Analyzing Software Patterns Network obtained from Portland Pattern Repository

Hironori Washizaki, Suthinan Thanintranon,
Masashi Kadoya and Yoshiaki Fukazawa
Waseda University
3-4-1 Okubo, Shinjuku-ku, Tokyo, 169-8555 Japan
washizaki@waseda.jp
suthinan_mart@fuji.waseda.jp
kage_purinman@ruri.waseda.jp
fukazawa@waseda.jp

Takeshi Kawamura
USOL Tokyo Co., Ltd.
takeshi.kawamura@usol.co.jp

Joseph W. Yoder
The Refactory, Inc.
joe@refactory.com

***Abstract*— Software patterns are general reusable solutions to commonly occurring problems within a given context. Patterns usually form a network of relationships that support how to understand and utilize the patterns efficiently and effectively. However little is known about the nature of such pattern networks, such as the centrality of a pattern and its meaning. To clarify such characteristics of software patterns, we mine a network consisting 283 patterns from the current world-largest online pattern repository called Portland Pattern Repository (PPR). By applying network analysis techniques to the mined network and careful review of the result, we revealed several interesting characteristics of the pattern network and patterns in PPR, such as that the degree centrality could be useful to support developers and to more easily understand whether patterns under consideration are core patterns or peripheral ones in the entire pattern network. The "betweeness centrality" seems to be useful to support ways to identify those patterns that play a role of hub and grasp relationships among different pattern groups.** [1] **Moreover we believe these findings could contribute to further researches on pattern networks.**

Keywords—software patterns; network analysis; pattern repository; pattern languages

I. Introduction

A Software pattern is a general reusable solution to a commonly occurring problem within a given context, specifically for software development. Most patterns have relations to other patterns [2][20-21], such as inside usage (X uses Y in its solution), combinational usage (X can be combined with Y), similar patterns (X is similar to Y), and alternative patterns (X is an alternative to Y) [3]. Such relations are usually described in a "Related Patterns" section or other related sections of each pattern document. As a result, patterns usually form a network having relations among them to support users understand and utilize patterns efficiently and effectively.

However little is known about the nature of entire or partial pattern network, such as how are patterns connected with many patterns relatively important from the viewpoint of frequent applications. Knowing such characteristics could be beneficial for understanding, reusing and extending existing patterns and writing new ones.

To clarify such characteristics of pattern networks and organizational patterns, we mine a network consisting of 283 patterns (March 2013) from the online pattern repository called Portland Pattern Repository (PPR) [4]. We choose PPR as the target repository because, to the best of our knowledge, PPR is the oldest, largest and still active pattern repository so that its analysis should be beneficial to discover common, major and current characteristics in pattern networks containing various patterns. PPR is an origin of Wiki and being actively updated; each Wiki page in PPR describes a pattern or a document related to patterns. For example a page "Pattern Index" is an index for many patterns and was last edited on April 2013.

It should be noted that anyone can contribute to editing patterns and related documents in PPR. Because of this there are some peculiarities such as existence of non-pattern pages, the lack of a common format and no guarantee that all relationships have been captured. The first peculiarity has been solved by manual filtering described in the next section; however other two are not currently addressed so that these might affect the analysis result such as missing important relationships. In the future we will attempt to analyze other

[1] Preliminary result focusing on organizational patterns is presented at [1]. In this paper we show and discuss in detail about various pattern groups and their comparison.

smaller, somewhat old but more refined repositories and catalogs such as the Pattern Almanac [16] and well-accepted books on patterns (such as [13]), and compare the results with the findings reported in this paper.

By applying network analysis techniques to the mined network and careful review of the result, we revealed several interesting characteristics of the pattern network and several patterns such as organizational patterns. For example, the degree centrality seems to reflect the commonness and generality of the corresponding pattern.

The remainder of this paper is organized as follows. First, we describe our analysis procedure in Section II. We then describe and discuss obtained results and findings in Section III. Finally, we conclude our work in Section IV.

II. Analysis Procedure

We analyzed the pattern network in PPR using the following steps.

(Step1) We collected 483 pattern names, incoming and outgoing relations, and belonging groups by crawling the PPR Wiki pages by using a Web crawler based on WebSPHINX [5]. We systematically dealt with any link described in Wiki pages as relations. Most of those links are about inside usages, combinational usages and similar patterns. PPR contains several non-pattern documents so that we regarded pages that are linked from the "Pattern Index" page and the "Category Pattern" page as patterns. We eliminated pages containing two or more patterns, in order to ensure that each page contains just one pattern. Moreover, we identified groups where patterns belong by checking links manually from pattern pages to category pages. Some pattern pages have links to multiple groups; in that case we choose one group manually by considering the major property of target patterns.

(Step2) We manually filtered out several non-pattern documents such as the pages whose names contain "guideline" and "discussion". Finally we obtained 283 patterns and 19 groups; each pattern belongs to just one group.

(Step3) We measured three major types of centrality (degree, closeness and "betweeness" defined below) [6] for each pattern by using a network analysis tool Pajek [7]. When measuring centrality, to make analysis simple, we did not distinguish the direction of relations; measuring centrality for each direction (i.e. incoming or outgoing) could be future work.

- The normalized degree centrality[2] (hereafter "degree centrality") C_d of a pattern *i* is defined by the following formula, where v_i is a node corresponding to the pattern *i*, $\deg(v_i)$ is the number of nodes connected to the node v_i, and *n* is the total number of the given network. The degree centrality can be seen as the simplest centrality. We assume that the degree centrality of each pattern quantifies the pattern's importance and the commonness of the context of the pattern in the belonging entire pattern network. If two patterns belong to a meaningful relationship such as the internal usage, combinatorial usage or similar patterns, their contexts should have some commonality; it results in a link between corresponding pattern pages if page authors are aware of the relationship.

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

- The closeness centrality C_c is defined by the following formula, where $s(v_i)$ is the sum of v_i's distances to all other nodes. We assume that the closeness centrality of each pattern reflects the pattern's importance within its corresponding pattern group since the closeness centrality in general tends to give high scores to nodes that are near the center of local clusters in an overall network [17].

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

- The betweeness centrality C_b is defined by the following formula, where $BC(v_i)$ is the betweenness of $v_{i,}$ which reflects the number of shortest paths from all nodes to all others that pass through v_i. We assume that the betweeness centrality of each pattern quantifies the extent to which the pattern plays a role of hub connecting different pattern groups.

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

III. Analysis Results

According to the above-mentioned procedure, we obtained (1) centrality of pattern. Moreover by focusing one pattern group, organizational patterns, we discuss (2) implications of centrality of patterns.

(1) Centrality of patterns in main network

Fig 1 illustrates entire pattern network. Interestingly, there are two independent pattern networks: main patterns network and organizational patterns one. 14 organizational patterns together with two non-organizational patterns form an independent network from other 267 patterns such as design patterns and architecture patterns. It is because that most of 283 patterns are about software product and have no relationship with organizational patterns.

Fig 2, 3 and 4 show the histogram of degree, closeness and betweeness centrality of 267 patterns in the main network, respectively.

In Fig 2, many patterns have a small degree centrality; which means that many patterns refer to a small number of patterns through a Wiki page link. However the distribution does not follow the well-known network property "power law distribution" so we cannot state that the pattern network obtained from PPR is a scale-free network. Though previous Wikipedia research reported that various quantities including

[2] Instead of the degree centrality, we use the normalized degree centrality so that future work can compare different pattern repositories and catalogs.

the in-degree of links from other Wiki pages are distributed according to the power law [18]. The reason why the pattern network in PPR does not follow the power law distribution might be because of the smallness of the size or some nature of its evolution process; in the future we will investigate larger pattern networks and their evolution process in PPR.

In Fig 3, the distribution of patterns regarding closeness centrality seems to follow the normal distribution. In Fig 4, most of patterns have very low betweeness centrality; which means that in the pattern network, few patterns seem to play a role of hub that connects different pattern groups. However there might be a possibility that the evolution process of the pattern network affects the betweeness centrality; for example patterns that were written earlier in time may not refer to patterns that were written later, although it is always possible to update every Wiki page. Indeed, we confirmed that the last update dates of the target 283 pattern pages vary from 3 weeks ago to 16 years ago. In the future we will investigate their evolution process and its impact on the betweeness centrality.

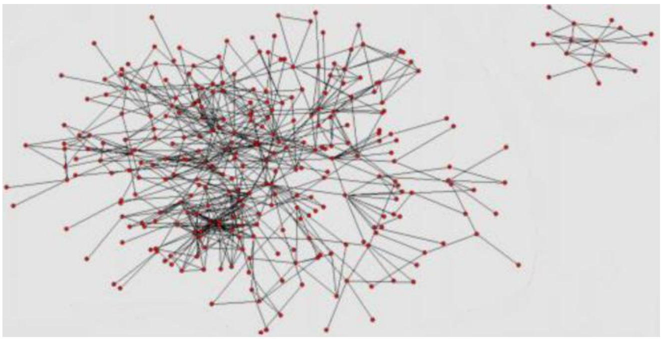


Fig. 1. Two pattern networks in PPR; main network at left hand side and oranizational patterns network at right hand side (each node indicates a pattern; each link indicates a relation between two patterns.)

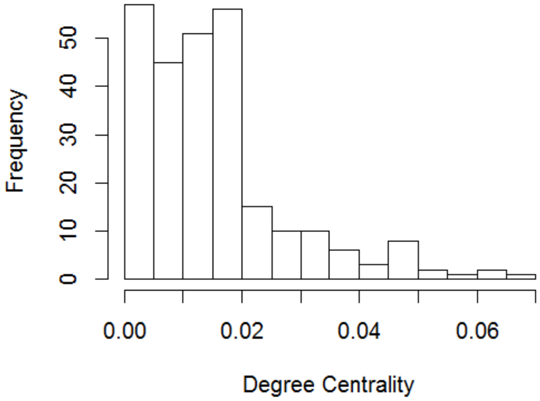


Fig. 2. Histogram of degree centrality of patterns in the main network

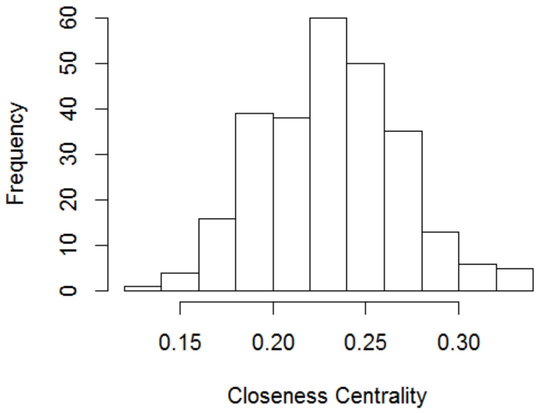


Fig. 3. Histogram of closeness centrality of patterns in the main network

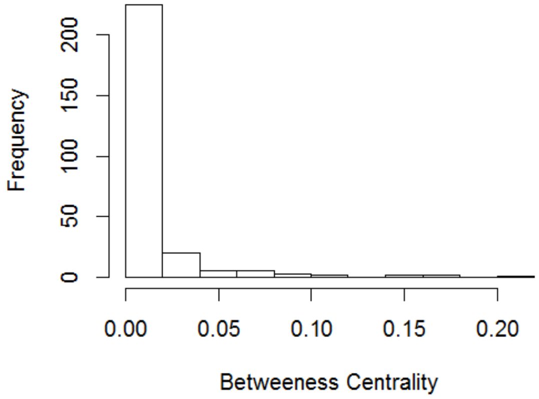


Fig. 4. Histogram of betweeness centrality of patterns in the main network

TABLE I. MEASUREMENT RESULTS OF TOP-10 PATTERNS

Pattern name	N. patterns referred by the pattern	N. patterns referring to the pattern	Degree centrality	Closeness centrality	Betweenes s centrality
ModelViewController	12	12	0.065789	0.329208	0.202666
AdapterPattern	6	16	0.06391	0.304	0.08354
ValueObject	4	15	0.06391	0.27622	0.055064
CompositePattern	4	13	0.056391	0.334591	0.172935
StrategyPattern	5	11	0.052632	0.336283	0.111931
NullObject	4	12	0.050752	0.327183	0.141363
MockObject	5	10	0.048872	0.306805	0.161318
SceneGraph	6	11	0.048872	0.284188	0.066935
TestDrivenDevelopment	4	10	0.048872	0.28178	0.148753
VisitorPattern	7	11	0.046992	0.339286	0.107648

(2) Implications of centrality of patterns in organizational patterns network

We try to figure out implications of centrality of patterns by focusing on another pattern network, i.e. organizational patterns network. There are 15 organizational patterns including positive patterns and anti-patterns. We chose that group because organizational patterns have a long history and are known as the basis for Agile software development movement [8-15], which is now widely accepted.

Fig 5 shows how most of relationships are closed within the group. Among them, Fig 6 shows details of selected four patterns having high centrality.

Table II shows the number of related patterns (i.e. patterns referred by the pattern, and the number of patterns referring to the pattern), degree, closeness and betweeness centrality of each organizational pattern. Moreover, Table III shows information on non-organizational patterns referring to organizational patterns; there are two patterns referring to "Scape Goat": "Six Thinking Hat" in Patterns For Effective Meetings group and "Discordant Reward Mechanism" in Process Anti Patterns.

In Table II, many organizational patterns have several related patterns. Especially, there are two patterns having high degree centrality (i.e. having many related patterns) : "Scape Goat" and "Train Hard Fight Easy". "Scape Goat", as an anti-pattern, gives other normal organizational patterns a common problematic context so that it is connected to other organizational patterns including the above-mentioned two patterns. In contrast, organizational patterns having low degree centrality such as "Slow Poison" are specific to their own contexts. It means that the degree centrality seems to somewhat reflect the commonness and generality of the corresponding pattern. In other word, the degree centrality could be useful to support developers to easily understand whether patterns under consideration are core patterns or peripheral patterns in the entire pattern network.

Regarding the betweeness centrality, "Scape Goat" seems to play the roles of hub. Actually they connect organizational patterns to other groups: Process Anti Patterns and Patterns for Effective Meetings.

Regarding the closeness centrality, "Scape Goat" has the highest centrality. It might be seen as a relatively important pattern within the group according to our assumption described in section I.

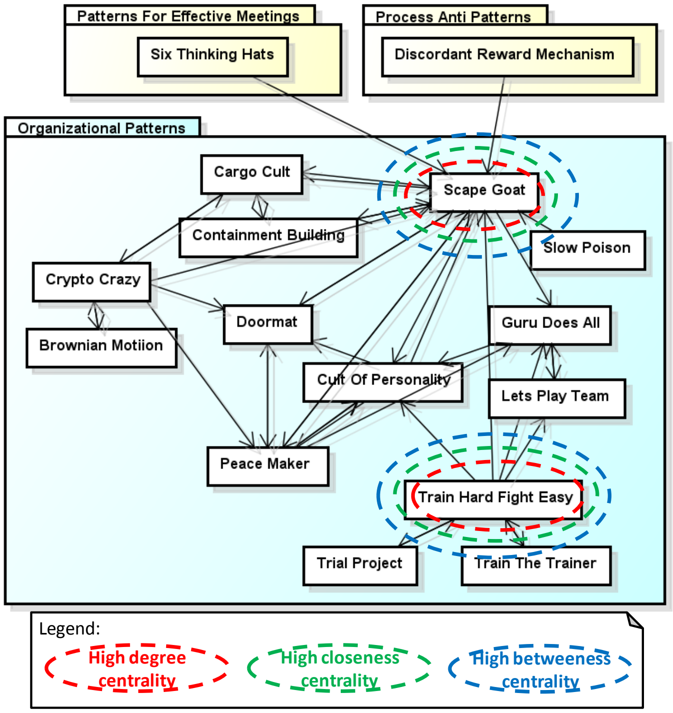
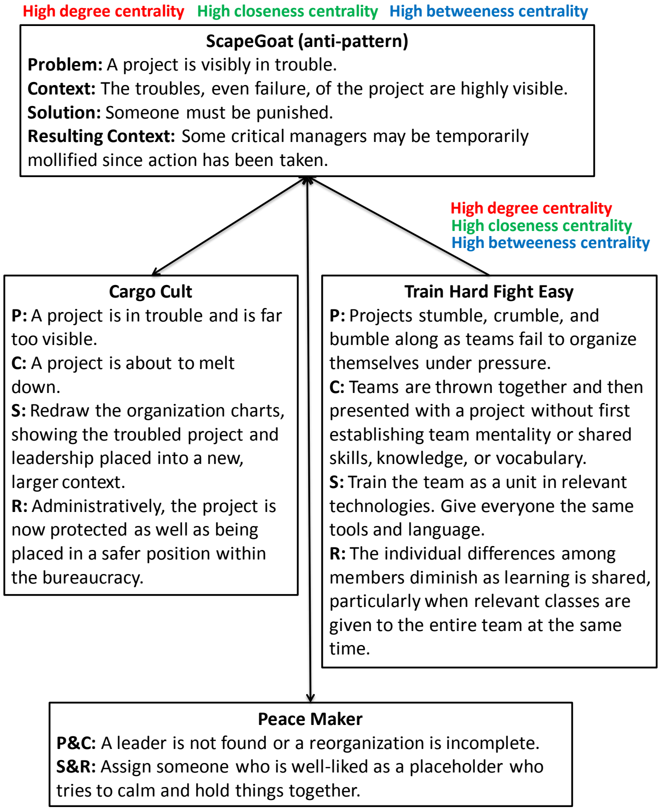


Fig. 5. Relationships among organizational patterns and related patterns belonging to other groups in UML class diagram (each class box indicates a pattern; each package box indicates a pattern group; due to space limitation not all of patterns are shown for non-organizatoinal pattern groups.)


High degree centrality
High closeness centrality
High betweeness centrality
High degree centrality
High closeness centrality
High betweeness centrality

ScapeGoat (anti-pattern)
Problem: A project is visibly in trouble.
Context: The troubles, even failure, of the project are highly visible.
Solution: Someone must be punished.
Resulting Context: Some critical managers may be temporarily mollified since action has been taken.

Cargo Cult
P: A project is in trouble and is far too visible.
C: A project is about to melt down.
S: Redraw the organization charts, showing the troubled project and leadership placed into a new, larger context.
R: Administratively, the project is now protected as well as being placed in a safer position within the bureaucracy.

Train Hard Fight Easy
P: Projects stumble, crumble, and bumble along as teams fail to organize themselves under pressure.
C: Teams are thrown together and then presented with a project without first establishing team mentality or shared skills, knowledge, or vocabulary.
S: Train the team as a unit in relevant technologies. Give everyone the same tools and language.
R: The individual differences among members diminish as learning is shared, particularly when relevant classes are given to the entire team at the same time.

Peace Maker
P&C: A leader is not found or a reorganization is incomplete.
S&R: Assign someone who is well-liked as a placeholder who tries to calm and hold things together.

Fig. 6. Partial relationships and details of selected patterns having high centrality ('P', 'C', 'S' and 'R' denote Problem, Context, Solution and Resulting Context, respectively.)

TABLE II. MEASUREMENT RESULTS OF ORGANIZATIONAL PATTERNS

Pattern name	N. patterns referred by the pattern	N. patterns referring to the pattern	Degree centrality	Closeness centrality	Betweenees s centrality
ScapeGoat	6	10	0.733333	0.789474	0.640317
TrainHardFightEasy	6	2	0.4	0.6	0.300635
CryptoCracy	5	1	0.333333	0.535714	0.142857
CultOfPersonality	3	4	0.333333	0.576923	0.030794
GuruDoesAll	2	4	0.333333	0.576923	0.070794
PeaceMaker	4	4	0.333333	0.555556	0.027302
DoorMat	2	4	0.266667	0.517241	0.006349
CargoCult	3	3	0.2	0.5	0.009524
BrownianMotion	1	2	0.133333	0.357143	0
ContainmentBuilding	2	2	0.133333	0.46875	0
LetsPlayTeam	1	2	0.133333	0.405405	0
SlowPoison	1	0	0.066667	0.454545	0
TrainTheTrainer	1	1	0.066667	0.384615	0
TrialProject	1	1	0.066667	0.384615	0

TABLE III. MEASUREMENT RESULTS OF NON-ORGANIZATIONAL PATTERNS REFFERRING TO ORGANIZATIONAL PATTERNS

Pattern name	N. patterns referred by the pattern	N. patterns referring to the pattern	Degree centrality	Closeness centrality	Betweenees s centrality
DiscordantRewardMechanisms	1	0	0.066667	0.454545	0
SixThinkingHats	1	0	0.066667	0.454545	0

IV. CONCLUSION AND FUTURE WORK

In summary, we mined the PPR patterns network consisting of 283 patterns including 14 organizational patterns, analyzed the pattern network, and found the following interesting characteristics of the pattern network.

- Although the pattern network cannot be seen as a scale-free network, it is still true that many patterns refer to a small number of patterns. In the pattern network, few patterns play a role of hub that connects different pattern groups.
- There are two independent pattern networks: main patterns network and organizational patterns one. Most of 283 patterns are about software product and have no relationship with organizational patterns.
- The degree centrality seems to somewhat reflect the commonness and generality of the corresponding pattern. The degree centrality could be useful to support developers to more easily understand whether patterns under consideration are core patterns or peripheral ones in the entire pattern network.
- The betweeness centrality seems to be useful to support users identify patterns playing a role of hub and grasp relationships among different pattern groups.
- The closeness centrality might reflect the pattern's importance within its corresponding group; this assumption needs further investigation.

We believe that these findings could contribute to the software development community for understanding, reusing and extending existing patterns and for writing new patterns. For example, developers or managers who want to form agile teams could consider reusing those organization patterns in PPR starting by referring to ones that have a high degree centrality such as "Scape Goat". Or, if developers and managers want to grasp relationships among organizational patterns and other groups' patterns, "Scape Goat" should be considered first. In the future we plan to develop a pattern-browser that shows preferred sequences for learning about a system of patterns based on these findings. Moreover we believe these findings could contribute to further research on pattern networks.

In this analysis, we regarded Wiki page links as pattern relationships; however this is often based on page authors' awareness of other patterns so they might be different from pattern authors' intentions for the actual relationships. We will handle this threat to validity by referring to relationships specified in original pattern documents. Moreover, we did not deal with the meaning (i.e. type) of relationships such as the internal usage and combinational usage; we will try to discriminate these in future work as well. Other relationships of patterns could also be considered, such as commonality and relationships among authors, and actual combinational usages in actual software and organizations; analysis on those multiple and overlay networks in patterns could discover characteristics of patterns from the viewpoint of relationships in detail. It might lead to detecting various community structures in the network [19] .

In the future, we will investigate how are these findings related to actual agile or non-agile software development adapting organizational patterns and product ones. Such investigation will include further analysis for the impact of directions and types (such as inside usage [3], combinational usage [3], and abstraction [22][23]) of relations on characteristics of pattern networks and patterns.

ACKNOWLEDGEMENT

Our thanks go to our shepherd Michael Weiss for his valuable comments and guides to improve our paper significantly during its shepherding process. Moreover we thank Writer's Workshop leader and participants to give us valuable comments for improvement.

REFERENCES

[1] Hironori Washizaki, Masashi Kadoya, Yoshiaki Fukazawa and Takeshi Kawamura, "Network Analysis for Software Patterns including Organizational Patterns in Portland Pattern Repository," Agile 2014 Conference, Research Lightning Talk, Orlando, July 28-August 1, 2014.

[2] Atsuto Kubo, Hironori Washizaki, Atsuhiro Takasu and Yoshiaki Fukazawa, "Extracting Relations among Embedded Software Design Patterns," Integrated Design & Process Science, vol.9, no.3, 2005.

[3] Walter Zimmer, "Relationships between design patterns," In Pattern Languages of Program Design, Vol.1, Addison-Weslay, 1995.

[4] Cunningham & Cunningham, Inc., "Portland Pattern Repository," http://c2.com/ppr/

[5] Rob Miller, "WebSPHINX: A Personal, Customizable Web Crawler," http://www.cs.cmu.edu/~rcm/websphinx/

[6] Stephen Borgatti and Marting Everett, "A Graph-Theoretic Perspective on Centrality," Social Networks, Vol.28, No.4, Elsevier, 2005.

[7] Vladimir Batagelj and Andrej Mrvar, "Pajek - Program for Large Network Analysis," http://pajek.imfm.si/

[8] “Organizational patterns” from Wikipedia, http://en.wikipedia.org/wiki/Organizational_patterns

[9] Ward Cunningham, “Episodes: a pattern language of competitive development,” In Pattern Languages of Program Design, Vol.2, Addison-Wesley, 1996.

[10] Scrum Pattern Community and ScrumPLoP, “Scrum Patterns,” https://sites.google.com/a/scrumplop.org/published-patterns/home

[11] James O. Coplien and Jeff Sutherland, “Software Scrum Patterns,” 2008, https://sites.google.com/a/scrumorgpatterns.com/www/scrumpatternssummary

[12] Mike Beedle, Martine Devos, Yonat Sharon, Ken Schwaber and Jeff Sutherland, “SCRUM: An extension pattern language for hyperproductive software development,” Addison-Wesley Sofware Patterns Series, 1999.

[13] James O. Coplien and Neil B. Harrison, “Organizational Patterns of Agile Software Development,” Prentice Hall, 2004.

[14] James O. Coplien, “Organizational Patterns,” in Pattern Languages of Program Design, Addison-Wesley, 1995.

[15] Neil B. Harrison and James O. Coplien, “Patterns of productive software organizations,” Bell Labs Technical Journal, Vol.1, No.1, 1996.

[16] Linda Rising, “The Pattern Almanac,” Addison Wesley, 2000.

[17] Activate Networks, “Who is central to a social network? It depends on your centrality measure,” http://www.activatenetworks.net/who-is-central-to-a-social-network-it-depends-on-your-centrality-measure/

[18] Jeffrey Stuckman and James Purtilo, “Analyzing the Wikisphere: Methodology and Data to Support Quantitative Wiki Research,” Journal of the American Society for Information Science and Technology, Vol.62, No.8, 2011.

[19] M. E. J. Newman, "Detecting community structure in networks," The European Physical Journal B - Condensed Matter and Complex Systems, Vol.38, No.2, 2004.

[20] Atsuto Kubo, Hironori Washizaki, Atsuhiro Takasu, Yoshiaki Fukazawa, “Analyzing relations among software patterns based on document similarity,” International Conference on Information Technology Coding and Computing (ITCC 2005), Las Vegas, April 4-6, 2005.

[21] Hironori Washizaki, Atsuto Kubo, Atsuhiro Takasu, Yoshiaki Fukazawa, “Relation analysis among patterns on software development process,” 6th International Conference on Product Focused Software Process Improvement (PROFES 2005), Oulu, July 13-15, 2005.

[22] Eduardo B. Fernandez, Hironori Washizaki, Nobukazu Yoshioka, Atsuto Kubo, Yoshiaki Fukuzawa, “Classifying security patterns,” The 10th Asia Pacific Web Conference (APWeb2008), Shenyang, China, April 26-28, 2008.

[23] Eduardo B. Fernandez, Hironori Washizaki, Nobukazu Yoshioka, “Abstract security patterns,” 15th Conference on Pattern Languages of Programs (PLoP 2008), 2008.