頂点までの距離の総和 $s(v_i)$

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

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 $BC(v_k)$

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，そのパターンと参照関係にある汎用性のあるパターンを組み合わせて適用することが多い．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．

また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，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，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，得られた構造的特徴がパターン利用者にとって有用かどうか検討した．

3 Portland Pattern Repository の分析

パターン間の参照関係の構造的特徴について明らかにするため，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をリンクのクロー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．

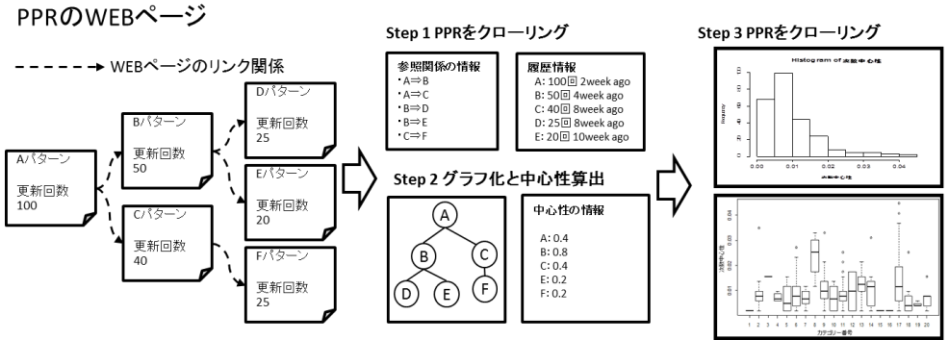

図 1　分析手順

3.1 分析結果と考察

（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．それらのヒストグラムを図 2 から図 6 に示す．また，次数中心性の高い上位 8 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．次数中心性の高い上位 8 パターンをグラフ化した．

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次数中心性が高い傾向があるということがわかった．

図 7 から実際に次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.

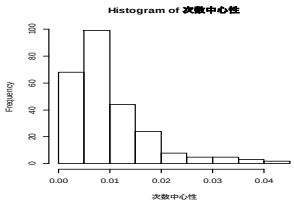

図 2　次数中心性

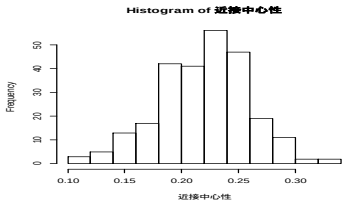

図 3　近接中心性

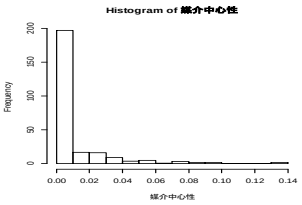

図 4　媒介中心性

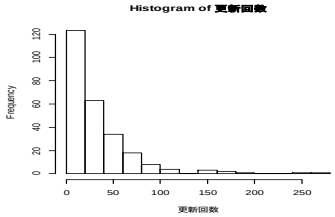

図 5　更新回数

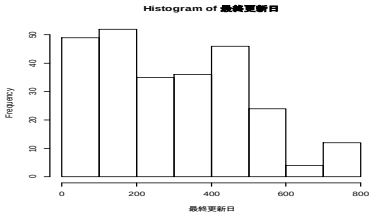

図 6　最終更新日

表 1　次数中心性の高い 8 パターン

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

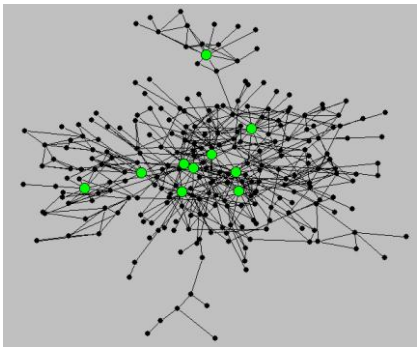

図 7　パターン間の参照関係

（2）各パターンカテゴリーのメトリクスの分類

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，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．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，次数中心性の平均値，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，次数中心性，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ー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 のカテゴリーの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ー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，カテゴリー内でパターン同士が参照関係になっているかという指標ではない．し

かし，図 10-12 からカテゴリー内のパターン同士が近くに位置し，参照していることが確認できた．よって同一カテゴリー内のパターンは比較的近くに位置しており，カテゴリー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．

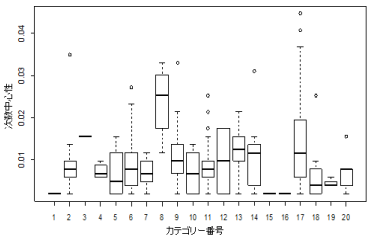

図 7　次数中心性とカテゴリー番号

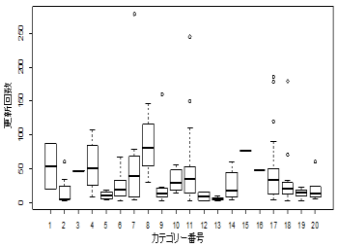

図 8　更新回数とカテゴリー番号

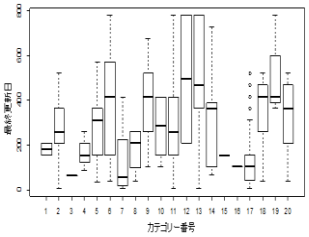

図 9　最終更新日とカテゴリー番号

（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．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であった．

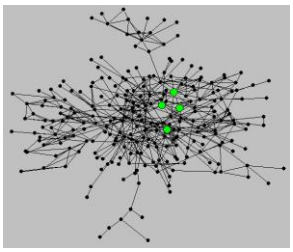

図 10　カテゴリー8

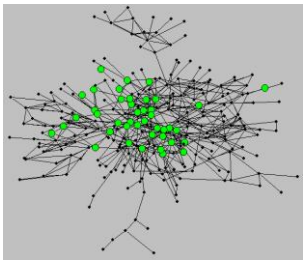

図 11 カテゴリー17

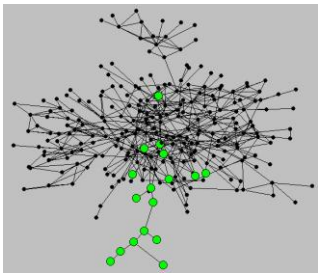

図 12　カテゴリー18

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，算出したパターンのデータ分析とメトリクス分析を行った．得られた知見を以下にまとめる．

パターン選択時において，次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，次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次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次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，次数中心性が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．

表 2　カテゴリー名一覧

No	グループ名	パターン数
1	カテゴリーなし	2
2	Category Concurrency Patterns	9
3	Category Creational Patterns	1
4	Category Security Patterns	4
5	CategoryStructuralPatterns	6
6	Component Design Patterns	25
7	Derivations & extensions to MVC	8
8	Functional Pattern System For Object Oriented Design	4
9	Graphics Patterns	23
10	Individual patterns	6
11	Java Idioms	69
12	Not a pattern	2
13	Object Based Programming	10
14	Organizational Patterns	15
15	Patterns For Effective Meetings	1

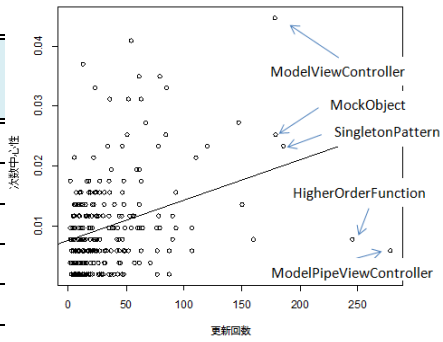

図 13　次数中心性と更新回数

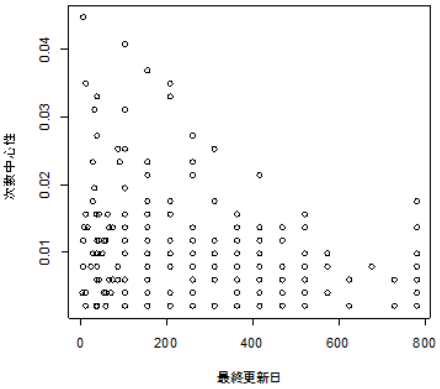

図 14　次数中心性と最終更新日

5　参考文献

[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_I.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.,Atsuhiro,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).