

Composing Organizational Patterns More Precisely

Waheedullah Sulaiman Khail*

Institute of Informatics, Information System and Software Engineering
Faculty of Informatics and Information Technologies
Slovak University of Technology in Bratislava
Ilkovičova 2, 842 16 Bratislava, Slovakia
wsulimankhail@gmail.com

Abstract

Organizing people is very important and one of the great challenges, and particularly in software development. Organizational patterns document organizational knowledge, which can be used to build an organization from the start or solve existing issues within organizations. Although patterns in isolation are not uncommon, they are more effective when they are composed and applied together in sequences. In order to apply organizational patterns, one has to understand not only the patterns, but also the relationships between them.

Although there has been much research done on the patterns and how patterns are applied, little work has been done in finding out the relationships between patterns. What the thesis reported in this paper states is that if organizational patterns are composed more precisely, that will improve the accessibility, comprehensibility, and applicability of the organizational knowledge. The main contributions of the thesis are all in favor of the thesis statement: identifying the relationships between patterns in a pattern format makes these relationships explicit, using text mining techniques and natural language processing techniques helps identify and extract further relationships between patterns, formalizing the relationships between patterns with a context free grammar makes the relationships exact, while applying the pattern format once again to capture whole pattern languages makes them more comprehensible.

Each of the approaches was evaluated on nontrivial examples. Additionally, a comprehensive study involving thirty participants from ten organizations and five countries was conducted. It embraced sixteen organizational patterns. Furthermore, a scenario was developed where these approaches were applied in synergy. The study re-

*Recommended by the thesis supervisor: Assoc. Prof. Valentino Vranić
To be defended at Faculty of Informatics and Information Technologies, Slovak University of Technology in Bratislava on xx June, 2021.

© Copyright 2021. All rights reserved. Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies show this notice on the first page or initial screen of a display along with the full citation. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, to republish, to post on servers, to redistribute to lists, or to use any component of this work in other works requires prior specific permission and/or a fee. Permissions may be requested from STU Press, Vazovova 5, 811 07 Bratislava, Slovakia.

sults confirm that applying the approaches proposed in the thesis reported here makes organizational knowledge more accessible, comprehensible, and applicable. These approaches also help practitioners navigate through patterns which allow them to identify, learn, and apply proven techniques in their organizations.

Categories and Subject Descriptors

H.4 [Information Systems Applications]: Miscellaneous;
D.2.8 [Software Engineering]: Metrics—*complexity measures, performance measures*

Keywords

patterns, organizational patterns, formalization, grammar, association mining

1. Introduction

A pattern is an element of design that solves a problem by adding structure to a system [6]. Patterns can be applied in isolation. However, the full power of patterns lies in applying them in a network [17]. A pattern doesn't exist apart from a pattern language: its main purpose is to establish connections to other patterns in the language [2]. Software community adopted Alexander's ideas: particularly in object-oriented programming [15] and agile software development [6].

Organizational patterns of agile software development by Coplien and Harrison can be used to build an organization from start or solve an organizational issue within an organization [6]. Application of a pattern has resulting consequences. These consequences show where this pattern can fit in a larger, more comprehensive design and which other patterns can improve this design. This is where the relationships between patterns are created [3]. For example, a larger pattern might be the *Size The Organization* pattern, where the organization will build a start up team. Growing the organization graciously is part of building the team. One way to grow the team is through the *Phasing It In* pattern. Another way to do this is through the *Apprenticeship* pattern.

Although there has been much research done on the patterns and how patterns are applied, little work has been done in finding out the relationships between patterns [11]. What the thesis reported here states is that if organizational patterns are composed more precisely, that will improve the accessibility, comprehensibility, and applicability of the organizational knowledge. The main contributions of the thesis reported here are all in favor of

the thesis statement and are presented in some detail in the remaining part of this report as follows. Identifying the relationships between patterns in a pattern format makes these relationships explicit (Section 2), using text mining techniques and natural language processing techniques helps identify and extract further relationships between patterns (Section 3), formalizing the relationships between patterns with a context free grammar makes the relationships exact (Section 4), while applying the pattern format once again to capture whole pattern languages makes them more comprehensible (Section 5). Apart from evaluating each of the approaches, an overall evaluation of the approaches in synergy has been conducted, too (Section 6).

2. Reflecting Pattern Relationships in a Pattern Format

Pattern authors put a lot of effort into writing their patterns. However, if readers do not understand patterns or the links between them, then it will be difficult to apply them in sequences. Numerous patterns have been documented. However, they are not well connected [10].

Relationships between patterns are identified by describing areas of interest of patterns in a pattern language. However, relationships to other patterns are hidden within the pattern description. It is hard to identify them in a pattern. In this approach, those keywords which can potentially be used as relationship links to other patterns are identified. These keywords (relationship links) can be called indirect relationships.

In order to recognize the relationship between patterns, the pattern format has been amended to reflect the relationship links [14]. Subsections called *Keywords* to the problem and solution sections of a pattern are added. The *Keywords* subsection in the problem section of a pattern includes relationship links “keywords” to all the patterns that can be applied before this pattern, while the *Keywords* subsection in the solution section of a pattern includes links to all the patterns that can be applied after this pattern. The added subsections will reflect the indirect relationships in the form of “keywords.” These relationship links or the indirect relationships can be used to create proper pattern sequences.

With these relationship links, one can identify which patterns can be used together in sequences. Here are three example pattern sequences that are created based on the matching keywords:

- *Size The Organization -> Phasing It In -> Domain Expertise In Roles -> Developing In Pairs -> Group Validation*
- *Size The Organization -> Domain Expertise In Roles -> Phasing It In -> Developing In Pairs -> Group Validation*
- *Developing In Pairs -> Gate Keeper -> Firewalls -> Size The Organization -> Phasing It In.*

Adding the *Keywords* subsections to the pattern format makes organizational pattern relationships more readable. The relationship links in the form of keywords help in selecting the correct set of patterns for the problem in hand.

3. Extracting Relationships Between Organizational Patterns Using Association Mining

Patterns are not a new phenomenon, but there are still difficulties in selecting the appropriate pattern for a particular existing problem [8, 4]. To achieve a better result and fully utilize an appropriate organizational pattern, one must thoroughly understand not only the pattern itself, but also the pattern language it belongs to [2, 12, 13]. Furthermore, pattern languages, in general, do not always reveal every reliable connection with other patterns [7].

Most of the authors refer to other related patterns in the pattern description. However, there are many hidden relationships between patterns in the pattern description. These hidden relationships can be described by areas of interest.

The relationships between patterns are defined on a semantic level. The usefulness of pattern sequences solely depends on the nature of these relationships. These relationships can be extracted automatically using text mining and natural language processing techniques.

For this, an automatic approach is introduced that will extract similarities and connection links between organizational patterns [16]. This involves text mining techniques, such as the bags-of-words model, n-gram model, tf-idf, and association mining.

The term frequency-inverse document frequency (tf-idf) is a numerical statistic that reflects the importance of a word to a document in a collection. It is used as a weighting factor in text mining. The tf-idf is used to weigh and score the bag-of-words vocabulary for discovering relationships between organizational patterns. The relationship strength is determined from the percentage of the vocabulary that appears in the patterns.

The tf-idf is used to scale the frequency of words by how often they appear across the documents. The words with document frequency one are disregarded. By having a bag-of-words in place, the required matrix for mining association rules is built using the Apriori algorithm [1]. The columns of the matrix represent organizational patterns, while rows represent the vocabulary of the bag-of-words, where each row indicates the existence of the vocabulary word against each organizational pattern. The existence criteria is constrained by the tf-idf parameter.

This approach extracts relationships/associations that exist between organizational patterns. This helps not just in navigating and selecting a pattern to be applied in a given context, but it also helps in identifying which pattern is best suited after a given pattern. This way, patterns can be composed meaningfully and more useful and appropriate pattern sequences will be created.

4. Formalizing Pattern Relationships with Grammars

To deal with a large number of patterns, software developers, managers and any stakeholder involved in software development need tool support for finding the pattern that will address their certain problems. For this purpose, a formal approach which will help in selecting the right pattern for the problem in hand is required.

A defining characteristic of any language is its grammar [9]. It is from a grammar that one can generate sequences of patterns. For this, the relationships between organizational patterns are formalized using a context free grammar. The organizational pattern language grammar helps in pattern selection based on the description of a pattern. First, the process of capturing a pattern language with a grammar is explained. An example grammar for the people and code pattern language [6] is also provided. The example grammar includes production rules for sixteen patterns from organizational patterns repository [6]. Using this approach more production rules can be created for any specific pattern language. The results are evaluated by applying them in various organizations and interviewing their users in these organizations.

This approach will help practitioners identify correct sets of patterns for their problems. The user can input a problem and the grammar will propose a set of patterns which will resolve the problem. The organizational pattern language grammar will also help practitioners navigate through patterns.

The organizational pattern language grammar can also be used to evaluate organizations and their daily business activities. Practitioners can identify those concerning activities in their organizations that do not comply to these proven patterns. This allows them identify any gaps in their organizations and address these gaps by applying correct set of patterns proposed by the grammar and become a high-performance organization.

5. Representing Pattern Sublanguages as Patterns

The pattern format can be applied once more to capture whole pattern languages and make them more comprehensible for purposes of easier finding of appropriate pattern sequences for the problem at hand [13]. This is not a mere application of the pattern format and that pattern languages can actually be perceived as patterns, as others have observed, too [5]. More precisely, this approach does not target whole pattern languages, but their meaningful parts that can be seen as sublanguages. The resulting patterns can be called summary level patterns.

The *New Project* pattern has been captured using this approach. This pattern emerged from the pattern story of establishing a new project [6]. As organizational patterns are like structures in an organization, the *New Project* pattern is adding some organizational structure to the organization, so it can be perceived as yet another organizational pattern.

While a pattern story tells a specific event or occurrence, the corresponding summary level pattern is more general and corresponds to many different situations with a matching context. Unlike a summary level pattern, pattern stories are not backed by strong arguments that state why a specific pattern was used. In other words, the discussion of forces is not covered.

Using the pattern grammar, pattern sequences can be created that can solve bigger problems or a problem as a whole. The derived pattern sequences can also be presented as patterns. By representing organizational pattern sublanguages as patterns, they become more compre-

hensible for the purposes of easier finding the appropriate pattern sequence for the problem.

6. Evaluation

In order to evaluate how the grammar representation helps in using organizational patterns specifically and patterns in general, a study was conducted in which software developers, architects, and managers were asked to create pattern sequences in order to solve certain design problems. The study was conducted in the form of in-depth interviews, where the application of organizational patterns was discussed with thirty participants working in software development organizations across five countries: Slovakia, Germany, Turkey, UAE, and Afghanistan. Twenty five pattern sequences and five pattern stories have been documented from the study.

These interviews were semi-structured and the list of questions was used only to guide discussions. The actual discussions were open-ended, with many unplanned questions and opinions.

This study results confirm that the grammar achieved the underlying goal of providing a formalization of organizational patterns and that it helps practitioners in understanding and applying organizational patterns.

7. Conclusions and Challenges

The thesis reported here brings four approaches that address the problem of composing organizational patterns more precisely in order to improve the accessibility, comprehensibility, and applicability of the organizational knowledge. Each of the approaches was evaluated on non-trivial examples. Additionally, a comprehensive study involving thirty participants from ten organizations and five countries was conducted. It embraced sixteen organizational patterns. Furthermore, a scenario was developed where these approaches were applied in synergy. The study results confirm that applying the approaches proposed in the thesis reported here makes organizational knowledge more accessible, comprehensible, and applicable.

Pattern composition is still a challenging task because of the current structure in which patterns exists. As their description is written in natural language, more sophisticated algorithms are required in order to extract relationship information automatically from pattern descriptions. Semantic analysis of these links will support pattern composition. To enable identifying the type of a mined relationship, i.e., is it either a solution of a problem or is it a supportive type, semantic information via natural language processing should be added.

8. Acknowledgments

The work reported here was supported by the Scientific Grant Agency of Slovak Republic (VEGA) under grant No. VG 1/0759/19 and by the Operational Programme Integrated Infrastructure for the project Research and Development of Software Solution with the Application of Blockchain Technology in the Field of International Rail and Container Transport of Goods (ITMS code: 313022V816), co-funded by the European Regional Development Fund (ERDF).

References

- [1] R. Agrawal, T. Imieliński, and A. Swami. Mining Association Rules between Sets of Items in Large Databases. In *Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data*, SIGMOD '93, page 207–216. Association for Computing Machinery, 1993.
- [2] C. Alexander. *The Timeless Way of Building*. Oxford University Press, 1979.
- [3] K. Beckers, S. Fassbender, and M. Heisel. Deriving a Pattern Language Syntax for Context-patterns. In *Proceedings of the 19th European Conference on Pattern Languages of Programs*, EuroPLoP '14, pages 2:1–2:25. ACM, 2014.
- [4] A. Birukou. A Survey of Existing Approaches for Pattern Search and Selection. In *Proceedings of the 15th European Conference on Pattern Languages of Programs*, pages 1–13, 2010.
- [5] F. Buschmann, K. Henney, and D. Schmidt. *Pattern-Oriented Software Architecture: On Patterns and Pattern Language*, volume 5. John Wiley & Sons, 2007.
- [6] J. O. Coplien and N. B. Harrison. "Organizational Patterns of Agile Software Development". Prentice-Hall, 2004.
- [7] M. Falkenthal, U. Breitenbücher, and F. Leymann. The Nature of Pattern Languages. *cit. on*, page 14, 2018.
- [8] T. Frtala and V. Vranic. "Animating Organizational Patterns". In *Proceedings of 8th International Workshop on Cooperative and Human Aspects of Software Engineering, CHASE 2015, ICSE 2015 Workshop*, Florence, Italy, 2015. IEEE.
- [9] K. Henney, E. Schlossberg, and A. Kelly. Context Encapsulation—Three Stories, a Language, and Some Sequences. In *In Proceedings of EuroPlop 2005*, Irsee Monastery, Germany, 2005.
- [10] G. Mussbacher, D. Amyot, and M. Weiss. Formalizing Patterns with the User Requirements Notation. In *Design Pattern Formalization Techniques*, pages 302–323. Igi Global, 2007.
- [11] A. Rouhi and B. Zamani. Formalizing Patterns and Pattern Languages: A Case Study Approach. In *Technical Report. Model-Driven Software Engineering Research Group*, University of Isfahan Isfahan, Iran, 2015.
- [12] N. Seidel. Empirical Evaluation Methods for Pattern Languages: Sketches, Classification, and Network Analysis. In *Proceedings of the 22nd European Conference on Pattern Languages of Programs*, pages 1–24, 2017.
- [13] W. Sulaiman Khail and V. Vranic. Treating Pattern Sublanguages As Patterns with an Application to Organizational Patterns. In *Proceedings of the 22Nd European Conference on Pattern Languages of Programs*, EuroPLoP '17, pages 6:1–6:12. ACM, 2017.
- [14] W. Sulaiman Khail and V. Vranic. Reflecting Pattern Relationships in a Pattern Format. In *Proceedings of the 24th European Conference on Pattern Languages of Programs*, pages 1–5, 2019.
- [15] J. Vlissides, R. Helm, R. Johnson, and E. Gamma. *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley, 1995.
- [16] S. Waseeb, W. S. Khail, H. G. Wahaj, and V. Vranic. Extracting Relations Between Organizational Patterns Using Association Mining. In *Proceedings of the European Conference on Pattern Languages of Programs 2020*, pages 1–9, 2020.
- [17] U. Zdun. Systematic Pattern Selection Using Pattern Language Grammars and Design Space Analysis. *Software: Practice and Experience*, 37(9):983–1016, 2007.
- S. Waseeb, W. Sulaiman Khail, H. G. Wahaj, and V. Vranic. Extracting Relations Between Organizational Patterns Using Association Mining. In *Proceedings of the 25th European Conference on Pattern Languages of Programs Online*, EuroPLoP 2020. ACM, 2020.
- V. Vranic A. Vranic, and W. Sulaiman Khail. Growing Organizations with Patterns: Lessons from Drama. In *Proceedings of the 25th European Conference on Pattern Languages of Programs Online*, EuroPLoP 2020. ACM, 2020.

Selected Papers by the Author

- W. Sulaiman Khail and V. Vranic. Treating Pattern Sublanguages as Patterns with an Application to Organizational Patterns. In *Proceedings of the 22nd European Conference on Pattern Languages of Programs*, EuroPLoP 2017. Kloster Irsee in Bavaria, Germany, 2017. ACM, 2017.
- W. Sulaiman Khail and V. Vranic. Reflecting Patterns Relationships in a Pattern Format. In *Proceedings of the 24th European Conference on Pattern Languages of Programs*, EuroPLoP 2019. Kloster Irsee in Bavaria, Germany, 2019. ACM, 2019.