Social Network Analysis and Mining (2025) 15:88
https://doi.org/10.1007/s13278-025-01510-2

ORIGINAL ARTICLE

Establishing pattern sequences using stochastic processes with an application to organizational patterns

Viktor Matovič[1] · Valentino Vranić[2] · Waheedullah Sulaiman Khail[3] · Neil Harrison[4]

Received: 10 March 2025 / Revised: 21 July 2025 / Accepted: 22 July 2025

Abstract

Software developers who solve recurring problems employ similar practices, which are documented and reused as patterns. Organizational patterns can fix organizational issues. Security patterns can be used to implement security measures into software or an organization. Expected solutions to complex problems of establishing security in the organization can be embodied in pattern sequences. Sequences of patterns that are expected to be used in practice cannot be established by applying stochastic processes, because multiple sequences may have the same probability of use. The symmetry of the relationship between patterns can be calculated using stochastic processes, and these symmetries can be used to distinguish pattern sequences that are truly expected to be used. This article presents two methods of establishing pattern sequences using stochastic processes. Both methods, based on stochastic trees and Bayesian networks, can be used to establish pattern sequences that are expected and unexpected in practice. The method that combines stochastic trees and symmetry of relationships between patterns can be used to establish sequences of conditionally independent patterns. The method that combines Bayesian networks and symmetry of relationships between patterns can be used to establish sequences of conditionally dependent patterns in multiple pattern languages or catalogs without constructing multiple stochastic trees. We used these methods to establish 19 sequences of organizational and security patterns. We show how expected sequences of organizational and security patterns can establish security in an organization. These two methods can also be used to establish expected sequences of patterns for engineering software for the cloud.

Keywords Organizational patterns · Pattern sequence · Pattern language · Stochastic trees · Bayesian networks · Security patterns

1 Introduction

Patterns are an organized way to exchange best practices between actors in a specific domain area (Alexander et al. 1977). Software developers dealing with commonly recurring problems converge on using similar practices documented and reused as design patterns. In this, software developers themselves are also organized according to patterns, which are known as organizational patterns of software development (Coplien and Harrison 2004). Pattern languages in the work of Alexander et al. (1977) relate patterns that are closely related and applicable together. Pattern sequences identify viable orders by which patterns in pattern languages can be applied (Alexander 2002).

Solving extensive problems requires the application of several patterns in a sequence. After applying one pattern, the question is which one to apply next. This is often expressed

✉ Viktor Matovič
xmatovicv@stuba.sk

Valentino Vranić
valentino.vranic@paneurouni.com

Waheedullah Sulaiman Khail
wsulimankhail@gmail.com

Neil Harrison
Neil.Harrison@uvu.edu

1 Faculty of Informatics and Information Technologies, Slovak University of Technology in Bratislava, Bratislava, Slovakia

2 Faculty of Informatics, Pan-European University, Bratislava, Slovakia

3 Helmholtz-Zentrum Berlin, Berlin, Germany

4 Utah Valley University, Orem, USA

Published online: 16 August 2025

explicitly in the descriptions of individual patterns or anecdotes on their application, known as pattern stories.

One cannot be sure about using any pattern at any given time, and the probabilities of their use must be calculated.

Text descriptions of patterns often contain links to other patterns without specifying which of these linked patterns are most likely to be applied next. Some explicitly linked patterns create pattern compounds and do not specify the order in which these patterns are expected to be applied in pattern sequences. Parts of the text descriptions of patterns may not contain links to all related patterns.

In our work, we used the strength of symmetry of the relationship between patterns to establish sequences from patterns, such that if the order of patterns used in these sequences is changed, the pattern sequences stay meaningful. These strengths of symmetries of relationships between patterns take into account conditional dependencies between patterns, which reflect applications of patterns as dependent events. In our work, we establish meaningful pattern sequences from patterns that have the strongest symmetry of relationship, and we do this without altering text descriptions of patterns or without statistics about the past use of these patterns.

In this article, we present two methods of establishing pattern sequences using stochastic processes. Both methods calculate the strengths of symmetric relationships between patterns using conditional probabilities to establish pattern sequences that are the most expected, the expected to be used, and the unexpected to be used. The meaningfulness of the three most expected and the two expected pattern sequences was evaluated by discussing their usefulness with participants in an experiment.

In effect, we rely on design science as a methodological approach, which involves two primary activities: the creation of new knowledge through the design of novel or innovative artifacts and the analysis of their use and/or performance with reflection and abstraction (Vaishnavi et al. 2015; Vaishnavi and Kuechler 2004–2017)).

The rest of the article is structured as follows. Section 2 explains how explicit and implicit relationships between organizational patterns establish pattern sequences. Section 3 proposes a method of establishing pattern sequences based on stochastic trees. Section 4 proposes a method for establishing pattern sequences based on Bayesian networks. Section 5 explains how to choose the right method. Section 6 discusses the results. Section 7 presents the evaluation. Section 8 concludes the article.

2 Existing methods of establishing pattern sequences

Pattern sequences are generally established based on explicit or implicit relationships between patterns.

2.1 Methods based on explicit relationships between patterns

Pattern descriptions contain references to other patterns. Many authors document relationships between patterns in text descriptions. According to Sulaiman Khail and Vranić (2019), links to related patterns occur mainly in the problem and solution sections. According to Waseeb et al. (2020), not all combinations of patterns based on the explicit links between patterns result in meaningful pattern sequences because the usefulness of these combinations depends on the nature of these relationships. According to Waseeb et al., hidden relationships between organizational patterns may suggest different combinations of patterns to solve complex problems. According to Waseeb et al., hidden relationships between organizational patterns can be extracted using natural language processing techniques. The method of extracting hidden relationships between patterns from their text descriptions presented by Waseeb et al. finds combinations of patterns useful when their text descriptions use similar words. Some combinations of patterns may be useful even when their descriptions do not use similar words.

Pattern users can use pattern catalogs, compounds, or languages to establish pattern sequences. Pattern languages relate patterns that often emerge together (Alexander et al. 1977), and pattern sequences identify viable orders by which patterns can be adopted.

Patterns with similar intent can be used together in pattern sequences if the application of these patterns outside the pattern sequence cannot achieve the desired goal. According to Buschmann et al. (2007), relating patterns using common intent is a technique preferred by pattern users. Buschmann et al. do not present any technique to compare commonalities between intents from text descriptions of patterns. A technique that takes into account conditional dependencies between patterns in establishing pattern sequences using patterns whose application helps achieve the desired goal should be proposed.

Conditional dependence between patterns can also be expressed by categorizing problems that these patterns solve into the same problem frame. Problem frames can also be used to identify commonly used patterns while solving specific problems categorized in problem frames. Patterns can provide incomplete solutions that require the application of another pattern categorized in a different problem frame.

Jackson's problem frames (Jackson 1995, 2001) is a technique that helps to understand recurring problems by describing them without using terms of the solution domain. Problem frame names, bounds, and describes this recurring problem by generalizing this problem into a problem class. The assumption is that if a new problem fits the same problem frame, methods that were used in the past can also solve this new problem. Problem frames document constraints on solutions provided by patterns in sequences. Problem frames name and bound problems that can be solved by applying multiple patterns in a sequence, but problem frames do not provide information about the expected order in which these patterns should be applied. The expected order in which patterns should be applied can be identified by calculating the conditional probabilities of use of patterns.

Sousa et al. (2022) note that the problem with using explicitly documented relationships between patterns to establish pattern sequences is that the ontology used for naming these relationships does not determine joint or conditional use of patterns.

Patterns are applied in sequences with varying probability. The value of this probability varies with previously used patterns. The conditional probability of using patterns in sequences must be calculated to determine the expected use of these patterns.

2.2 Methods based on implicit relationships between patterns

Coplien and Schmidt (1995) and Noble (1998) manually extracted implicit relationships between object-oriented design patterns from Gamma et al. (1994) by interpreting their text descriptions. Noble acknowledges that these implicit relationships can be used to identify expected combinations of patterns. Noble also identifies the need to identify implicit relationships between patterns in different pattern catalogs. Identifying implicit relationships from pattern catalogs with many patterns can be difficult. A method capable of extracting these implicit relationships from a large number of patterns should be devised.

Implicit relationships between patterns identifying their expected combinations are not present in the pattern languages of Coplien and Harrison (2004), nor in the catalog of security patterns from Cordeiro et al. (2022). If they were present, they could be used to establish pattern sequences. Coplien and Schmidt (1995) reorganizes design patterns based on their relationships to understand these patterns and apply them in software development. The patterns in catalogs and pattern languages can be understood by showing examples of expected pattern sequences established out of their patterns without the need to reorganize their patterns. The expected order of patterns in pattern sequences can be established using a combination of stochastic processes.

According to Falkenthal et al. (2018), pattern languages do not always reveal every reliable connection to other pattern languages that can be used to interrelate them and establish meaningful sequences out of their patterns. They present a method to extract inherent relations between patterns from text descriptions of patterns documented in pattern languages and visualize them in a pattern graph. The pattern graph from Falkenthal et al. (2018) can ease the search for meaningful pattern combinations because patterns expected to be combined are represented as adjacent nodes in this graph. Pattern sequences must also be established without the need to create this graph.

Janeiro et al. (2010) identify and classify relationships between user interface design patterns. They let participants in the study select user interface design patterns using classified relationships. Conditional probabilities of use of related patterns, which they documented using the Pattern Language Markup Language from Fincher et al. [9], could be calculated to compare how selections of patterns correlate to probabilities of their actual use. This comparison could discover new combinations of patterns that solve user interface design problems.

Buschmann et al. (2007) document techniques to categorize patterns expected to be used together or in pattern sequences.

Implicit relationships that can be used to establish pattern sequences, according to Waseeb et al. (2020), can be described by areas of interest of authors documenting organizational patterns. Pattern sequences are hard to establish if these areas of interest are not properly documented in text descriptions of patterns.

Implicit relationships between patterns may necessitate a need to switch roles between pattern story participants if this action is necessary to continue in the pattern story (Vranić et al. 2020). Identification of conditional dependencies between patterns with stochastic processes could provide answers on why there is a need to switch roles in pattern story participants. Buschmann et al. (2007) provided examples of pattern stories to describe the use of combinations of patterns, meaningful pattern sequences can also be identified by checking whether they can be described in pattern stories.

According to Seidel (2017), implicit relationships between patterns in pattern languages can be hidden emphasis or gaps because the emphasis of a pattern language often stays unrepresented in its inherent structure, but they cannot be used to establish pattern sequences because they are rarely provided.

The hierarchical structure between patterns also expresses implicit relationships between patterns, which can be used to establish pattern sequences. Hierarchical structure between patterns is rarely provided by pattern authors.

As it was found in our previous work, the expected order in which patterns are expected to be used is another form of an implicit relationship between patterns (Matovič and Vranić 2024, 2025).

According to Sousa et al. (2022), verbs used to express relationships between patterns for engineering software for the cloud do not imply joint use of patterns, conditional correlation, or causality between these patterns. They documented a pattern language for engineering software for the cloud. Text descriptions of its patterns lack explicit links to other patterns that could be used to establish pattern sequences. Implicit relationships between patterns in their language can be identified and used to establish pattern sequences with patterns that lack or do not have all documented explicit relationships. The absolute value of asymmetry of relationships calculated by Sousa et al. for patterns in their language could ease their comparison to identify patterns with the strongest and weakest asymmetry of relationships.

Sousa et al. (2022) denote the difference between the probabilities of the subsequent use of patterns as asymmetry in pattern adoption. For patterns X and Y, this is the $p(X|Y) - p(Y|X)$ value. According to Sousa et al., relationships between patterns identify more useful combinations of patterns for engineering software for the cloud, the closer the difference of probabilities of the subsequent use of patterns X and Y: $p(X|Y) - p(Y|X)$ is to zero. As they explain further, asymmetry in pattern adoption can be used to identify correlated patterns expected to be used together, because correlated patterns expected to be used together are patterns X and Y, where the difference between the conditional probability of the subsequent use of pattern X after pattern Y and the opposite probability to it calculated with Bayes' rule is lower than the difference between the conditional probability of the subsequent use of pattern X after any other pattern Z and the opposite probability to it: $|p(X|Y) - p(Y|X)| < |p(X|Z) - p(Z|X)|$.

According to Coplien and Zhao (2000), understanding the relationship between programming language features and design patterns regarding symmetry and symmetry breaking may answer which patterns should and should not be used. They assume that applying design patterns temporarily breaks symmetry in systems where software patterns are applied. Asymmetry in these systems can be transformed back to symmetry by applying more design patterns. It is important to note that this concept of symmetry is different from the symmetry of relationships between two organizational or security patterns.

Washizaki et al. (2014) connected text descriptions of organizational patterns from the Portland Pattern Repository in the network using explicitly stated links to other patterns. Using this network, Washizaki et al. calculated three types of centrality of patterns that can be used to identify nodes representing patterns in a network that are commonly used in practice. The strength of symmetry of relationships between patterns can be used to establish pattern sequences without requiring connecting these patterns in a network. Pattern sequences can be established using the strongest symmetries of relationships between patterns without requiring these patterns to have explicit relationships with each other.

Janeiro et al. (2010) found common relationships between user-interface design patterns, which can be used to establish pattern sequences. The expected order of patterns in sequences can be found in other software patterns. Janeiro et al. identified commonly used patterns from the experiments with participants solving their problems with a list of pre-defined patterns. Information about the previous use of software patterns is not mandatory while establishing expected pattern sequences, if stochastic processes are used to calculate the probability of use of patterns and pattern sequences.

Kodituwakku and Bertok (2003) found the types of relationships between design patterns that can be used to organize patterns and establish the sequences of them. Kodituwakku et al. require categorization of design patterns before commonly used patterns can be identified. Re-categorization of software patterns is not mandatory before establishing expected pattern sequences if symmetries of relationships are used to establish these sequences, because these symmetries can be calculated between explicitly or implicitly related patterns.

Sulaiman Khail and Vranić (2019) proposed changes to the pattern format to more sophisticatedly reflect the relationships with other patterns. They assume that relationships between patterns can be used for pattern composition. If the strengths of symmetries of relationships between patterns are provided in pattern text descriptions, these strengths would help pattern authors to see a natural path through pattern catalogs or languages.

Greene (2003) implemented a software tool to document patterns in XML format that can be used to select appropriate patterns for a particular problem using decision trees. There are many pattern formats, and pattern sequences can be established only using pattern relationships without putting constraints on pattern formats.

Kubo et al. (2005) use a pattern relation graph to identify similar contexts in software design patterns and uses this similarity as a guide to establishing pattern sequences. Because contexts of consecutively applied patterns are similar, patterns differ in the contradicting forces they try to resolve. Pattern sequences that are expected to be used do not have to consist of patterns that have similar contexts.

Kaliyar (2020) presented a language model based on the BERT language model, which can be used to understand links to other patterns using the context in which these links occur, thus shifting from lexical analysis of the text to syntactic analysis. Text descriptions of patterns that are meant to be established in pattern sequences should be available to pattern users, such that they will be able to understand the usefulness of the established pattern sequences.

Noble (1998) named relationships between patterns that can be extracted from their text descriptions as relationships between primary and secondary patterns. According to his definitions, primary patterns reference other patterns that can be used together and be found in the *related patterns* or *see also* sections. Noble (1998) recognizes implicit relationships between object-oriented design patterns about which the author of the patterns does not have to know. The strengths of symmetries of relationships between patterns can be considered as implicit secondary relationships between patterns.

Falkenthal et al. (2018) searched for natural sequences in which patterns in pattern language tend to be established using an Alexandrian pattern graph. These natural sequences can also be established using patterns documented in other sources, not only in pattern languages.

According to Sousa et al. (2022), directed graphs can represent natural adoption sequences of patterns and help identify the precedence of patterns in use. The same can be achieved by visualizing the Bayesian network for the natural adoption sequences. Natural adoption sequences can also be established without visualizing pattern relationships in graphs.

Furthermore, Sousa et al. (2022) searched for what causes companies to use patterns. They strived to find causal relationships between the patterns companies use. They used the adoption rate to discuss the prevalence of one type of pattern against the others. They acknowledge that relationships between patterns extend the borders of the pattern language. They evaluated the usefulness of established pattern sequences by surveying professionals developing or managing software development for the cloud. Also, Sousa et al. (2022) note that very few authors assess pattern use with practitioners. This can lead to avoiding working with patterns that might be conditionally usable because their use in the sequence depends on the technology used for their implementation. One cannot rely only on documented and explicitly stated relationships between patterns when choosing which pattern to apply next. Sousa et al. imply that adoption sequences of two patterns are very likely to be used if the probability of their use is higher than 0.8, and adoption sequences are only likely to be used if the probability of their use is between 0.6 and 0.8. Pattern sequences that are very likely to be used can be established using the strongest symmetries of relationships between patterns while establishing pattern sequences. Probabilities of use of these sequences can be calculated using stochastic processes.

Waseeb et al. (2020) proposed an automatic approach to discover relationships and the strength between patterns using text mining and natural language processing techniques. They used text descriptions of the organizational patterns documented by Coplien and Harrison (2004) to mine association rules. Mining associations between patterns says nothing about the expected order in which associated patterns should be applied. Semantic meaning of text descriptions should also be used while extracting implicit pattern relationships. The process of mining relationships between patterns from their text descriptions must also understand the meaning of the text, not just the pattern names mentioned within the text descriptions of patterns. Any method of establishing pattern sequences using implicit relationships should support pattern users to let them choose which relationships will be used to establish pattern sequences.

2.2.1 Alternatives to symmetries of pattern relationships

Variational distance discussed by Koller and Friedman (2009) and presented in Formula (1) can be used to calculate how much the probability of use of a pattern sequence would be changed by the use of the next applicable pattern in this pattern sequence. The higher this probability is, the greater the incompatibility of patterns in this pattern sequence is.

$$D_{var}(P, Q) = (1/2) * \sum_{applicable\ pattern} |P(original\ sequence) - Q(extended\ sequence)| \tag{1}$$

Because the pattern sequence that is expected to be used in practice should be composed of compatible patterns that can be applied together, the probability of use of this pattern sequence should be changed only minimally.

The calculation of the probability of using a pattern sequence with the variational distance is possible if the probability of the pattern sequence and the probability of use of this sequence, extended with the use of another pattern, can be calculated. The problem with this metric is that if the probability of use of pattern sequences were calculated with stochastic trees, multiple pattern sequences with different patterns would have the same probability of their use, and it would be hard to identify the pattern that is most expected to be applied next. The same reason applies to the use of the Euclidean distance (Koller and Friedman 2009) and Chebyshev inequality (Koller and Friedman 2009).

Conditional relative entropy discussed by Koller and Friedman (2009) and presented in Formula (2) can also be used to calculate the probability with which two software patterns would be used together. The use of this entropy to calculate the probability with which two patterns are applied would be suitable if the probability of use of another software pattern $P(another\ pattern)$ in Formula (2) could be calculated by observing the relative frequency of its use, which can change in time and is also difficult to obtain in the domain of software patterns.

$$\sum_{another\ patterns} P(another\ pattern) * D_{KL}(P(first\ pattern \mid another\ pattern) \,||\, Q(first\ pattern \mid second\ pattern)) \quad (2)$$

Because the symmetry of relationships between patterns was previously shown to be used in identifying software patterns that can be combined (Sousa et al. 2022), and they can be calculated without having information about the past use of patterns, these symmetries can be checked to see if they can be used to establish meaningful pattern sequences.

2.2.2 Probabilities of symmetric pattern relationships

The expected order of patterns in pattern sequences can be identified by establishing patterns that have the strongest strengths of symmetries of relationships. In order to distinguish the most expected from expected pattern sequences and to distinguish expected from unexpected pattern sequences, the symmetries of the relationships between patterns have to be calculated because stochastic processes are insufficient to make this distinction. This is so because they can produce the same highest probability for multiple pattern sequences, even if one of them is not meaningful. The calculation of the strength of the symmetry of the relationship between patterns is dependent on the conditional probability of applying those patterns.

The probability of applying software patterns can be calculated as a relative frequency of their use, but this requires statistics about the past use of patterns. If the probability of use of all patterns in the pattern language or catalog would conform to a uniform probability, it would not be possible to determine which pattern is expected to be applied next.

The probability of applying explicitly linked patterns can also be calculated with the Beta distribution or the Metropolis algorithm discussed by Kruschke (2015) but the problem with these techniques is that they need statistics about the past use of patterns as input to calculations of these probabilities, which is hard to obtain. The probability of applying explicitly linked patterns could also be calculated with the Probability density function discussed by Kruschke (2015), but this function can be modeled only if we have the statistics about the past use of patterns in practice.

There are often multiple explicit relationships between patterns, but these do not specify which of these patterns are expected to be used. The strengths of symmetries of relationships between patterns can be used as one type of implicit relationship that can be extracted from text descriptions of patterns to identify patterns expected to be used together in pattern sequences. These strengths can be used to establish pattern sequences without statistics about the past use of patterns and without altering text descriptions of patterns. The strengths of symmetries of relationships between patterns can be calculated with the stochastic trees. The strengths of symmetries of relationships between patterns can be compared to establish pattern sequences that are most expected, expected, or unexpected to be used in practice.

Stochastic trees can be used to model any pattern sequence, and they can point to multiple sequences as those with the highest probability of their use. Probabilities inside the stochastic tree can be calculated as $1/N$ where N is the number of all patterns to be established in sequences, reduced by the number of already used patterns. Pattern sequences with the same highest probability of use calculated with stochastic trees are expected pattern sequence candidates, and without further information, it is not possible to determine which sequence is expected. These are initial sequences that are candidates for sequences expected to be used. As there may be more than one candidate for the expected pattern sequence, all candidate sequences identified in the stochastic tree (from nodes with the highest probabilities) must be examined to distinguish truly expected pattern sequences. One way to examine them is to check if subsequently used patterns in these sequences have the strongest symmetry of relationships. It turns out that the strongest strength of symmetry of the relationship between patterns can be used to establish meaningful pattern sequences. Bayesian belief networks can also be used to quantify the degree of belief in these expected pattern sequences.

3 Establishing pattern sequences based on stochastic trees

Here, a new method for establishing expected (probable) pattern sequences is proposed. The method combines Hazen's stochastic tree method (Hazen 1992) and the maximum-likelihood estimation method (Penn State 2023). A stochastic tree, which is a continuous-time Markov chain (Hazen 1992), is used to select expected pattern sequence candidates.

The input to the method is a set of text descriptions of patterns. Text descriptions of patterns used in this method do not have to adhere to any pattern forms commonly used for documenting patterns. The output is a set of expected pattern sequences and a set of unexpected pattern sequences. Expected pattern sequences are recommended, while unexpected ones are to be avoided. The method consists of these steps:

1. Establish all meaningful kick-off pattern sequences using all patterns such that these sequences consist of at least two unique patterns using explicit or implicit relationships.

2. Construct all stochastic trees of pattern relationships on top of all kick-off pattern sequences established in Step 1.
3. Sort probabilities of the stochastic tree nodes in descending order, identify and select the stochastic tree nodes with the highest probabilities because these nodes represent the expected pattern sequence candidates. If there is more than one node in a stochastic tree with the highest probability, all of these nodes must be selected because they identify expected pattern sequence candidates.
4. Construct the pattern map of applicable patterns of the first pattern in all expected pattern sequence candidates selected in step 3.
5. For each pattern map of applicable patterns constructed in step 4, extract conditional probabilities that the patterns used in step 4 would be used before other applicable patterns in pattern map of applicable patterns $p(\textit{another pattern}|\textit{pattern})$ and extract these probabilities from the stochastic tree constructed in step 2 which contains this probability. For each pattern map of applicable patterns constructed in step 4, calculate conditional probabilities that the patterns used in step 4 would be used after other applicable patterns in the pattern map of applicable patterns $p(\textit{pattern}|\textit{another pattern})$ with Bayes' rule. For each pattern map of applicable patterns constructed in step 4, calculate the absolute difference $|p(\textit{another pattern}|\textit{pattern}) - p(\textit{pattern}|\textit{another pattern})|$ and store this difference in the list. Attach this list to the pattern map of applicable patterns.
6. Sort lists attached to pattern maps constructed in step 5 in ascending order.
7. Lowest absolute differences in sorted lists from step 6 point to patterns expected to be applied next after patterns for which pattern maps of applicable patterns were constructed in step 4.
8. Absolute differences in sorted lists from step 6, which are not lowest in these lists, point to patterns unexpected to be applied next after patterns for which pattern maps of applicable patterns were constructed in step 4.
9. If the patterns established by following steps 4–7 are the same as those in the expected pattern sequence candidate identified in step 3, then add this candidate sequence to the resulting set of expected pattern sequences.
10. If the patterns established by following steps 4–7 are different from patterns in the expected pattern sequence candidate identified in step 3, then a different pattern sequence is established and added to the resulting set of expected pattern sequences.
11. If the patterns established by following steps 4–8 are the same as those in the expected pattern sequence candidate from step 3, then add this candidate sequence to the resulting set of *unexpected* pattern sequences.
12. If the patterns established by following steps 4–8 are different from patterns in the expected pattern sequence candidate identified in step 3, then a different pattern sequence is established and added to the resulting set of unexpected pattern sequences.

The search for expected pattern sequences can be started by identifying those sequences that are candidates for expected pattern sequences. Candidates for expected pattern sequences can be identified using the visualization tool like the stochastic tree used in Sect. 3.1. Before the expected pattern sequence candidates can be identified from stochastic trees, these trees must model some sequences that we established from all patterns in the Piecemeal Growth pattern language in Sect. 3.1 to learn more about how the organization starts to work together. Candidates for expected pattern sequences are those sequences that have the highest probability in the stochastic tree, but there is no guarantee that pattern sequences with the highest probability in the tree will always be meaningful because stochastic trees can be used to model any possibly meaningless pattern sequence. Implicit relationships like the strongest symmetry of relationship we used in Sect. 3.2 can be used to establish meaningful pattern sequences from patterns, whose order in these sequences can be reversed.

3.1 Construction of the stochastic tree

Before the expected pattern sequences can be established, sequences that are candidates for expected sequences must be identified. No pattern sequence with a high probability of its use can be declared as expected, as there can be multiple pattern sequences with the same probability.

Candidate pattern sequences for expected pattern sequences can be identified from a visualization of the stochastic tree modeling any input pattern sequence (which we call the kick-off pattern sequence). Kick-off pattern sequence can be established using explicit links between patterns. The kick-off pattern sequence we used was established by chaining all patterns in the Piecemeal Growth pattern language from Coplien and Harrison (2004) in the order they are documented in this language (they are also linked to each other) to learn more about the piecemeal growth of an organization.[1] The following kick-off pattern sequence

[1] This kick-off sequence can be seen here: https://github.com/viktorFIIT/fiit-research-resources/blob/main/app/Use-On-Organizational-Patterns-1.md.

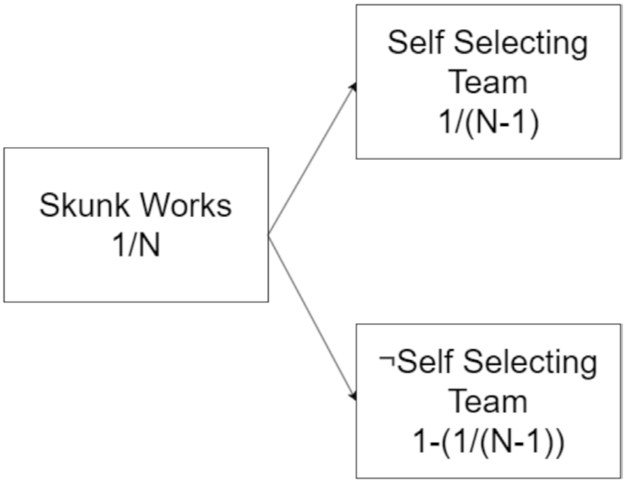

Fig. 1 An example of a stochastic tree for pattern sequence Skunk Works → Self-Selecting Team

was established by following explicit links between patterns in the Piecemeal Growth pattern language, if these explicit links mentioned other applicable patterns. For example, Skunk Works mentions Self-Selecting Team as a way to form a limited-cost team.

> Skunk Works → Self-Selecting Team → Diverse Groups → Unity of Purpose → Patron Role → Fire Walls → Gate Keeper → Compensate Success → Size The Organization → Phasing It In → Apprenticeship → Solo Virtuoso → Developing In Pairs → Holistic Diversity → Domain Expertise In Roles → Subsystem By Skill → Moderate Truck Number

Each application of a pattern is represented by a single node in this tree, and this node is divided into two other nodes: a node that denotes application of the next applicable pattern in the kick-off pattern sequence; a node that denotes that the next applicable pattern in the kick-off pattern sequence won't be applied. The node with the highest probability in the stochastic tree constructed for the whole kick-off pattern sequence is then the naturally expected pattern sequence candidate. Figure 1 shows the simplest example of a stochastic tree that can be used to identify expected pattern sequence candidates. Each node represents one pattern in the pattern sequence. Skunk Works, as the first pattern in the pattern sequence, can only be applied with the probability *1/N* where N stands for the number of patterns in the pattern language where this pattern is documented (there are 29 patterns in the Piecemeal Growth pattern language). If the next applicable pattern Self-Selecting Team is going to be applied, it can only be selected from $N - 1$ patterns in this language because Skunk Works is not going to be repeated in the expected pattern sequence candidate.

Each stochastic tree needs to consist of nodes representing patterns in pattern sequences, which it models, and probabilities of applying patterns represented by these nodes, if it is useful to identify the candidates for expected pattern sequences. Because each pattern can be either applied or rejected to be applied, a tree suitable for the identification of the expected pattern sequences can only be a binary tree. Because patterns are applied in sequences in consecutive order, links between nodes in these trees can only be unidirectional. No weights for the links are needed. Visualization, like in Fig. 1, allows its readers to identify how decisions to apply patterns in any pattern sequence modeled in the tree have an impact on the resulting probability of applying the pattern sequence that is modeled with the tree.

The stochastic tree used in the experiment with this method was constructed on top of the one kick-off pattern sequence established using all patterns from the Piecemeal Growth pattern language using explicit links between patterns found in the work of Coplien and Harrison (2004). We decided to model this pattern sequence with the tree to establish at least one of the expected pattern sequences from patterns in this language, and learn about the piecemeal growth of a working organization.[2]

To construct the stochastic tree, all organizational patterns documented in the Piecemeal Growth pattern language by Coplien and Harrison (2004) were used to establish one kick-off pattern sequence that was modeled with the stochastic tree. The pattern sequence of all patterns in the Piecemeal Growth pattern language, which we established and decided to model with the stochastic tree, can be described by a pattern story and, therefore, can be considered meaningful.[3] It was established following explicit links between its patterns. This kick-off pattern sequence helps to transform a small Skunk Works team into a working organization.

The node in the stochastic tree with the highest probability identified in the tree represented one expected pattern sequence candidate in the Piecemeal Growth pattern language. Node *p(Skunk Works → ¬Self Selecting Team → ¬Diverse Groups)* = 0.96429 in the stochastic tree represented the expected pattern sequence candidate. Not only one candidate node had to be chosen. If so, all these nodes will be expected pattern sequence candidates. Value 0.96429 is the probability of the use of the pattern sequence *Skunk Works → ¬Self Selecting Team → ¬Diverse Groups* by the assumed domain expert having access to the

[2] The stochastic tree used in experiments with this method is available at https://github.com/viktorFIIT/fiit-research-resources/blob/main/app/Use-On-Organizational-Patterns-1.md.

[3] The pattern story for this pattern sequence is available at https://github.com/viktorFIIT/fiit-research-resources/blob/main/stochastic-tree/pattern-stories/pattern-story-of-all-patterns.pdf.

Piecemeal Growth pattern language in the work of Coplien and Harrison (2004). The pattern sequence represented by this node with the highest probability is a sub-sequence of the kick-off pattern sequence of all patterns in the Piecemeal Growth pattern language, established from patterns documented in the work of Coplien and Harrison (2004). It is reasonable to expect this sequence because it can be used to shift a limited-cost team into a full-fledged software project team. Whether this node truly represents the expected pattern sequence had to be verified.

Hazen's stochastic trees (1992) can be constructed on top of any deliberately chosen pattern sequences or sequences extracted from pattern stories. Each pattern language can be used to construct at least one stochastic tree. The number of stochastic trees built on top of the pattern language depends on the number of sequences that are possible to establish with patterns in this language. The kick-off pattern sequence to be modeled with the stochastic tree can be established by following explicit links between patterns. The kick-off pattern sequence should consist of at least two patterns to successfully construct a stochastic tree. The type of relationship between patterns in the kick-off pattern sequence modeled in the stochastic tree does not matter. The kick-off pattern sequences established according to step 1 must consist of at least two patterns to be able to construct at least one stochastic tree. The minimum number of patterns used in the expected or unexpected pattern sequences will be equal to the number of patterns in the shortest expected pattern sequence candidate.

The stochastic tree speeds up the search for the expected pattern sequences because the expected pattern sequence candidate can be identified as a true expected pattern sequence. The stochastic tree may serve as the source of conditional probabilities that can be used in the calculation of the symmetry of relationships between patterns (in step 5 of this method) to establish meaningful pattern sequences. Currently, no similar technique capable of identifying pattern sequences with the highest probability of being expected is known to the authors of this work.

Patterns used in the candidates for the expected pattern sequences are systematically checked by constructing maps of other applicable patterns after them in the next section. We construct these maps to help us identify which strengths of symmetries of relationships we need to calculate if the assumption is that the strongest symmetries of relationships between patterns in candidate pattern sequences and other applicable patterns point to those patterns that are meaningful and expected to be applied next.

3.2 Construction of pattern maps

For each organizational pattern used in the expected pattern sequence candidate identified in the stochastic tree constructed in Sect. 3.1, searching for other patterns that can be used next was necessary to establish pattern sequences based on the strongest symmetries of relationships between patterns.

Stochastic trees cannot be used to identify relationships between patterns in the candidate for the expected pattern sequence and other applicable patterns. Therefore, the simplest visualization technique, where each pattern from the candidate pattern sequence would be linked to another applicable pattern, was employed. Visualizing other applicable patterns with links between patterns in pattern maps of applicable patterns is a similar technique to the use of Alexandrian pattern graphs in the work of Sousa et al. (2022) and one of the simplest visualizations we could use to systematically identify strongest symmetries of relationships between patterns in the candidate pattern sequence and other applicable patterns.

To systematically identify strongest symmetries of relationships between patterns from the stochastic tree and other applicable patterns, we constructed a directed graph for each pattern in the candidate for the expected pattern sequence, where the source of all directed edges is a vertex representing pattern from expected pattern sequence candidate (Skunk Works, Self-Selecting Team, and Diverse Groups), and edges point from this source vertex to other vertices representing patterns expected to be used next after the pattern represented by the source vertex. A bidirectional edge pointing toward the central vertex was created if another pattern was stated as applicable before the pattern represented by the central vertex in its text description.

Search for the next applicable patterns was conducted between patterns documented in the work of Coplien and Harrison (2004). If the use of another pattern linked by the pattern used in the expected pattern sequence candidate was not prohibited from being used in its text description by the author of this pattern, an explicit link to this other pattern was interpreted as pointing to the next applicable pattern. The text description of the pattern referenced by the pattern in the expected pattern sequence candidate was checked to see if it allowed the use of the pattern in the expected pattern sequence candidate, and if its use was allowed, the bidirectional arrow was created.

Maps of applicable patterns do not primarily visualize explicit or implicit relationships between patterns. Patterns for which pattern maps of applicable patterns were constructed can have explicit or implicit relationships with patterns expected to be used next.

Patterns that are not allowed to be used next after the pattern for which the pattern map of applicable patterns was constructed by the authors of the text descriptions of these patterns were not visualized in these maps. Because of this, patterns that are not allowed to be used will not be part of the most expected, expected, or unexpected pattern sequences.

Patterns that are allowed to be used next after the pattern for which the pattern map of applicable patterns was constructed by the author of its text description were visualized in pattern maps. Patterns in the most expected, expected, and unexpected pattern sequences are those patterns that are visualized in pattern maps of applicable patterns and for which the symmetry of the relationship is calculated.

The application of the step 4 of this method can be seen in Figs. 2, 3 and 4, which present patterns documented by Coplien and Harrison (2004) which are applicable after or before patterns used in the expected pattern sequence candidate identified in Sect. 3.1. Examples of the pattern maps of applicable patterns can be seen in Figs. 2, 3 and 4, where a central vertex represents a pattern linking to other patterns documented in the work of Coplien and Harrison (2004).

Central vertices in Figs. 2, 3 and 4 represent Skunk Works, Self-Selecting Team, and Diverse Groups organizational patterns from the expected pattern sequence candidate. Edge pointing onward from the central vertex in the pattern map of applicable patterns that can be seen in Figs. 2, 3 and 4 points to the pattern expected to be used after the pattern represented by the central vertex. The edge pointing toward the central vertex in the pattern map of applicable patterns represents the pattern expected to be used before the pattern represented by the central vertex.

Because the first expected pattern sequence candidate Skunk Works → ¬ Self-Selecting Team → ¬ Diverse Groups from the kick-off pattern sequence of all patterns in the Piecemeal Growth pattern language starts with Skunk Works organizational pattern, a pattern map of applicable patterns before or after Skunk Works in Fig. 2 was constructed.

According to Coplien and Harrison (2004), the Skunk Works organizational pattern can be used to create a temporary team to test the capability of incorporating innovative technology into already existing software products. In the work of Coplien and Harrison (2004), six patterns linked by Skunk Works in Fig. 2 can be used after Skunk Works and

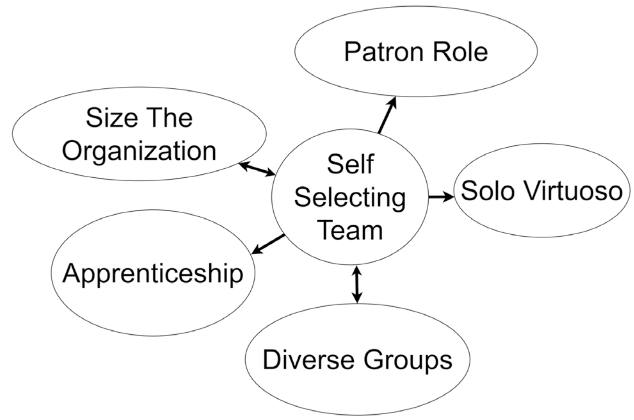

Fig. 3 The pattern map of the patterns applicable after Self-Selecting Team

are documented in the Piecemeal Growth pattern language. Figure 2 also identifies the direction in which Skunk Works links to other patterns in the Piecemeal Growth pattern language, whether these links are unidirectional or bidirectional. For example, the unidirectional edge between Skunk Works and Fire Walls expresses that Fire Walls can be used after Skunk Works to isolate the Skunk Works team from the rest of the organization. On the other hand, the bidirectional edge between Skunk Works and Patron Role expresses that a leading role in the organization can create a Skunk Works team, and the Skunk Works team can also be promoted to a project team by Patron Role. Probabilities of the subsequent use of patterns before and after Skunk Works are calculated and provided in Sect. 3.3.

According to Coplien and Harrison (2004), the Self-Selecting Team can be used to choose members of the Skunk Works team based on their track record and broader interests. The Self-Selecting Team organizational pattern from the expected pattern sequence candidate in Fig. 3 links five

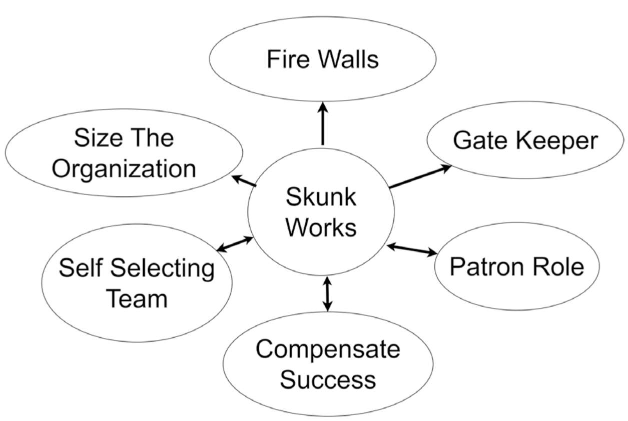

Fig. 2 The pattern map of the patterns applicable after Skunk Works

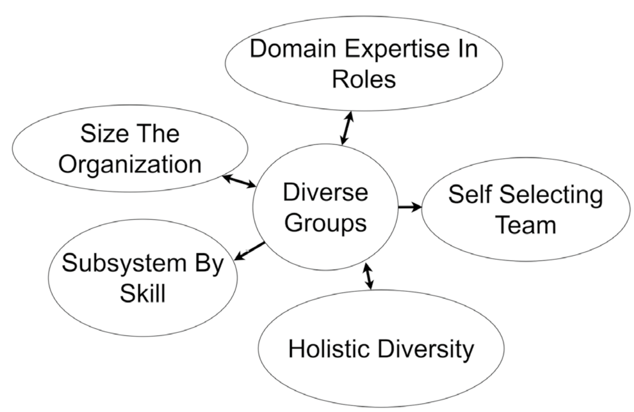

Fig. 4 The pattern map of the patterns applicable after Diverse Groups

other organizational patterns that can be used after the Self-Selecting Team, while only two of these patterns refer to Self-Selecting Team. The pattern map of applicable patterns for the Self-Selecting Team can be seen in Fig. 3.[4]

The last organizational pattern in the expected pattern sequence candidate is Diverse Groups. According to Coplien and Harrison (2004), the Diverse Groups can be used to choose members of the Skunk Works team based on their temperaments. The Diverse Groups links to five other organizational patterns that can be used after the Diverse Groups. Self-Selecting Team does not refer to and is not expected to be used before the Diverse Groups. Figure 4 below provides a pattern map of applicable patterns before or after the Diverse Groups.

3.3 Probability of the subsequent use of patterns

The probabilities of the subsequent use of patterns referenced by those represented by central vertices in Figs. 2, 3 and 4 were extracted from the stochastic trees constructed in Sect. 3.1 according to step 5 of this method to identify patterns which can be applied before or after those in candidate pattern sequences identified in Sect. 3.1 and still establish meaningful pattern sequence. The probabilities opposite to them were calculated with Bayes' rule.

To understand how the conditional probability of the use of patterns applicable after the central pattern in the pattern maps in Figs. 2, 3 and 4 can be calculated, consider that the probability of the subsequent use of the Size The Organization pattern after Skunk Works can be calculated as $1/(29-8)$ because there are 29 patterns in the Piecemeal Growth pattern language, and eight patterns were applied before the Size The Organization pattern in the kick-off pattern sequence presented in Sect. 3.1. This can be generalized and the probability of applying each applicable pattern in each of the pattern maps in Figs. 2, 3 and 4 can be calculated as $1/(N-M)$ where N is the number of patterns user works with and M is the number of patterns applied in kick-off pattern sequence before the next applicable pattern. The probability of applying a pattern like this is then calculated in the same way as in a stochastic tree.

If the conditional probability of applying the next applicable pattern, which is Size The Organization, after Skunk Works (and this applies to any pattern in the candidate pattern sequence as well) can be calculated as $p(STO|SW) = 1/(M-N)$, then the opposite probability to it can be calculated with Bayes' rule as $p(SW|STO) = (p(STO|SW) * p(SW))/((p(STO|SW) * p(W))+ p(STO|\neg SW) * p(\neg SW))$ where:

- The probability of applying Skunk Works $p(SW)$ can be calculated by dividing the chance of using Skunk Works by the number of patterns in its pattern language $p(SW) = 1/29$.
- The probability of applying Size The Organization, knowing that Skunk Works will not be applied before it can be calculated by ignoring Skunk Works in the kick-off pattern sequence $p(STO|\neg SW) = 1/(29-7)$.
- The probability of not applying Skunk Works is the complement to the probability that it will be applied and is equal to $p(\neg SW) = 28/29$.

The absolute difference between the probability of using a pattern from the expected pattern sequence candidate before and after the pattern it refers to was calculated for pattern maps of applicable patterns in Figs. 2, 3 and 4. All these differences between probabilities were sorted in ascending order and attached to Figs. 2, 3 and 4 to identify patterns that can be used one after the other and still establish meaningful pattern sequences. This resulted in creating three ordered lists for three pattern maps of applicable patterns. Absolute differences between conditional probabilities calculated for each relationship that were visualized in pattern maps in Sect. 3.2 and opposite probabilities help to compare the positive numbers in the search for patterns which can be used one after another and still establish meaningful pattern sequences.

3.4 Strength of the symmetry of relationships between organizational patterns

To establish meaningful expected pattern sequences, we identified the lowest difference from the list of absolute differences (created in Sect. 3.3) between probabilities of the subsequent use of patterns from expected pattern sequence candidates before and after the patterns they refer to, and used this difference as an implicit relationship to establish meaningful pattern sequences. A list of these differences was created for each pattern map of applicable patterns in Figs. 2, 3 and 4 during the execution of step 5 of this method to identify patterns expected to be applied next after central patterns in these maps. It was identified that the smallest absolute differences point to patterns that establish meaningful pattern sequences.

We define the absolute difference between conditional probabilities of subsequent use of patterns A and B: $|p(A|B) - p(B|A)|$ as the symmetry of relationships between patterns A and B. This definition of the symmetry of relationships between two patterns is similar to the definition of asymmetry in pattern adoption provided by Sousa et al. (2022), with the difference that the symmetry defined here takes only positive values which helps to compare only positive values and was shown to always identify patterns which

[4] The position of the vertices in pattern maps of applicable patterns is arbitrary.

establish meaningful pattern sequence even if their order is exchanged.

The strongest symmetry of the relationship between pairs of patterns A and B in pattern maps of applicable patterns in Figs. 2, 3 and 4 was found, where the lowest absolute difference $|p(A|B) - p(B|A)|$ of conditional probabilities of their subsequent use was calculated in the list attached to each pattern map of applicable patterns.[5] Step 5 of this method calculates the strength of symmetry of the relationship between patterns using conditional probabilities extracted from those stochastic trees constructed in step 2, which contain these conditional probabilities.

The symmetry of the relationships between the pattern represented by the central vertex in Fig. 2 and other patterns it refers to allowed us to identify the pattern expected to be used after the Skunk Works in the pattern sequence that is expected to be used.

Because stochastic trees cannot guarantee meaningfulness of the highest probable pattern sequence and to prove that the strongest symmetries of relationships help to establish meaningful pattern sequences, we checked both subsequent patterns in the candidate for the expected pattern sequence Skunk Works → ¬ Self-Selecting Team → ¬ Diverse Group identified in the stochastic tree in Sect. 3.1 to see if they have the strongest symmetry of relationships. The strongest symmetry of the relationships between the pattern represented by the central vertex in Fig. 2 and other patterns it refers to allows us to identify the pattern expected to be used the most after Skunk Works in the most expected pattern sequence.

According to the list of differences between conditional probabilities created for pattern map of applicable patterns of Skunk Works in Fig. 2, of all the links to patterns applicable after or before Skunk Works in Fig. 2, difference between probabilities of using Skunk Works before or after the Self-Selecting Team is the lowest one. Symmetry of the relationship between Skunk Works and Self-Selecting Team could be calculated as absolute difference between conditional probabilities: $|p(Self\ Selecting\ Team \mid Skunk\ Works) - p(Skunk\ Works \mid Self\ Selecting\ Team)| = 0.03571 - 0.03329 = 0.00242$, which was also the strongest symmetry because of the lowest value for all patterns Skunk Works links to. Using this difference between probabilities, it was possible to consider Skunk Works and Self-Selecting Team as organizational patterns with the strongest symmetric relationship that can be used one after another because Self-Selecting Team can be used to let people select their own Skunk Works team (Self-Selecting Team → Skunk Works) and Skunk Work teams can also naturally grow by selecting new members based on common interests (Skunk Works → Self-Selecting Team).

The implicit relationship represented as the strength of symmetry between Skunk Works and Self-Selecting Team organizational patterns is also explicit because Skunk Works links to the Self-Selecting Team in its text description.

The strongest symmetry of relationships was also identified between the Self-Selecting Team and Diverse Groups. The Diverse Groups organizational pattern can be used after the Self-Selecting Team to enrich the screening process by checking temperaments and diverse experience backgrounds of potential candidates. Whether the check for common interests or temperaments will be conducted first clearly does not matter, but the strength of symmetry of the relationship between the Self-Selecting Team and Diverse Groups tells us that order of use of these two patterns in pattern sequence can be reversed and still establish the most expected and meaningful pattern sequence Skunk Works → Self-Selecting Team → Diverse Groups.

The strengths of the symmetries of relationships between Skunk Works, Self-Selecting Team, Diverse Groups, and other patterns they relate to are available in the repository along with the complete calculations.[6]

3.5 Weak symmetries of relationships between organizational patterns

Weak relationships are formed between patterns with weaker symmetric relationships. The method for establishing expected pattern sequences based on stochastic trees also produces a set of unexpected pattern sequences established using the weakest symmetries of the relationships between software patterns. Pattern users should avoid using unexpected pattern sequences.

Looking at the strengths of symmetries of relationships of Skunk Works to other applicable patterns provided in the repository, it can be seen that Skunk Works has the lowest strength of the symmetry of the relationship (highest symmetry value) with the Size The Organization because the difference $|p(Skunk\ Works|Size\ The\ Organization) - p(Size\ The\ Organization|Skunk\ Works)|$ is the highest one. This is reasonable because Skunk Works may not transform into the regular project team through the application of the Size The Organization. The Skunk Works team can be dissolved after achieving its goal.

The weakest strength of symmetry of the relationship was also found between the Self-Selecting Team and the Solo Virtuoso organizational patterns. The Solo Virtuoso organizational pattern can be used to assign an effective software

[5] Lists can be seen here: https://github.com/viktorFIIT/fiit-research-resources/blob/main/app/Use-On-Organizational-Patterns-1.md.

[6] https://github.com/viktorFIIT/fiit-research-resources/blob/main/app/Use-On-Organizational-Patterns-1.md.

engineer to the task if assigning more professionals would not speed up progress. The Self-Selecting Team organizational pattern helps select compatible team members for the team project, and there is no relation to the conflicting forces that Solo Virtuoso tries to resolve. The weakest symmetry of relationship points here to the patterns that do not establish meaningful pattern sequences, even if the order of their subsequently used patterns is reversed.

3.6 Verifying the usefulness of the patterns in candidate pattern sequence with the maximum-likelihood function

Before it can be stated that the expected pattern sequences are established from useful patterns, verification of the usefulness of the pattern from the expected pattern sequence candidate may be performed to remove the subjective assumption in the usefulness of this pattern provided by its author in the form of a confidence level placed in its text description.

The likelihood of the usefulness of any software pattern with a confidence level can be calculated with the maximum-likelihood function. This likelihood can then be compared to the numerically represented confidence level provided by the author of the text description of a pattern (for example, to the confidence levels provided by Coplien and Zhao 2000).

The likelihood of usefulness of the pattern used in step 4 of this method was calculated by multiplying the complement values to the conditional probabilities that patterns in the pattern map of applicable patterns constructed in step 4 of this method would not be used after the pattern represented by the central vertex in this map. These conditional probabilities for which complements were used were calculated using the process described in Sect. 3.3. The usefulness of the pattern used in step 4 can be verified if the output value from the maximum likelihood function is higher than the numerically represented confidence level for this pattern.

Confidence levels for patterns documented by Coplien and Harrison (2004) consist of zero to two asterisks. If we want to represent this numerically to be able to compare it to the output of the maximum-likelihood function, we can divide the interval $0 - 100$ into chunks based on the distribution of these asterisks:

- If there is no asterisk provided as the confidence level, this means the author is 33% sure that his pattern is useful.
- If there is one asterisk provided as the confidence level, this means the author is 66% sure that his pattern is useful.
- If there are two asterisks provided as the confidence level, the author is more than 99% sure that his pattern is useful.

Here a verification of the usefulness of the organizational pattern is reported which works by calculating conditional probabilities that the patterns from Figs. 2, 3 and 4 linked by the patterns represented by central vertices in these maps would not be used and inserting them into the maximum-likelihood function as is shown in the Sect. 3.6.

The maximum likelihood function multiplies all probabilities together for each pattern map of applicable patterns to output the likelihood of the usefulness of the pattern represented by the central vertex in the pattern map of applicable patterns.[7] Only patterns represented by central vertices in Figs. 2, 3 and 4 were used to calculate these likelihoods. This means that the usefulness of only Skunk Works, Self-Selecting Team, and Diverse Groups was verified with the maximum-likelihood function. The idea behind this verification technique is that the surroundings of each organizational pattern tell more about its usability than this pattern alone.

The first pattern whose usefulness was necessary to verify was Skunk Works from the expected pattern sequence candidate. To verify the usefulness of Skunk Works, Self-Selecting Team, and Diverse Groups, complements to conditional probabilities of subsequent use of the patterns linked by these patterns, as depicted in Figs. 2, 3 and 4, had to be calculated. The complements to these conditional probabilities were multiplied (the maximum likelihood function is a multiplication of the components $(1 - \theta)$), which resulted in calculating the likelihood of usefulness of Skunk Works in Sect. 3.6. The verification of the usefulness of the Skunk Works organizational pattern was successful. The confidence level of Skunk Works provided in the work of Coplien and Harrison (2004) reflects this confidence in the usefulness of this pattern similarly.

$$
\begin{aligned}
&p(\mathit{Skunk\ Works}|\theta_1,\theta_2,\theta_3,\theta_4,\theta_5,\theta_6)\\
&\quad = p(\neg \mathit{Fire\ Walls}|\theta_1) * p(\neg \mathit{Gate\ Keeper}|\theta_2)\\
&\qquad p(\neg \mathit{Patron\ Role}|\theta_3) * p(\neg \mathit{Compensate\ Success} \mid \theta_4)\\
&\qquad * p(\neg \mathit{Self\ Selecting\ Team} \mid \theta_5)\\
&\qquad p(\neg \mathit{Size\ The\ Organization} \mid \theta_6)\\
&\quad = 0.95833 * 0.95652 * 0.96\\
&\qquad * 0.95455 * 0.96429 * 0.95238 = 0.77143
\end{aligned}
$$

Because the likelihood of the usefulness of Skunk Works is slightly higher than the numerically represented confidence level ($0.77143 > 0.66$), it can be concluded that this likelihood is sufficiently high. Because of the number of patterns in the Piecemeal Growth pattern language, the probability of the subsequent use of patterns is small, and multiplying them together produces even smaller numbers. This was

[7] The maximum likelihood function is available at https://github.com/viktorFIIT/fiit-research-resources/tree/main/appendices.

the reason behind accepting the use of this pattern in the expected pattern sequence.

3.7 Establishing expected pattern sequences

The strength of the symmetry of the relationship between patterns was used to identify patterns expected to be applied, which resulted in establishing meaningful pattern sequences. For each pattern sequence, a pattern story was created to check its meaningfulness. All sequences of patterns established this way had to be evaluated first with respect to the following two properties, which can be naturally expected to be true for any most expected pattern sequence:

- Any two subsequently used patterns in the sequence have the strongest symmetries of relationships because the existence of this implicit relationship assures that patterns establish meaningful pattern sequences. This means there must *not* be another applicable pattern with which the pattern established in the expected pattern sequence has the strongest symmetry of relationship; otherwise, it would have to be part of this pattern sequence, and the expected pattern sequence would consist of different patterns.
- The probability of the use of the sequence of patterns $X_1 ... X_N$, calculated as $p(X_2|X_1) * p(X_3|X_2) ... p(X_N|X_{N-1})$, is higher than the cumulative probability that patterns in these sequences would be used outside of the sequence, calculated as $p(X_1) * p(X_2) * ... * p(X_N)$. If it were more probable that the patterns in the sequence would be used individually, they could not establish an expected pattern sequence.

Based on the strongest symmetries of the relationships between Skunk Works and other patterns, the first most expected pattern sequence in the Piecemeal Growth pattern language is Skunk Works → Self-Selecting Team → Diverse Groups. The probability of applying a pattern sequence Skunk Works → Self-Selecting Team → Diverse Groups can be calculated as $p(\textit{Diverse Groups} \mid \textit{Self Selecting Team}) * p(\textit{Self Selecting Team} \mid \textit{Skunk Works}) = 0.03704 * 0.03571 = 0.001322$. The probability of applying patterns Skunk Works, Self-Selecting Team, and Diverse Groups independently is $p(\textit{Skunk Works}) * p(\textit{Self Selecting Team}) * p(\textit{Diverse Groups}) = (1/29) * (1/29) * (1/29) = 0.03448 * 0.03448 * 0.03448 = 0.00004099225$. The probability of using this sequence is higher than the probability of using its patterns outside of the sequence.

The stochastic tree constructed in Sect. 3.1 can model any possibly meaningless pattern sequence and point to it as the sequence with the highest probability. This is why the strength of symmetry of the relationship must be used to identify patterns expected to be applied next that establish a meaningful pattern sequence.

Expected pattern sequences do not differ from unexpected pattern sequences by the probability of their use. The most expected pattern sequences can be differentiated from expected pattern sequences and unexpected pattern sequences using the strengths of symmetries of relationships between their patterns. The most expected pattern sequences are established following the strongest symmetries of relationships between patterns. Strongest symmetries of relationships between patterns make it possible to distinguish pattern sequences that are likely to be used (their probability of use is always higher than the use of their patterns outside the sequence) and that are always meaningful. Stochastic processes can be used to identify pattern sequences that are likely to be used. The problem is that they can calculate the same probability of use for meaningful and non-meaningful pattern sequences. The use of the strengths of symmetries of relationships between patterns solves this problem. Expected pattern sequences established by following weaker strengths of symmetries of relationships between patterns have a higher probability of use than the probability of use of their patterns outside the sequence. Because these are based on weaker symmetries, they may require applying more patterns (because they solve problems partially), but they always stay meaningful. Unexpected pattern sequences that are established by following the weakest strengths of symmetries of relationships between patterns are always non-meaningful, regardless of the probability of their use.

4 Establishing expected pattern sequences based on Bayesian networks

Another way to establish pattern sequences is with the help of Bayesian networks presented by Barber (2011), probabilistic reasoning, and graphical networks based on the Bayesian theorem (Zhang et al. 2019). Bayesian networks can be used to identify conditionally dependent patterns that stochastic trees are incapable of doing. Bayesian belief networks can be used to calculate the likelihood of the existence of implicit relationships between patterns, which serves as the basis for the expected pattern sequence, and at the same time to calculate this likelihood considering conditional dependencies between patterns, which can be extracted from their text descriptions.

Bayesian networks consist of vertices that can represent events like applications of patterns. Vertices in these networks are connected through directed arcs, which can represent causal relationships between patterns if their text descriptions link to each other, and if their subsequent use is allowed in these text descriptions.

The input to the method we propose here is a set of text descriptions of patterns. Text descriptions of patterns used in this method do not have to adhere to any pattern forms commonly used for documenting patterns. The output is a set of expected pattern sequences and a set of unexpected pattern sequences. Expected pattern sequences are recommended, while unexpected ones are to be avoided. The method consists of these steps:

1. Establish all meaningful kick-off pattern sequences using all patterns in the pattern language such that these sequences consist of at least two unique patterns using explicit or implicit relationships between patterns in this pattern language.
2. Construct stochastic trees of pattern relationships on top of all kick-off pattern sequences established in step 1.
3. For each kick-off pattern sequence established in step 1, choose any pattern expected to be used in them that is not already used in this kick-off pattern sequence.
4. For each kick-off pattern sequence established in step 1, identify the pattern expected to be used in this kick-off pattern sequence which you expect to be used *before* the pattern chosen in step 3.
5. For each kick-off pattern sequence established in step 1, identify the pattern expected to be used in this kick-off pattern sequence which you expect to be used *after* the pattern chosen in step 3.
6. For each kick-off pattern sequence established in step 1, extract conditional probability that the pattern chosen in step 3 would be used after the pattern identified in step 4 and extract this probability from the stochastic tree constructed in step 2 which contains this probability. Calculate the opposite probability to the probability $p(\mathit{pattern\ before}|\mathit{chosen\ pattern})$ with Bayes' rule as $p(\mathit{chosen\ pattern} \mid \mathit{pattern\ before}) = (p(\mathit{pattern\ before} \mid \mathit{chosen\ pattern}) * p(\mathit{chosen\ pattern})) / ((p(\mathit{pattern\ before} \mid \mathit{chosen\ pattern}) * p(\mathit{chosen\ pattern})) - -p(\mathit{pattern\ before} \mid \neg \mathit{chosen\ pattern}) * p(\neg \mathit{chosen\ pattern}))$, where the prior probability $p(\mathit{chosen\ pattern})$ is calculated as the $1/N$ where N is number of patterns user works with. . Calculate the absolute difference between these two probabilities.
7. For each kick-off pattern sequence established in step 1, extract conditional probability that the pattern chosen in step 3 would be used before the pattern identified in step 5 and extract this probability from stochastic tree constructed in step 2 which contains this probability. Calculate the opposite probability to this probability with Bayes' rule as $\mid \mathit{pattern\ after}) = (p(\mathit{pattern\ after} \mid \mathit{chosen\ pattern}) * p(\mathit{chosen\ pattern}))/((p(\mathit{pattern\ after} \mid \mathit{chosen\ pattern}) * p(\mathit{chosen\ pattern})) - p(\mathit{pattern\ after} \mid \neg \mathit{chosen\ pattern}) * p(\neg \mathit{chosen\ pattern}))$, where the prior probability $p(\mathit{chosen\ pattern})$ is calculated as the $1/N$ where N is number of patterns user works with. Calculate the absolute difference between these two probabilities.
8. For each kick-off pattern sequence established in step 1, extract conditional probability that the pattern identified in step 5 would be used after the pattern identified in step 4 and extract this probability from the stochastic tree constructed in step 2 containing this probability. Calculate the opposite probability to this probability with Bayes' rule. Calculate the absolute difference between these two probabilities.
9. In each kick-off pattern sequence established in step 1, if both calculated strengths of symmetry calculated in steps 6–7 are lower than the strength of symmetry calculated in step 8, then pattern chosen in step 3 *is expected* to be used after the pattern identified in step 4 and before the pattern identified in step 5 in this kick-off pattern sequence. Calculating the strength of symmetry of a relationship like this allows us to use it to establish patterns into likely and meaningful pattern sequences without using statistics about the past use of these patterns in practice. The expected pattern sequence is established starting with the application of the pattern identified in step 4, continuing with the application of the pattern chosen in step 3, and ending with the application of the pattern identified in step 5. Add this expected pattern sequence to the set of expected pattern sequences.
10. In each kick-off pattern sequence established in step 1, if at least one strength of symmetry calculated in steps 6–7 is not lower than the strength of symmetry calculated in step 8, then pattern chosen in step 3 *is not expected* to be used after the pattern identified in step 4 and before the pattern identified in step 5 in this kick-off pattern sequence. Unexpected pattern sequence is established starting with the application of the pattern identified in step 4, continuing with the application of the pattern chosen in step 3, and ending with the application of the pattern from step 5. Add this expected pattern sequence to the set of unexpected pattern sequences.
11. If the *expected* pattern sequence is established in step 9, use the Bayesian network to calculate the likelihood that the pattern chosen in step 3 would be used after the pattern identified in step 4 and before the pattern identified in step 5 in the expected pattern sequence established in step 9. If the unexpected pattern sequence is established in step 10, do not use the Bayesian network to calculate the likelihood that the pattern chosen in step 3 would be used after the pattern identified in step 4 and before the pattern identified in

step 5 in the unexpected pattern sequence established in step 10.

12. Identify another pattern that is not already used in one of the kick-off pattern sequences established in step 1 and that you expect to apply between any two patterns that are already used in one of the kick-off pattern sequences, such that at least one of these two already used patterns is different from those previously identified in steps 4–5. The pattern that you choose can be another pattern from the pattern languages used by the user of this method, which they wish to use in the pattern sequence. Continue to step 4 with the identified pattern that is not already used in all kick-off pattern sequences. If another pattern to be included cannot be identified by the user of this method, then all expected and unexpected pattern sequences are stored in the set of expected and unexpected pattern sequences from the iterations of this method, and the application of steps 1–12 in these iterations is finished. If the established pattern sequences are not satisfactory for the resolution of a problem of the user of this method, and the user of this method still has new patterns that he wishes to apply, this method can be applied again. If the established pattern sequences are not satisfactory and no new patterns are available, there is no reason to reapply this method, as it is designed to establish meaningful pattern sequences from patterns with existing text descriptions.

Barber (2011) uses a graphical representation of a Bayesian network to support calculations because it allows visualization of direct and indirect relationships between events, such as the application of organizational patterns. However, the Bayesian network is used in this method to calculate the likelihood of using a pattern from the Organizational Style pattern language between two surrounding patterns from the Piecemeal Growth pattern language.

Drawing of the Bayesian network was avoided in experiments with this method because its visualization is not needed to establish expected pattern sequences. The Bayesian networks are used here to calculate the likelihood of using expected pattern sequences from the Piecemeal Growth and Organizational Style pattern language.

The expected sequences established using this method are based on the strongest symmetries of relationships between their patterns.

Using at least two patterns in the kick-off pattern sequence in step 1 of this method results in establishing either one expected or unexpected pattern sequence. The number of patterns in the kick-off pattern sequences established in step 1 of this method determines the number of established expected and unexpected pattern sequences.

4.1 Establishing the kick-off pattern sequence

The first method presented in Sect. 3 requires the construction of a new stochastic tree if patterns from multiple pattern languages or catalogs are meant to be established in the most expected or expected pattern sequence. Redesigning this method resulted in designing another method that allows the combination of patterns from multiple pattern languages using the strongest symmetry of the relationship between them, which does not require the construction of a stochastic tree for each of these languages, and produces a likelihood of the resulting pattern sequence calculated considering conditional dependencies between patterns. This kick-off pattern sequence was established by chaining all organizational patterns in the Piecemeal Growth pattern language in order that they are described by Coplien and Harrison (2004) and by introducing one organizational pattern from the Organizational Style pattern that is linked by at least one pattern in the sequence of the Piecemeal Growth pattern language. These pattern languages were chosen to learn more about the piecemeal growth of the organization and how its initial structure unfolds as well.

Consider this example kick-off pattern sequence:

Skunk Works → Self-Selecting Team → Diverse Groups → Unity of Purpose → Patron Role → Size The Organization → Phasing It In → Apprenticeship → Solo Virtuoso → Developing In Pairs → Holistic Diversity → Domain Expertise In Roles → **Responsibilities Engage** → Subsystem By Skill → Moderate Truck Number → Application Design Is Bounded By Test Design → Group Validation → Engage Quality Assurance → Scenarios Define Problem → Engage Customers → Fire Walls → Gate Keeper → Team Pride → Legend Role → Matron Role → Public Character → Wise Fool → Compensate Success → Failed Project Wake → Surrogate Customer

Apart from the Responsibilities Engage pattern from the Organizational Style pattern language, this kick-off pattern sequence consists only of the patterns from the Piecemeal Growth pattern language. This kick-off pattern sequence describes the construction process of an organization from scratch, because that is Piecemeal Growth pattern language is all about.

Establishing the first most expected pattern sequence started according to step 3 of this method with a question of whether the Responsibilities Engage organizational pattern is expected to be used directly after the Domain Expertise In Roles and before Subsystem By Skill. To establish the most expected pattern sequence, the strength of the symmetry of relationships between the Responsibilities Engage and both surrounding patterns in the kick-off pattern sequence had to be compared. If the strengths of the symmetry of

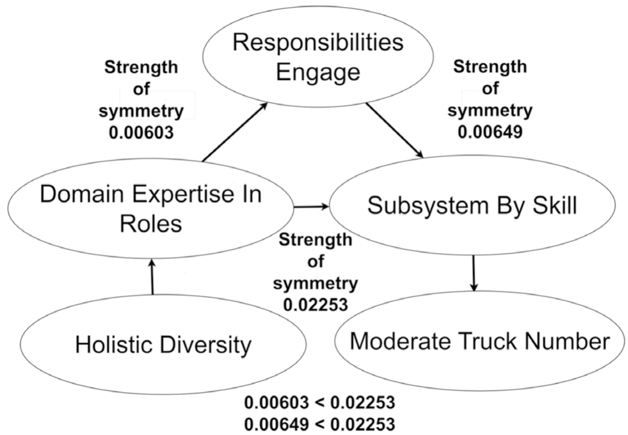

Fig. 5 The first most expected pattern sequence

the relationships of the patterns from the Organizational Style pattern language to two organizational patterns from the Piecemeal Growth pattern language were found to be stronger than the strength of the symmetry of the relationship between two subsequently applied patterns from the Piecemeal Growth pattern language, the most expected pattern sequence with three patterns (two surrounding ones and one included in-between) could be established.

The likelihood of the need to resolve forces between organizational patterns from the Piecemeal Growth pattern language by including organizational patterns from the Organizational Style pattern language was calculated using the Bayesian network.

4.2 Establishing expected pattern sequences

The Bayesian networks could be employed to calculate the likelihood of the existence of the implicit relationship between Domain Expertise In Roles and Self-Selecting Team from the Piecemeal Growth pattern language and Responsibilities Engage organizational pattern from the Organizational Style pattern language, given that this relationship serves as a precondition for the application of Subsystem By Skill or Diverse Groups organizational pattern.

After calculating the strength of symmetry of the relationships between Domain Expertise In Roles and Subsystem By Skill, and between Responsibilities Engage, Domain Expertise In Roles and Subsystem By Skill in Fig. 5, calculation of the likelihood of the existence of the implicit relationship between Domain Expertise In Roles and Responsibilities Engage, given this relationship serves as a precondition for applying Subsystem By Skill started.

As can be seen in Fig. 5, the strength of the symmetry of the relationship between Domain Expertise In Roles and the Responsibilities Engage organizational pattern was calculated. The strength of the symmetry of the relationship between Domain Expertise In Roles and Responsibilities Engage is calculated according to step 6 of this method because its value is the strength of symmetry of the relationship between the pattern chosen in step 3 and the pattern identified in step 4 of this method. Calculating the strength of symmetry of the relationship between two patterns like this is similar to calculating asymmetry in pattern adoption by Sousa et al. (2022).

As seen in Fig. 5, the strength of the symmetry of the relationship between Subsystem By Skill and Responsibilities Engage was calculated. The strength of the symmetry of the relationship between Subsystem By Skill and Responsibilities Engage is calculated according to step 7 of this method because its value is the strength of symmetry of the relationship between the pattern chosen in step 3 and the pattern identified in step 5 of this method.

As can be seen in Fig. 5, the strength of the symmetry of the relationship between Domain Expertise In Roles and Responsibilities Engage and the strength of the relationship between Subsystem By Skill and Responsibilities Engage is stronger than the strength of the symmetry of the relationship between Subsystem By Skill and Domain Expertise In Roles organizational patterns. The strength of the symmetry of the relationship between Domain Expertise In Roles and Subsystem By Skill was calculated according to step 8 of this method because this value represents the strength of symmetry of the relationship between the pattern identified in step 4 and the pattern identified in step 5 of this method.

Because of this, Responsibilities Engage is expected to be used after Domain Expertise In Roles and before Subsystem By Skill organizational patterns. As a result, the first expected pattern sequence Domain Expertise In Roles → Responsibilities Engage → Subsystem By Skill was established.

At the same time, the Subsystem By Skill organizational pattern is not expected to be used directly after the Domain Expertise In Roles organizational pattern in the pattern sequence of patterns from the Piecemeal Growth pattern language. A set of all unexpected pattern sequences established using this method is available in repository.[8]

The Responsibilities Engage can be used directly after the Domain Expertise In Roles and before Subsystem By Skill when there is a need to avoid burdening the communication in the software project team on project managers. Responsibilities Engage is more helpful in guiding the software development team to form communication centers organized around Domain Expertise In Roles.

Responsibilities Engage is expected to be used the most after Self-Selecting Team and before Diverse Groups

[8] https://github.com/viktorFIIT/fiit-research-resources/tree/main/bayesian-networks/unexpected-sequences.

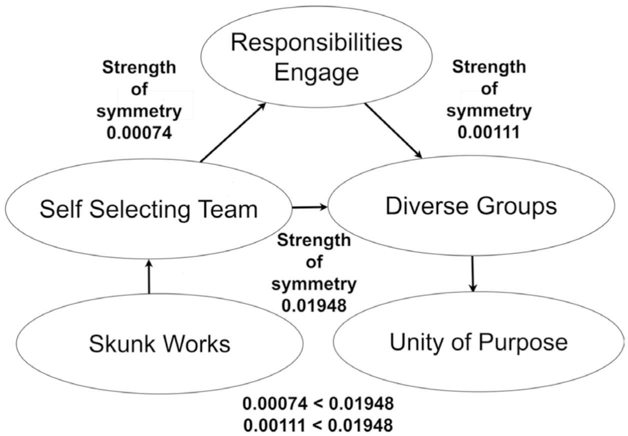

Fig. 6 The second most expected pattern sequence

because the strength of the symmetry of the relationship between Responsibilities Engage and Self-Selecting Team, and the strength of the symmetry of the relationship between Responsibilities Engage and Diverse Groups was found in Fig. 6 to be stronger than the strength of the symmetry of relationship between Self-Selecting Team and Diverse Groups.

As can be seen in both supportive calculations in the repository, Responsibilities Engage is expected to be used less likely between Domain Expertise in Roles and Subsystem By Skill organizational patterns, compared to the likelihood of use of Responsibilities Engage between Self-Selecting Team and Diverse Groups organizational patterns. Supporting likelihoods, calculated using Bayesian networks, are available in repository.[9]

It turns out that sequences of patterns established from patterns from one pattern language are more likely to be used than sequences established from patterns documented in multiple pattern languages. This can be presented as an example of pattern sequence Holistic Diversity → Domain Expertise In Roles → Responsibilities Engage → Subsystem By Skill → Moderate Truck Number, with probability of use 0.0000000577. The probability using another pattern sequence established solely from patterns in the Piecemeal Growth pattern language, Holistic Diversity → Domain Expertise In Roles → Subsystem By Skill → Moderate Truck Number is a bit higher: 0.0000107. When pattern authors are aware of the equal probability of use of all patterns in their pattern languages, they should divide them and create smaller pattern languages to establish their patterns in expected pattern sequences.

[9] https://github.com/viktorFIIT/fiit-research-resources/tree/main/bayesian-networks/supporting-likelihoods.

4.3 Establishing security in organization

The most expected sequence of organizational patterns Skunk Works → Self-Selecting Team → Diverse Groups established with the method of establishing expected pattern sequences presented in Sect. 3 can be combined with the most expected sequence of security patterns Access Control Requirements → Single Access Point → Security Session → Role-Based Access Control → Authorization established with the same method from security patterns of Cordeiro et al. (2022). This sequence of security patterns was established by applying Bayesian networks to identify conditionally dependent security patterns from the pattern catalog of Cordeiro et al. (2022) that can be established in pattern sequences by Matovič and Vranić (2024).

The resulting pattern sequence Skunk Works → Self-Selecting Team → Diverse Groups → Access Control Requirements → Single Access Point → Security Session → Role-Based Access Control → Authorization can establish a team of self-selected members (*Skunk Works → Access Control Requirements → Diverse Groups*) who try the new technology or create the first software prototype. Members of this team must provide access to the software to its external and internal users. To do that, they need to establish a security and access policy *Access Control Requirements*. Users will sign in through the login page *Single Access Point*. After they authenticate, a secure session *Security Session* will be established to recognize them and to securely communicate. Users of this software will be assigned the appropriate permissions based on the roles they need to act upon (*Role-Based Access Control*). Having these permissions assigned, they will be properly authorized (*Authorization*) to access the protected parts of the software.

5 Choosing the right method

The method of establishing pattern sequences using stochastic trees can be used for patterns for which textual descriptions are available. It can only be used if at least two pattern textual descriptions are available; otherwise stochastic tree cannot be constructed. This method produces a set of pattern sequences that are most expected to be used, expected to be used, and unexpected to be used in practice. This method combines frequentist and Bayesian approaches to calculating the probability of applying patterns while establishing pattern sequences. The frequentist statistical approach is embodied in calculating probabilities of using software patterns using the stochastic trees. The Bayesian statistical approach is embodied in calculating opposite probabilities to probabilities of using conditionally dependent patterns through Bayes' rule. The advantage of using this method lies in the fact that it enables its users to establish the most expected and expected to be used pattern sequences using explicit or implicit relationships between patterns, because it allows calculating strengths of symmetries of relationships even between implicitly related patterns. Patterns that do not have a documented explicit relationship in their textual descriptions can still be established in most expected or expected to be used pattern sequences using this method. The disadvantage of using this method stems from the fact that the user of this method must have textual descriptions of these patterns available to validate the usefulness of the established pattern sequences.

The strongest symmetries of the relationships between patterns help the method of establishing pattern sequences using stochastic trees to establish pattern sequences that stay meaningful even if the order of their subsequently used patterns is reversed. It can be shown that pattern sequences established using this method consist of shorter pattern sequences that are most expected and expected to be used, and that these pattern sequences have a higher probability of use than the probability of use of their patterns outside the pattern sequence. The symmetry of the relationship between patterns A and B calculated like $p(pattern\ A|pattern\ B) - p(pattern\ B|pattern\ A)$ in this method is just one of the distance metrics which can be used and other distance metrics described in Sect. 2 can also be used to establish pattern sequences expected to be used in practice.

Following the strongest symmetry between patterns by using the method of establishing pattern sequences using stochastic trees can lead the user of this method to establish the most expected to be used or expected to be used pattern sequence that is one of the kick-off pattern sequences if all reasonable pattern sequences are established from patterns, whose descriptions are available to user of this method. In this situation, it means this method would not be needed at all, but nobody can be sure beforehand if one of the kick-off pattern sequences established by the pattern user at the start of applying this method is based on the strongest symmetries of relationships between patterns. Using all patterns available to the user of this method to establish kick-off pattern sequences in this method mitigates the risk of not establishing all the most expected and expected to be used pattern sequences from these patterns. This method allows its user to establish all the most expected to be used and expected to be used pattern sequences from all patterns whose textual descriptions are available to its user. This mitigates the risk of resolving problems with solutions embodied in pattern sequences that are not expected to be used. If not all kick-off pattern sequences are established from all patterns available to the user of this method, there is a chance some of the most expected to be used and expected to be used pattern sequences would not be established.

The second method of establishing pattern sequences based on Bayesian networks can be used to establish the most expected to be used and unexpected to be used pattern sequences between patterns documented in multiple pattern languages or pattern catalogs without requiring user of this method to construct multiple stochastic trees, each for the kick-off pattern sequence established at the start of this method.

The method of establishing pattern sequences based on Bayesian networks requires fewer constructions of stochastic trees to establish the most expected to be used pattern sequences from patterns from multiple pattern languages or catalogs, compared to the number of required constructions by the first method of establishing pattern sequences based on stochastic trees.

The method of establishing pattern sequences based on Bayesian networks requires fewer calculations of conditional probabilities of subsequent use of patterns. The first method requires to calculate conditional probabilities of use of each software pattern applicable after the pattern used in kick-off pattern sequence (plus application of Bayes' rule for each probability) before the most expected to be used, expected to be used, and unexpected to be used pattern sequences can be established with it. Pattern sequences that are expected to be used can be established using the strengths of symmetries of relationships between patterns. The second method requires calculating three conditional probabilities of subsequent use of software patterns (plus three applications of Bayes' rule) before the most expected and unexpected pattern sequences can be established with it. The pattern sequences that are expected to be used can also be identified from a graphical visualization of the Bayesian belief network. Pattern sequences that are most expected to be used can be established using this method by employing rules from Barber (2011) or by analyzing a graphical visualization of the Bayesian network.

Pattern sequences that are expected to be used can be established with the method of establishing pattern sequences based on Bayesian networks from patterns between which the existence of the conditional dependence is confirmed. Unexpected pattern sequences to be used can be established with the method of establishing pattern sequences based on Bayesian networks between conditionally independent patterns. Use of the first method of establishing pattern sequences based on stochastic trees can result in establishing the most expected to be used and expected to be used pattern sequences from patterns between which conditional dependence is not confirmed, and which are only applicable together with a certain probability. Use of the method of establishing pattern sequences based on stochastic trees can result in establishing unexpected to be used pattern sequences between conditionally dependent patterns.

6 Discussion

It can be shown that the strongest symmetries of the relationships between patterns help to establish meaningful pattern sequences that stay meaningful even if the order of their subsequently used patterns is reversed. The symmetry of the relationship between any two software patterns A and B calculated as $|p(pattern\ A|pattern\ B) - p(pattern\ B|pattern\ A)|$, where $p(pattern\ A|pattern\ B)$ is calculated using stochastic tree in a way described in Sect. 3.1 and $p(pattern\ B|pattern\ A)$ is calculated using Bayes' rule in a way described in Sect. 3.3, is just one of the distance metrics which can be used and other distance metrics from the work of Koller and Friedman (2009) can be explored for their suitability to establish expected pattern sequences.

Relationships between patterns are not symmetrical all the time. Highest values of symmetries point to patterns that are not expected to be used in a sequence and that are asymmetric. Lowest values of symmetries point to patterns that are most expected to be used in the pattern sequence. We supported this finding by conducting an experiment, where 16 participants (18.8 % of them had a previous experience with organizational patterns) were interviewed on whether they found the most expected sequences of patterns meaningful and to propose a probability of their use. All 16 participants were able to describe these sequences in pattern stories. 43.8% of participants stated that the probability of use of one of these sequences: Skunk Works → Self-Selecting Team → Diverse Groups is between 0.76–1 (high probability), and 50 % of them stated that it is between 0.33–0.66 (moderate probability).

Following the strongest symmetry between patterns by using the method for establishing expected pattern sequences based on stochastic trees, can lead the user of this method to establish the most expected or expected pattern sequence that is one of the kick-off pattern sequences if all reasonable pattern sequences are established from patterns available to the user of this method. Pattern sequences are reasonable if they can be described in pattern stories like Buschmann et al. (2007) did. This means the method for establishing expected pattern sequences would not be needed at all, but nobody can be sure beforehand if one of the kick-off pattern sequences established by the pattern user is based on the strongest symmetries of relationships between patterns. Using all software patterns available to the user of this method to establish kick-off pattern sequences mitigates the risk of not establishing all the most expected and expected pattern sequences. This may result in resolving problems with nonoptimal solutions, as the pattern sequences that are neither most expected nor expected to be applied to solve these problems would be applied. If not all kick-off pattern sequences are established from all patterns in step 1 of the method for establishing expected pattern sequences based on stochastic trees, there is a chance that some most expected and expected pattern sequences would not be identified from expected pattern sequence candidates.

If kick-off pattern sequences are not established in step 1 of the method of establishing expected pattern sequences based on stochastic trees and in step 1 of the method of establishing expected pattern sequences based on Bayesian networks out of all patterns in the pattern language, less expected pattern sequence candidates will be identified from all stochastic trees compared to the case when all patterns from the pattern language are used. This can result in not establishing all expected pattern sequences from this pattern language, which can result in applying nonoptimal solutions by pattern users of these methods. Kick-off pattern sequences can also be extracted from pattern stories.

Neither the method of establishing expected pattern sequences based on stochastic trees presented in Sect. 3 nor the method of establishing expected pattern sequences based on Bayesian networks presented in Sect. 4 can be used if the pattern from the stochastic tree refers to other patterns not documented in the pattern language used to apply them. Looking at Figs. 2, 3 and 4, it would not be possible to calculate the conditional probability of using the pattern from another pattern language after the one used in the candidate pattern sequence. Text descriptions of those patterns not documented in the pattern language that the user of these methods wants to establish in expected pattern sequences must be retrieved. Access to text descriptions of all patterns users work with will help them validate the usefulness of the resulting expected pattern sequences and recognize their value in problem resolution. The users of the two methods presented in this article must understand the patterns from the pattern languages they work with to establish the kick-off pattern sequences and evaluate the meaningfulness of the resulting expected pattern sequences.

The method of establishing expected pattern sequences based on stochastic trees presented in Sect. 3 allows for the establishment of expected pattern sequences in which the use of patterns can be repeated, which is not allowed in the method of establishing expected pattern sequences based on Bayesian networks (presented in Sect. 4) because Bayesian networks are directed *acyclic* graphs. This can lead to establishing pattern sequences that provide nonoptimal solutions if the repeatable use of some pattern is needed for efficient problem resolution.

Expected pattern sequences can consist of shorter expected pattern sequences. These shorter pattern sequences may provide sufficient solutions to certain problems, and expected pattern sequences with more patterns applied may not be needed. The application of the method of establishing expected pattern sequences based on Bayesian networks to security patterns from the catalog of security patterns by Cordeiro et al. (2022) resulted in establishing this expected pattern sequence:

> Access Control Requirements → Single Access Point → Security Session → Role-Based Access Control → Authorization

which consists of these two shorter expected pattern sequences:

- Access Control Requirements → Single Access Point → Security Session
- Security Session → Role-Based Access Control → Authorization

Using a Bayesian network, it can be shown that there is a conditional dependency between any two subsequently applied patterns in the expected pattern sequence.

It can be shown by using the modus ponens rule from the work of Barber (2011), that if the expected pattern sequence consists of at least three patterns, the transitivity of relationships between patterns in this expected pattern sequence applies. This can be shown in the example of the expected pattern sequence Access Control Requirements → Single Access Point → Security Session, which was established using the method of establishing expected pattern sequences based on Bayesian networks. During the application of this method, it was found that if the application of Access Control Requirements is a precondition for the application of Single Access Point, and if the application of Single Access Point is a precondition for the application of Security Session, then the application of Access Control Requirements is a precondition for the application of Security Session. This transitivity assures users of these methods that expected pattern sequences expect pattern applications in the order specified by these sequences.

It can be shown that the probability of applying pattern B after pattern A in the expected pattern sequence of two patterns A → B is higher than the probability of applying any different pattern as second in this expected pattern sequence, with which the first pattern in this expected pattern sequence does not have the strongest symmetric relationship. This is yet another assurance that the symmetries of relationships between patterns respect the most probable uses of patterns that are expected to be applied together.

7 Evaluation

The three resulting expected pattern sequences were evaluated for their meaningfulness by letting 16 participants of an experiment describe their use in pattern stories and by suggesting the probability of their use. The evaluation was conducted to determine whether relationships between patterns in established symmetry groups form the most expected pattern sequences and to verify that these most expected pattern sequences are effectively composed. According to Coplien and Zhao (2000), patterns are effectively used in system composition if patterns are applied as structure-preserving transformations.

7.1 Input data to experiments

Text descriptions of patterns in the Piecemeal Growth pattern language and Organizational style pattern language from Coplien and Harrison (2004) used in experiments with both methods presented in this article were intentionally chosen to identify expected solutions to problems of growing organizations and their processes. These expected solutions are represented by the most expected and expected pattern sequences.

Figure 7 shows the kick-off pattern sequence established in Sect. 3.1 being inserted into the tool we developed to construct a stochastic tree. 262,141 nodes were required to implement a stochastic tree for this kick-off pattern sequence, if each node must hold information about all its parents, along with the probability of the use of each of these nodes representing the pattern sequence. Users of this tool have the option to insert new nodes into the stochastic tree by inserting the pattern name abbreviation into the relevant text field labeled “Pattern Abbrev.” A value has to be inserted into the text field labeled “Number of patterns” as well, and it’s the number of patterns the user works with. Each node is added to the stochastic tree after clicking on the submit button “Insert node.” After that, the user does not have to specify the number of patterns anymore, but the text field for this number becomes non-editable.

Because this kick-off sequence consists of 17 patterns, the stochastic tree for this sequence would be too big to be displayed with the JGraphT graphic library.[10] JGraphT library only allows drawing a binary tree, and functionality behind the stochastic tree had to be implemented by us on the client side. JGraphT library is also slow when it comes to loading a graph of more than 17 levels deep.

Stochastic trees are used only to find an expected pattern sequence candidate. If the visualization of the stochastic tree is too big to be visualized, the tool we used stops visualizing it, provides a warning, and displays the expected pattern sequence candidate. The structure of the stochastic tree is constructed, but its visualization is stopped.

As can be seen in the Fig. 8, the tool displays the expected pattern sequence candidate along with the probability of

[10] https://jgrapht.org/guide/UserOverview.

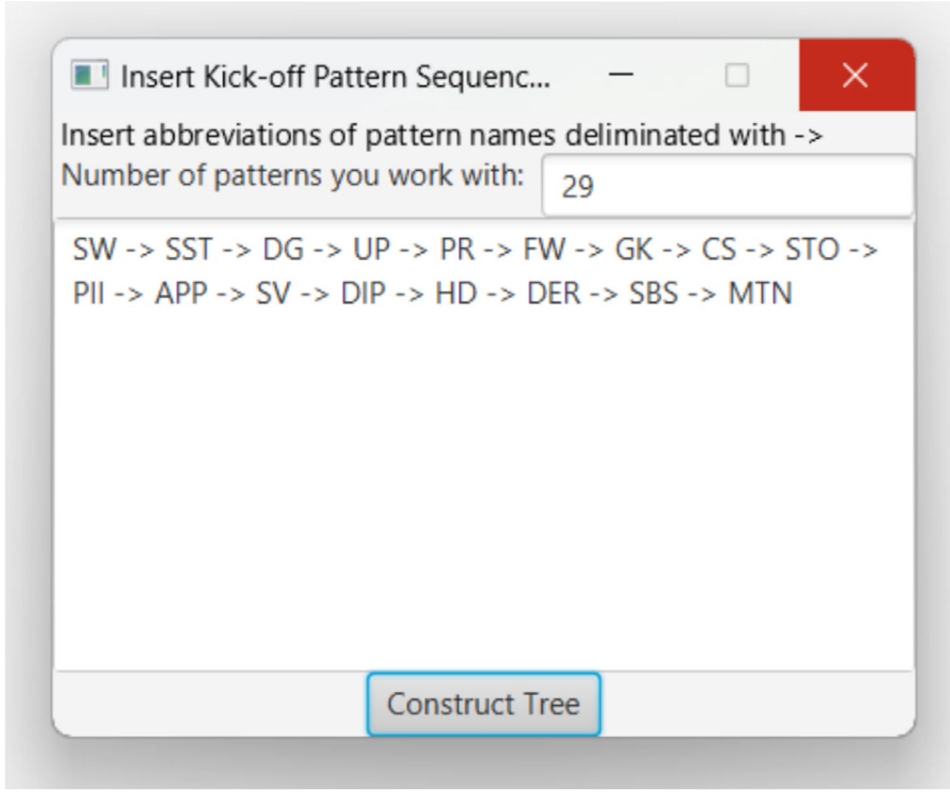

Fig. 7 A kick-off pattern sequence inserted into our tool to construct the stochastic tree

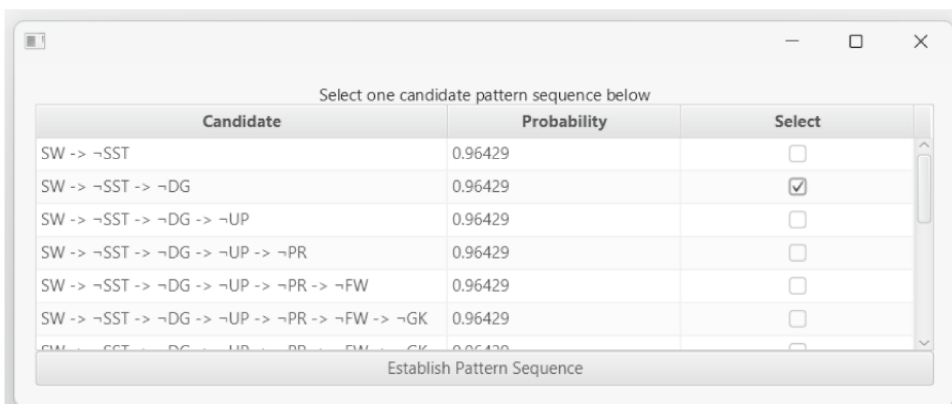

Fig. 8 All candidates for the expected pattern sequences identified in the stochastic tree

applying it by the assumed domain expert. This probability was calculated in a stochastic tree.

After clicking the button "Establish Pattern Sequence," users are redirected to a new tab where they can add additional applicable patterns after the first pattern in the candidate sequence into the pattern map of applicable patterns. Abbreviations of names of other applicable patterns can be inserted into the text field labeled "Next applicable pattern abbreviation." If any applicable pattern links to the first pattern that is used in the candidate for the expected pattern sequence, the checkbox "Two-way relationship" must be checked. If the central pattern in the pattern map does not link to the pattern, but this pattern links to the central pattern in the pattern map, the checkbox "Link to central pattern" must be checked. Each relationship between the central pattern in the pattern map and the additional applicable pattern is added to this map after clicking on the button "Insert node."

After each addition of the relationship into the pattern map, a new row is added to the table with the conditional probabilities and the symmetry of relationships. This table is displayed on the right side of the tab and it consists of four columns:

- The column "Relationship" indicated that the pattern in the expected pattern sequence candidate links to another applicable pattern
- The column "Probability" displays the conditional probability of applying an additional applicable pattern after the central pattern in the pattern map. This probability was extracted from the stochastic tree
- The column "Opposite Relationship" that provided the inverse probability of applying an additional applicable pattern before application of the central pattern in the pattern map, which is calculated using Bayes' rule
- The column "Symmetry of Relationship" provides the symmetry of the relationship between the central pattern in the pattern map and each applicable pattern, which is calculated as the absolute value of the difference between values displayed in columns "Probability" and "Opposite probability"

The pattern expected to be applied the most after the central pattern in the pattern map was the one that had the strongest symmetry (the lowest number in the column "Symmetry of Relationship" in Fig. 9) of the relationship with the central pattern in this map. The name of this next expected pattern was provided to the user after clicking on the button "Find next applicable pattern."

Using the strengths of symmetries of relationships between patterns in Fig. 9, one of the most expected to be used, four expected to be used, and one unexpected to be used pattern sequence can be established. The expected pattern sequences are: Skunk Works → Patron Role, Skunk Works → Fire Walls, Skunk Works →, Skunk Works → Gate Keeper, and Skunk Works → Compensate Success. Unexpected pattern sequence to be used due to the highest (weakest) symmetry of relationship is Skunk Works → Size The Organization.

7.2 Pattern stories

A sequence of organizational patterns, Skunk Works → Self-Selecting Team → Diverse Groups, which was found to be

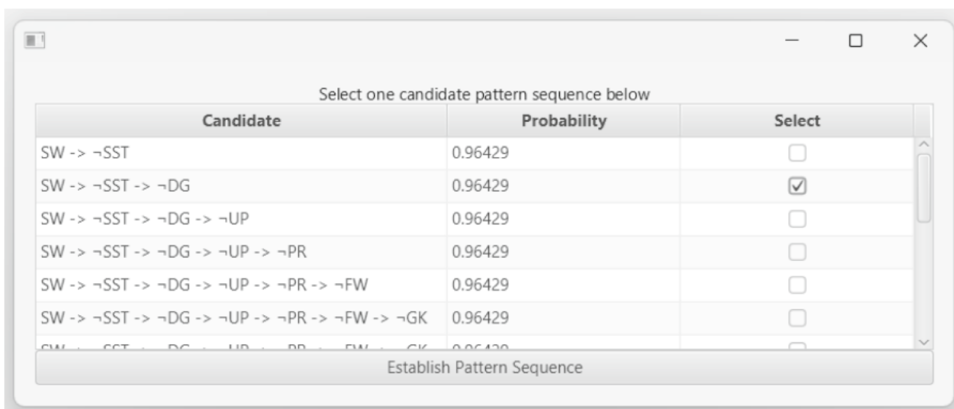

Fig. 9 Self-Selecting Team is the most expected to be used after the Skunk Works organizational pattern

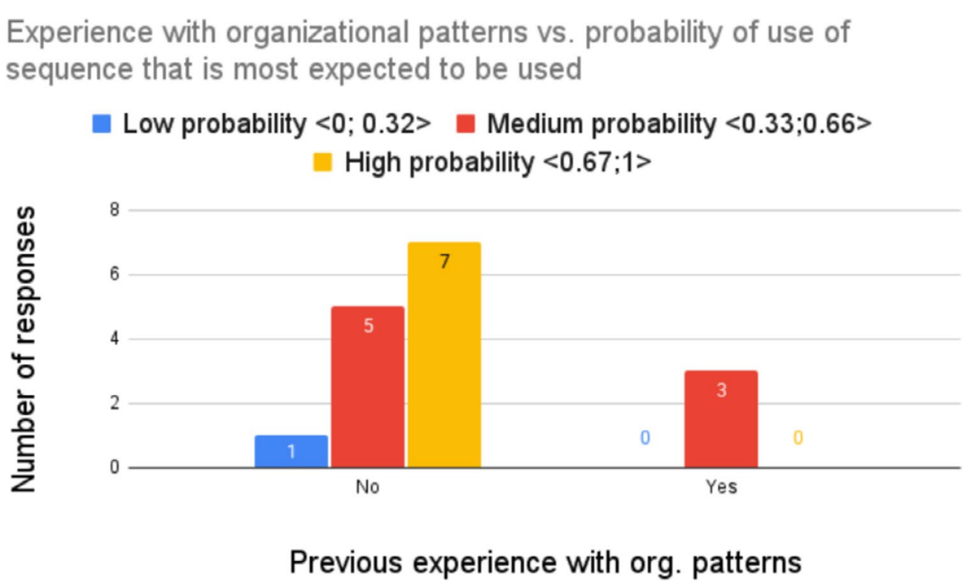

Fig. 10 The probability of the use of the most expected pattern sequence

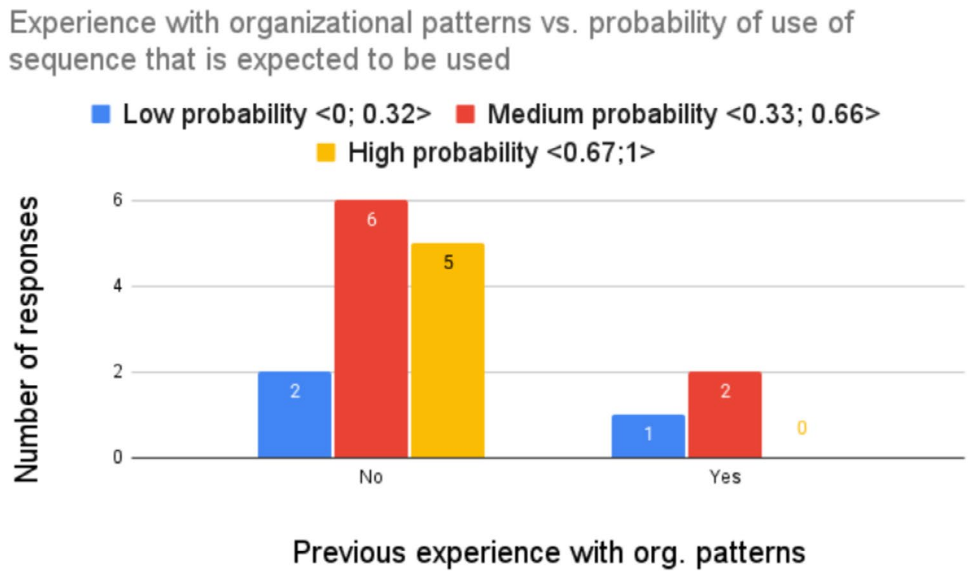

Fig. 11 The probability of the use of the expected pattern sequence

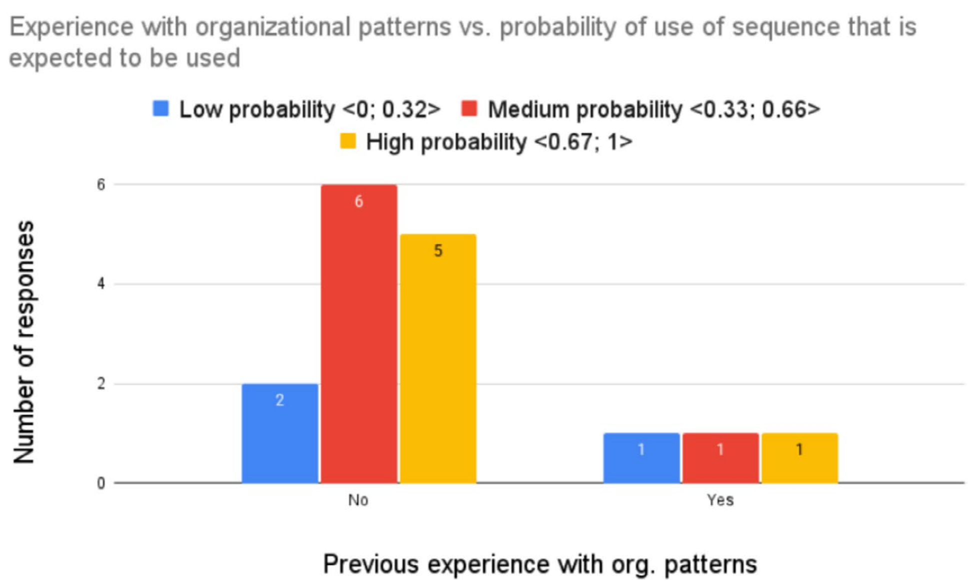

Fig. 12 The probability of the use of the expected pattern sequence

most expected to be used, was described by 16 participants during an experiment in a pattern story. All participants were also asked to suggest a probability of use of this pattern sequence. Their responses are visualized in Fig. 10. Participants with a previous experience with organizational patterns found the pattern sequence to be applicable with a medium probability. Seven participants without experience find the pattern sequence most likely to be used in practice. Although they had no experience with organizational patterns, they were able to describe how this pattern sequence can be used. All patterns stories written by participants during the experiment can be seen in the repository.[11]

A sequence of organizational patterns, Skunk Works → Self-Selecting Team → Patron Role, which was found to be expected to be used, was described by 16 participants during an experiment in a pattern story. All participants were also asked to suggest a probability of use of this pattern sequence. Their responses are visualized in Fig. 11. Participants with a previous experience with organizational patterns found the pattern sequence to be applicable with a medium to high probability. Five participants find the pattern sequence most likely to be used in practice. Although they had no experience with organizational patterns, they were able to describe how this pattern sequence can be used. All

[11] https://github.com/viktorFIIT/fiit-research-resources/tree/main/stochastic-tree/pattern-stories/questionnaire.

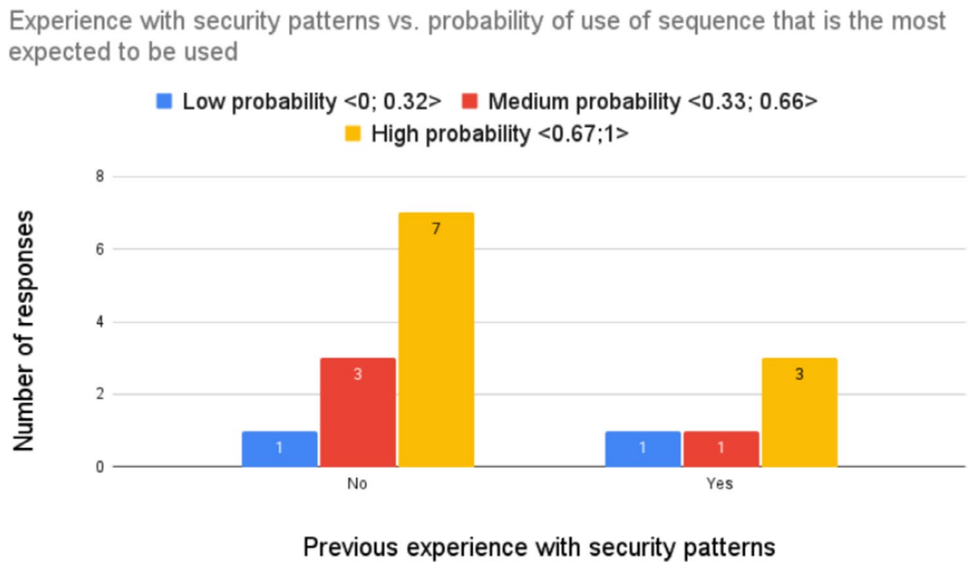

Fig. 13 The probability of the use of the most expected pattern sequence

pattern stories written by participants during the experiment can be seen in the repository.

A sequence of organizational patterns, Skunk Works → Self-Selecting Team → Apprenticeship, which was found to be expected to be used, was described by 16 participants during an experiment in a pattern story. All participants were also asked to suggest a probability of use of this pattern sequence. Their responses are visualized in Fig. 12. Participants with no previous experience with organizational patterns found the pattern sequence to be applicable with a medium to high probability. Five participants without experience find the pattern sequence most likely to be used in practice. All pattern stories written by participants during the experiment can be seen in the repository.

A sequence of organizational patterns, Access Control Requirements → Single Access Point → Security Session → Role-Based Access Control → Authorization, which was found to be most expected to be used, was described by 16 participants during an experiment in a pattern story. All participants were also asked to suggest a probability of use of this pattern sequence. Their responses are visualized in Fig. 13. Seven participants with no previous experience with security patterns found the pattern sequence to be applicable with a high probability. Three participants with experience find the pattern sequence most likely to be used in practice.

All four sequences of organizational patterns were observed in one of the big fintech and insurance companies. The pattern stories for the other two established pattern sequences are available in repository.[12]

[12] https://github.com/viktorFIIT/fiit-research-resources/tree/main/bayesian-networks/pattern-stories.

7.3 Statistical tests

A Chi-Square test discussed by McHugh (2013) was performed to check if there is a statistically significant association between participants' experience with Skunk Works → Self-Selecting Team → Diverse Groups and their probability judgments. It was found that this association is not statistically significant. An extension of Fisher's Exact Test (Freeman–Halton test) discussed by Van Auken et al. (2021) for small response counts was performed to check for statistically significant association. It was also found that this association is not statistically significant.

A Chi-Square test discussed by McHugh (2013) was performed to check if there is a statistically significant association between participants' experience with Skunk Works → Self-Selecting Team → Patron Role and their probability judgments. It was found that this association is not statistically significant. An extension of Fisher's Exact Test (Freeman–Halton test) discussed by Van Auken et al. (2021) for small response counts was performed to check for statistically significant association. It was also found that this association is not statistically significant.

A Chi-Square test discussed by McHugh (2013) was performed to check if there is a statistically significant association between participants' experience with Skunk Works → Self-Selecting Team → Apprenticeship and their probability judgments. It was found that this association is not statistically significant. An extension of Fisher's Exact Test (Freeman–Halton test) discussed by Van Auken et al. (2021) for small response counts was performed to check for statistically significant association. It was also found that this association is not statistically significant.

A Chi-Square test discussed by McHugh (2013) was performed to check if there is a statistically significant association between participants' experience with Access Control Requirements → Single Access Point → Security Session → Role-Based Access Control → Authorization and their probability judgments. It was found that this association is not statistically significant. An extension of Fisher's Exact Test (Freeman–Halton test) discussed by Van Auken et al. (2021) for small response counts was performed to check for statistically significant association. It was also found that this association is not statistically significant.

7.4 Symmetry of relationships between patterns

The resulting most expected pattern sequences were evaluated to see if symmetric relationships between the patterns they consist of form a symmetry group, the same way as Coplien and Zhao (2000) evaluated the existence of symmetric systems. If they form a symmetry group, the most expected pattern sequences are declared as symmetric. It was possible to verify that all symmetric relationships used to establish the most expected and expected pattern sequences in this article form symmetric groups. During the evaluation, it was also found that all three of the most expected pattern sequences established in this article are symmetric.

The strongest symmetric relationships between patterns forming a symmetric group can be inverted. According to Coplien and Zhao (2000), this invertible relationship is an invertible symmetry transformation if it is part of the symmetry group. If it is possible to identify a set of all invertible symmetry transformations of a state space, then the identity of the symmetry group for the equivalence relation is identified (Coplien and Zhao 2000; Rosen 2019).

The group is symmetric if it consists of symmetric operations. It is possible to evaluate if the most expected pattern sequence Skunk Works → Self-Selecting Team → Diverse Groups is based on symmetric relationships forming the symmetry group for the equivalence relation according to four axioms of group theory (Coplien and Zhao 2000; Rosen 2019):

1. *Closure*: Skunk Works → Self-Selecting Team, Self-Selecting Team → Diverse Groups subsequences of the most expected pattern sequence are part of the group G. The relationships between Skunk Works and Self-Selecting Team, and relationships between Self-Selecting Team and Diverse Groups in this most expected pattern sequence are closed under composition because they can be applied one after another.
2. *Associativity*: holds between patterns in sequences in general.
3. *An identity transformation*: means applying Skunk Works, Self-Selecting Team, and Diverse Groups again.
4. *Existence of inverses*: The organization can be returned to its previous state by dissolving the Skunk Works team.

The symmetric relationships in Skunk Works → Self-Selecting Team and in Self-Selecting Team → Diverse Groups form a group. Because applying this pattern sequence preserves the structure of the organization, both relationships form a symmetric group. The symmetric relationships between the subsequently used patterns in the other two most expected pattern sequences, established with the method of establishing expected pattern sequences based on Bayesian networks (presented in Sect. 4), form symmetric groups.

7.5 Threats to validity

Using text descriptions of organizational patterns from Coplien and Harrison (2004) containing links to other applicable patterns is a threat to the external validity, because these text descriptions are characterized by containing explicit links to other applicable patterns. This threat to the external validity can result in not establishing all expected pattern sequences of organizational patterns, or in establishing expected pattern sequences using only text descriptions mentioning other applicable patterns. This threat was mitigated by applying both methods to security patterns from the catalog of security patterns by Cordeiro et al. (2022). The application of both methods on security patterns showed that both methods apply not only to patterns in pattern languages but also to patterns in pattern catalogs.

The user of both methods decides whether kick-off pattern sequences established in both methods are meaningful, which is another threat to the external validity because falsely declaring meaningful kick-off pattern sequences as not meaningful can result in not establishing all expected pattern sequences. This threat to external validity can result in the establishment of expected pattern sequences that are not useful in practice. This threat to external validity can be mitigated by extracting pattern sequences from pattern stories associated with pattern languages and using these sequences instead of those provided by users of our methods.

The catalog of security patterns by Cordeiro et al. (2022) contains empty sections for some security patterns because the authors of these patterns did not provide sufficient details. A threat to external validity is that some expected pattern sequences may not be established if not all applicable patterns can be extracted from text descriptions. This can result in not establishing all expected pattern sequences or establishing only expected pattern sequences using text descriptions of patterns that mention other applicable patterns.

The requirement of both methods to prepare text descriptions of patterns as input to these methods is a threat to internal validity. Patterns provided as input to both methods must be understood by the users of these methods to be able to determine the meaningfulness of kick-off, the most expected, expected, and unexpected pattern sequences, which is not possible if text descriptions of patterns are not available to the users of these methods. This threat to internal validity can result in establishing expected pattern sequences using only patterns that have their text descriptions. This threat to internal validity can be mitigated by searching for existing pattern stories and using them to validate the meaningfulness of expected pattern sequences.

The threat to the internal validity is that the symmetry of the relationship cannot be calculated for those patterns that are visualized in pattern maps of applicable patterns and that are not used in kick-off pattern sequences established in step 1 of the method of establishing expected pattern sequences based on stochastic trees and in step 1 of the method of establishing expected pattern sequences based on Bayesian networks because the strength of symmetry of relationships in these methods is calculated using conditional probabilities extracted from stochastic trees modeling these kick-off pattern sequences. This threat to internal validity can result in ignoring some relationships between patterns and not establishing some expected pattern sequences.

If the last used pattern in the expected pattern sequence has the strongest symmetric relationship with the previously used pattern in this sequence, and if this previously used pattern has the strongest symmetric relationship with the last used pattern in this sequence, a circular dependency between the last two used patterns is created. The existence of circular dependency between patterns is a threat to internal validity because there is no other pattern expected to be applied next, and the pattern sequence cannot continue. This threat to internal validity can result in establishing constrained expected pattern sequences, solving selected complex problems only partially. An example of this threat to internal validity can be shown in the application of the method of establishing pattern sequences based on Bayesian networks to security patterns from the catalog of security patterns by Cordeiro et al. (2022). It can be shown that the pattern sequence Access Control Requirements → Single Access Point → Security Session → Role-Based Access Control → Authorization → Role-Based Access Control is the most expected pattern sequence, but there is a circular conditional dependency between Role-Based Access Control and Authorization. It can be shown that the probability that another security pattern from the catalog of security patterns by Cordeiro et al. (2022) would be used between Security Session and Authorization and would be able to break the circular dependency in this the most expected pattern sequence can be calculated with Bayesian network and is equal to zero, thus no other pattern can break this circular dependency between Role-Based Access Control and Authorization.

The probability of using the pattern from the pattern language does not have to conform to a uniform probability distribution. The probability of using the pattern can differ from pattern to pattern and domain to domain, and can be produced by their accompanying probability density function. According to Kruschke (2015), the probability density function can be approximated using the Markov chain Monte Carlo method for each pattern language to calculate the probability of using a pattern documented in this language without the need to calculate the denominator in Bayes' rule. Multiple ways of calculating the probability of applying patterns and expected pattern sequences threaten the conclusion validity because the probability of using patterns can be dramatically higher and can impact the use of the resulting expected pattern sequence by the user of these methods. This threat to conclusion validity can result in ignoring expected pattern sequences by pattern users in practice while solving selected complex problems.

If all documented organizational patterns in pattern language are to be used with the same probability, the probability of using expected pattern sequences established from these patterns tends to be very small which impacts their use by the pattern users. Because of this, the true expected pattern sequence can be falsely declared as unexpected. This is yet another threat to conclusion validity because these expected pattern sequences can be ignored by pattern users while solving selected complex problems.

In the method of establishing expected pattern sequences based on stochastic trees, some expected pattern sequences can also be established using documented relationships between patterns. One example of this situation is the existence of the pattern sequence Skunk Works → Compensate Success → Skunk Works, which can also be established without using this method because both Skunk Works and Compensate Success mention each other in their text descriptions. This is a threat to internal validity because some expected pattern sequences can be established without using this method.

In the method of establishing expected pattern sequences based on Bayesian networks, some expected pattern sequences can also be established using documented relationships between patterns. One example of this situation is the existence of the expected pattern sequence Skunk Works → Compensate Success → Skunk Works, which can also be established without using this method because the symmetry of the relationship between Compensate Success and Skunk Works will always be stronger than the symmetry of the relationship between Skunk Works and Skunk Works. This is a threat to internal validity because some expected pattern sequences can be established without using this method.

Using the stochastic tree in the method of establishing expected pattern sequences presented in Sect. 3 poses a threat to the internal validity because it prefers longer candidate sequences of patterns, as nodes with higher probabilities are present in the rightward spectrum of the tree. This threat to internal validity can result in not establishing shorter expected pattern sequences.

Threat to the internal validity is, according to Kruschke (2015), that Bayesian networks suffer from an inability to express all conditional dependency and independence relationships between patterns. According to Studený (2001), all graphical models suffer from the same disadvantage. This is a threat to the internal validity.

A threat to construct validity is that expected pattern sequences established using the methods of establishing expected pattern sequences based on stochastic trees presented in Sect. 3 and the method of establishing expected pattern sequences based on Bayesian networks presented in Sect. 4 may not have to be found useful in practice.

A threat to the external validity is that the method of establishing expected pattern sequences based on stochastic trees does not consider patterns referenced by patterns provided as input to this method for which textual descriptions are unavailable.

8 Conclusions and further work

This article presents two methods of establishing pattern sequences using stochastic processes. One is based on stochastic trees (presented in Sect. 3). The other method (presented in Sect. 4), based on Bayesian networks, extracts the strongest symmetric relationships between organizational patterns to use them in establishing meaningful sequences of patterns. Three expected sequences of patterns from the Piecemeal Growth and Organizational Style pattern language were established using these two methods. The pattern sequences we established are expected to be used because they were established using the highest probabilities of the subsequent use of patterns, and the probability of use of these sequences is higher than the probability of any different combinations of patterns. The pattern stories based on these pattern sequences demonstrate their meaningfulness.

The applicability of these two methods is not limited to organizational patterns. We applied them to the catalog of security patterns by Cordeiro et al. (2022).[13] This catalog is compiled from different sources, and there is a need to connect these patterns better. One of the applications of an improved catalog could be within the area of hybrid security threats. These two methods can also be used to establish expected pattern sequences of patterns for engineering software for the cloud, documented by Sousa et al. (2022).

For their effective use, the methods proposed in this article should be automated. For example, the computations behind Bayesian networks can be automated by implementing the junction tree algorithm presented by Barber (2011).

[13] https://github.com/xextreme22/Security-Patterns.

Author contributions Viktor Matovič is the main contributor and corresponding author. Valentino Vranić is the second contributor. Waheedullah Sulaiman Khail and Neil Harrison reviewed the text and contributed to the clarity of the approach.

Funding Open access funding provided by The Ministry of Education, Science, Research and Sport of the Slovak Republic in cooperation with Centre for Scientific and Technical Information of the Slovak Republic. This work is funded by the EU NextGenerationEU through the Recovery and Resilience Plan for Slovakia under the project Artificial Intelligence for Legal Professions (AILE) No. 09I05-03-V02-00038 and by the Cultural and Educational Grant Agency of Slovak Republic (KEGA) under grant No. KG 014STU-4/2024, also by the Slovak Research and Development Agency under Contract No. APVV-23-0408.

Data availability Not applicable.

Declarations

Conflict of interest On behalf of all authors, the corresponding author states that there is no conflict of interest.

Human and animal rights Not applicable.

Informed consent Not applicable.

References

Alexander C (2002) The process of creating life: nature of order, book 2: an essay on the art of building and the nature of the universe. Center for Enviromental Structure

Alexander C, Ishikawa S, Silverstein M (1977) A pattern language: towns, buildings, construction. Oxford University Press, Oxford

Barber D (2011) Bayesian reasoning and machine learning, 04 edn. Cambridge University Press, Cambridge (**in press**)

Buschmann F, Henney K, Schmidt DC (2007) Pattern oriented software architecture: on patterns and pattern languages. Wiley, Hoboken

Coplien J, Zhao L (2000) Symmetry breaking in software patterns. In: Proceedings of the second international symposium on generative and component-based software engineering, GCSE 2000, Revised Papers. Springer, Erfurt, Germany

Coplien JO, Harrison NB (2004) Organizational patterns of agile software development. Prentice-Hall, Upper Saddle River

Coplien JO, Schmidt DC (eds) (1995) Pattern languages of program design. ACM Press, New York

Cordeiro A, Vasconcelos A, Correia M (2022) A catalog of security patterns. In: Proceedings of 29th Conference on Pattern Languages of Programs Online, PLoP 2022. ACM

Falkenthal M, Breitenbücher U, Leymann F (2018) The nature of pattern languages

Gamma E, Helm R, Johnson R, and Vlissides J (1994) Design patterns: elements of reusable object-oriented software. Addison-Wesley Longman Publishing Co., Inc., USA.

Greene SI (2003) Tool-based decision support for pattern assisted development. Workshop at CHI 2003, Fort Lauderdale, Florida, USA, Perspectives on HCI Patterns: Concepts and Tools

Hazen G (1992) Stochastic trees: a new technique for temporal medical decision modeling. Med Decis Mak 12:163–78

Jackson M (1995) Software requirements & specifications: a lexicon of practice, principles and prejudices. ACM Press, New York (**cited in [4]**)

Jackson M (2001) Problem frames: analyzing and structuring software development problems. Addison Wesley, Reading

Janeiro, J., Barbosa, S.D.J., Springer, T., Schill, A. (2010). Improving the Search for User Interface Design Patterns through Typed Relationships. In: Forbrig, P., Paternó, F., Mark Pejtersen, A. (eds) Human-Computer Interaction. HCIS 2010. IFIP Advances in Information and Communication Technology, vol 332. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-15231-3_2

Kaliyar RK (2020) A multi-layer bidirectional transformer encoder for pre-trained word embedding: a survey of bert. In: Proceedings of 2020 10th International Conference on Cloud Computing, Data Science ‘I &’ Engineering (Confluence). IEEE, Noida, India

Kodituwakku SR, Bertok P (2003) Pattern categories: a mathematical approach for organizing design patterns. In: Proceedings of 2002 Conference on Pattern Languages of Programs, CRPIT ’02. Australian Computer Society, Inc., Victoria

Koller D, Friedman N (2009) Probabilistic graphical models: principles and techniques—adaptive computation and machine learning. The MIT Press, Cambridge

Kruschke J (2015) Doing Bayesian data analysis, 2nd edn. Academic Press, Cambridge

Kubo A, Washizaki H, Takasu A, Fukazawa Y (2005) Extracting relations among embedded software design patterns. J Integr Design Process Sci 9:39–52. https://doi.org/10.3233/JID-2005-9304

Matovič V, Vranić V (2024) Establishing pattern sequences using stochastic processes with an application to security patterns. In: Proceedings of the 29th European conference on pattern languages of programs, EuroPLoP 2024. ACM, Kloster Irsee in Bavaria, Germany. https://doi.org/10.1145/3698322.3698357

Matovič V, Vranić V (2025) Establishing pattern sequences using stochastic trees and Bayesian belief networks with an application to security patterns. IEEE Access 13:32443–32457. https://doi.org/10.1109/ACCESS.2025.3543319

Van Auken, R. M., & Kebschull, S. A. (2021). Type I error convergence of three hypothesis tests for small RxC contingency tables. RMS: Research in Mathematics & Statistics, 8(1). https://doi.org/10.1080/27658449.2021.1934959

McHugh ML (2013) The chi-square test of independence. Biochemia Medica 23(2):143–149. https://doi.org/10.11613/BM.2013.018

Noble J (1998) Classifying relationships between object-oriented design patterns. In: Proceedings of 1998 Australian Software Engineering Conference. IEEE, Adelaide, SA, Australia

Rosen J (2019) Symmetry in science: an introduction to the general theory. Springer, Berlin

Seidel N (2017) Empirical evaluation methods for pattern languages: sketches, classification, and network analysis. In: Proceedings of the 22nd European Conference on Pattern Languages of Programs, EuroPLoP ’17. ACM, Kloster Irsee in Bavaria, Germany. https://doi.org/10.1145/3147704.3147719

Sousa TB, Ferreira HS, Correia FF (2022) A survey on the adoption of patterns for engineering software for the cloud. IEEE Trans Softw Eng 48(6):2128–2140

Studený M (2001) On non-graphical description of models of conditional independence structure. Academic Press, Cambridge

Sulaiman Khail W, Vranić V (2019) Reflecting pattern relationships in a pattern format. In: Proceedings of 24th European Conference on Pattern Languages of Programs, EuroPLoP 2019. ACM, Irsee, Germany

Penn State (2023). STAT 415: Introduction to mathematical statistics. Maximum likelihood estimation. https://online.stat.psu.edu/stat415/lesson/1/1.2

Vaishnavi V, Kuechler B (2004–2017) Design science research in information systems. http://www.desrist.org/design-research-in-information-systems/

Vaishnavi VK, Vaishnavi VK, Kuechler W (2015) Design science research in information systems. CRC Press, Boca Raton

Vranić V, Vranić A, Sulaiman Khail W (2020) Growing organizations with patterns: lessons from drama. In: Proceedings of the 25th European Conference on Pattern Languages of Programs, EuroPLoP 2020 Online. ACM, Virtual Event

Waseeb S, Sulaiman Khail W, Gul Wahaj H, and Vranić V (2020) Extracting relations between organizational patterns using association mining. In: Proceedings of European Conference on Pattern Languages of Programs 2020, EuroPLoP 2020. ACM, Virtual Event, Germany

Hironori Washizaki, Masashi Kadoya, Yoshiaki Fukazawa, and Takeshi Kawamura. 2014. Network Analysis for Software Patterns Including Organizational Patterns in Portland Pattern Repository. In: Proceedings of the 2014 Agile Conference (AGILE '14). IEEE Computer Society, USA, 66–67. https://doi.org/10.1109/AGILE.2014.12.

Zhang J, Yue H, Wu X and Chen W, A brief review of Bayesian belief network. 2019 Chinese Control And Decision Conference (CCDC), Nanchang, China, 2019, pp. 3910–3914. https://doi.org/10.1109/CCDC.2019.8832649.