

Network Analysis for Software Patterns including Organizational Patterns in Portland Pattern Repository

Hironori Washizaki
Masashi Kadoya
Yoshiaki Fukazawa
Waseda University
3-4-1 Okubo, Shinjuku-ku, Tokyo, 169-8555 Japan
washizaki@waseda.jp

Takeshi Kawamura
USOL Tokyo Co., Ltd.
1-1-1 Toyosu, Koto-ku, Tokyo 135-8560 Japan
takeshi.kawamura@usol.co.jp

Abstract— Organizational patterns are known as the basis for Agile software development movement. Patterns usually form a network having relationships among them to support users understand and utilize patterns efficiently and effectively. However little is known about the nature of pattern networks, such as how are organizational patterns different from other patterns from the viewpoint of centrality. To clarify such characteristics, we mine a network consisting 285 patterns including 15 organizational patterns from an existing online pattern repository called Portland Pattern Repository. By applying network analysis techniques to the mined network, we revealed several interesting characteristics of the pattern network and organizational patterns such as that the degree centrality seems to somewhat reflect the commonness and generality of the corresponding pattern.

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

I. INTRODUCTION

Software pattern is a general reusable solution to a commonly occurring problem within a given context while software development and management. Among various software patterns, organizational patterns are known as the basis for Agile software development movement, especially for Scrum and Extreme Programming. For example, Episodes [1], a pattern language containing organizational patterns, describes key points of agile development [2]. Moreover essential elements and practices of Scrum are now described as organizational patterns and related process patterns [3-4].

Each pattern could have some relations to other patterns [5]. 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. Especially for agile development community and people, it is

beneficial to clarify the nature of organizational patterns in pattern networks.

II. ANALYSIS PROCEDURE

We mine a network consisting 285 patterns including organizational patterns from an existing online pattern repository called Portland Pattern Repository (PPR) [6] as of March 2013. 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 is last edited on April 2013. We analyzed the pattern network by the following steps.

1. We collected 483 pattern names, incoming and outgoing relations, and belonging groups by crawling Wiki pages linked from “Pattern Index” and “Category Pattern” page as patterns. At this time we eliminated pages containing two or more patterns, in order to ensure that each page contains just one pattern. Many of patterns have just one category specified by original page authors. However some are with multiple categories; in that case we choose one category manually by considering major property of target patterns.
2. We manually filtered out several non-pattern pages, and obtained 285 patterns and 20 groups.
3. We measured three major types of centrality (degree, closeness and betweenness) [7] for each pattern. When measuring centrality, we did not distinguish the direction of relations; measuring centrality for each direction (i.e. incoming or outgoing) could be our future work.

III. ANALYSIS RESULTS

Fig 1 illustrates entire pattern network; in Fig 4, 10 highlighted nodes indicate patterns having high degree centrality (> 0.03). These highlighted patterns are mostly design patterns such as “Model View Controller” and “Adapter”; however there is one organizational pattern “Scape Goat” having high degree centrality.

Many patterns have small degree centrality; it means that many patterns refer to small number of patterns by Wiki page

link. However the distribution does not follow the well-known network property “power law”. The distribution of patterns regarding closeness centrality seems to follow the normal distribution. Most of patterns have very low betweenness centrality; it means that in the pattern network few patterns play a role of hub that connects different pattern groups.

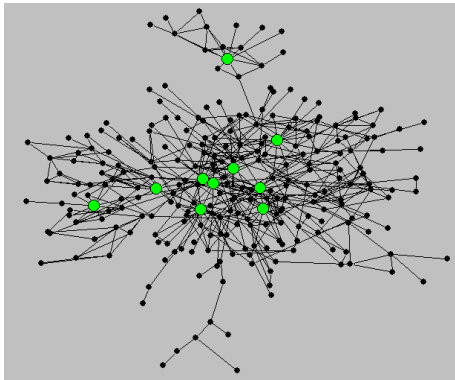


Fig. 1. Entire pattern network in PPR (each node indicates a pattern; each link indicates a relation between two patterns.)

Fig 2 show the box plot chart of degree centrality for each group. In Fig.2, groups tend to have different distribution of degree centrality. Among them, organizational patterns (ID 14) tend to have somewhat wide range of degree centrality compared with other groups except for some major groups (such as ID 17: Design Patterns).

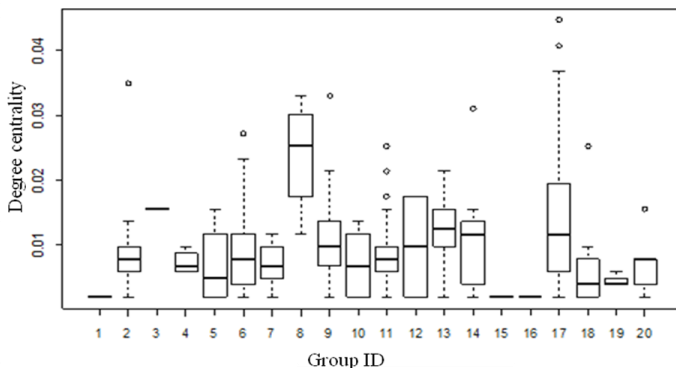


Fig. 2. Box plot of degree centrality for each group

Table I 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 betweenness centrality of each organizational pattern. In Table I, many organizational patterns have several related patterns. Especially, there are three patterns having high degree centrality (i.e. having many related patterns) : “Scape Goat”, “Peace maker” and “Train Hard Fight Easy”. “Scape Goat” is an anti-pattern, and other two are normal positive patterns. By reviewing these patterns carefully, these patterns are thought to be general ones and commonly used (or should be commonly avoided) in many projects because these are not so specific to a certain context, compared with other specific patterns having low degree centrality, like “Slow Poison” and “Change Of Setting”. It means that the degree centrality seems to somewhat reflect the commonness and generality of the corresponding pattern.

Regarding the betweenness centrality, “Scape Goat” seems to play a role of hub that connects organizational patterns to other patterns mostly about product design.

TABLE I. MEASUREMENT RESULTS OF ORGANIZATIONAL PATTERNS

Pattern name	N. patterns referred by the pattern	N. patterns referring to the pattern	Degree centrality	Closeness centrality	Betweenness centrality
Scape Goat	6	10	0.031128	0.176875	0.013398
Peace Maker	4	4	0.015564	0.15091	0.000084
Train Hard Fight Easy	6	2	0.015564	0.151088	0.000334
Cargo Cult	3	4	0.013619	0.210311	0.027906
Crypto Cracy	5	2	0.013619	0.175787	0.00646
Cult Of Personality	3	4	0.013619	0.150999	0.000258
Door Mat	2	4	0.011673	0.150733	0
Guru Does All	2	4	0.011673	0.150999	0.003344
Containment Building	2	2	0.007782	0.175307	0
Lets Play Team	1	2	0.005837	0.131458	0
Brownian Motion	1	1	0.003891	0.149593	0
Train The Trainer	1	1	0.003891	0.131323	0
Trial Project	1	1	0.003891	0.131323	0
Change Of Setting	1	0	0.001946	0.205272	0
Slow Poison	1	0	0.001946	0.15038	0

IV. CONCLUSION AND FUTURE WORK

The above-mentioned findings could contribute to software development community for understanding, reusing and extending existing patterns and writing new ones. Especially for agile development community and people, these findings could be beneficial to clarify the nature of organizational patterns in pattern networks. For example, developers or managers who want to form agile teams and conduct agile developments could consider reusing those organization patterns in PPR starting by referring to ones having high degree centrality such as “Scape Goat” and “Peace Maker”.

In the analysis, we regarded page links as pattern relationships; however formers are based on page authors' awareness of other patterns so that they might be different from pattern authors' intentions on relationships. We will handle this threat to validity by referring to relationships specified in original pattern documents if available in the future. Moreover we will investigate how are these findings related to actual agile or non-agile software development adapting organizational patterns and product ones.

REFERENCES

- [1] Ward Cunningham, “Episodes: a pattern language of competitive development,” In Pattern Languages of Program Design, Vol.2, Addison-Wesley, 1996.
- [2] “Organizational patterns” from Wikipedia, http://en.wikipedia.org/wiki/Organizational_patterns
- [3] Scrum Pattern Community and ScrumPLoP, “Scrum Patterns,” <https://sites.google.com/a/scrumplp.org/published-patterns/home>
- [4] Mike Beedle, Martine Devos, Yonat Sharon, Ken Schwaber and Jeff Sutherland, “SCRUM: An extension pattern language for hyperproductive software development,” Addison-Wesley, 1999.
- [5] Atsuto Kubo, Hironori Washizaki, Atsuhiko Takasu and Yoshiaki Fukazawa, “Extracting Relations among Embedded Software Design Patterns,” Integrated Design & Process Science, vol.9, no.3, 2005.
- [6] “Portland Pattern Repository,” <http://c2.com/ppr/>
- [7] Stephen Borgatti and Marting Everett, “A Graph-Theoretic Perspective on Centrality,” Social Networks, Vol.28, No.4, Elsevier, 2005.