Establishing Pattern Sequences Using Stochastic Processes with an Application to Security Patterns

Viktor Matovič**
xmatovicv@stuba.sk
Slovak University of Technology in Bratislava, Faculty of Informatics and Information Technologies, Institute of Informatics, Information Systems, and Software Engineering
Bratislava, Slovak Republic

Valentino Vranić*
vranic@stuba.sk
Slovak University of Technology in Bratislava, Faculty of Informatics and Information Technologies, Institute of Informatics, Information Systems, and Software Engineering
Bratislava, Slovak Republic

ABSTRACT

Security patterns can help organizations develop secure software. While it is typical to encounter isolated applications of one or more software patterns in practice, effectively responding to security threats demands a comprehensive grasp and utilization of an entire pattern language or catalog, comprising numerous intricately interconnected patterns. The composition of software patterns makes them more efficient. Solutions for establishing sequences of software patterns are often found documented in plain text through explicit pattern relationships. This article presents two experiments of establishing pattern sequences using stochastic processes, one based on stochastic trees and the other, based on Bayesian belief network to establish meaningful pattern sequences. We used stochastic trees and Bayesian belief networks to find conditionally dependent patterns between which we tried to find the most expected combinations in pattern sequences. By employing these two processes, we established four envisioned sequences of patterns from the security pattern catalog. The pattern stories based around these pattern sequences demonstrate their meaningfulness. These two processes are not exclusively relevant to security patterns. We consider applying them to the patterns for engineering software for the cloud.

CCS CONCEPTS

• **Software and its engineering**; • **Security and privacy** → **Software security engineering**;

KEYWORDS

Software and its engineering, Security and privacy, Probability and statistics

ACM Reference Format:
Viktor Matovič and Valentino Vranić. 2018. Establishing Pattern Sequences Using Stochastic Processes with an Application to Security Patterns. In *Proceedings of 29th European Conference on Pattern Languages of Programs (EuroPLop'29).* ACM, New York, NY, USA, 10 pages. https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

Security patterns are best practices that solve recurring security problems in specific contexts [6]. Security patterns can be used to implement secure and reusable software systems [6]. Conditional dependence between patterns is not considered to establish the most expected pattern sequences to challenge complex security problems. The stochastic processes can be used to establish the most expected pattern sequences and tried on the set of security patterns to identify expected solutions to security problems. Cordeiro 's [3] catalog of security patterns consists of 106 security patterns that help companies implement security guidelines. Security patterns collected by Cordeiro can be established in pattern sequences to help challenge complex problems. Guan et al. [6] note it is hard to implement security in legacy systems because of a lack of methods to select appropriate security patterns.

Implementing security in software requires the use of several security patterns. Conditionally dependent patterns identified from Bayesian belief networks [1] can be checked if they have the strongest symmetries of relationships to establish the most expected pattern sequences. Stochastic trees can be used to calculate probabilities of subsequent use of patterns because these probabilities are needed to calculate the strength of symmetries of relationships between these patterns.

This article presents two experiments of establishing meaningful and the most expected pattern sequences using Hazen's [7] stochastic trees and Bayesian belief networks discussed in the work of Barber [1].

The rest of this article is structured as follows. Section 2 explains how explicit and implicit relationships between patterns can establish pattern sequences. Section 3 presents two experiments of establishing expected pattern sequences using symmetries of relationships between patterns with the help of the stochastic trees and the Bayesian belief network. Section 4 discusses the pitfalls of establishing expected pattern sequences using stochastic trees and Bayesian networks. Section 5 explains how pattern sequences established here were evaluated in this article. Section 6 concludes the article with the findings and opportunities for further work.

[*] main contributor to the article

EuroPLop'29, July 03–07, 2024, Kloster Irsee, Germany

ACM ISBN 978-1-4503-XXXX-X/18/06
https://doi.org/XXXXXXX.XXXXXXX

2 EXISTING METHODS OF ESTABLISHING PATTERN SEQUENCES

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

Kubo et al. [8] used a model to calculate the strength of pattern relations to identify explicitly related design patterns from Gamma et al. [5] applicable together. Kubo et al. [8] did not specify the order in which patterns applicable together should be applied. The order in which patterns are expected to be applied the most is one type of implicit relationship between patterns.

Shameli-Sendi et al. [10] proposed Security Defense Patterns Aware Placement as a framework that allows efficient placement of network security defense patterns in a virtualized environment, such that this placement captures security constraints and optimizes resources. Shameli-Sendi et al. do not establish pattern sequences to optimize resources or achieve specific and predefined security goals. Shameli-Sendi et al. predefine the order of relationships between network security defense patterns before applying algorithms to propose their optimal placement, but this order might not be known. The placement algorithm proposed by Shameli-Sendi et al. is unsuitable for large cloud environments. It's estimated to work with networks of up to 69 nodes which is a hard constraint. The method to establish pattern sequences must apply to any software patterns, and established pattern sequences must not be constrained to security networks.

Motii et al. [9] designed an approach to select security patterns to tackle predefined security threats. We think this approach will not work if there are no security threats to react to.

Sousa et al. [11] identified 12 software patterns that are *very likely* to be used together and patterns that are only *likely* to be used together. Together with Kubo et al. [8], Sousa et al. did not specify the order in which patterns applicable together should be applied. Sousa et al. and Kubo did not consider conditional dependence between patterns to establish pattern sequences.

The formula used in the work of Sousa et al. [11] for calculating the symmetry of relationships between two software patterns specified as the difference between probabilities of their subsequent use can be used to identify security patterns from the pattern catalog of Cordeiro [3] that have the strongest symmetries of relationships.

We think that sequences of security patterns need to be established:

- Considering the symmetry of relationships that might point to patterns most expected to be applied together.
- Considering conditional dependence because these are patterns expected to be used in the pattern sequence.
- Without the need to update text descriptions of security patterns.

3 ESTABLISHING PATTERN SEQUENCES USING STOCHASTIC PROCESSES

Discrete-time stochastic processes of stochastic trees and Bayesian networks based on Markov stochastic processes $(X_t)_{t\in N}$ are used here in two experiments. Input to these experiments was a collection of security patterns. Uses of these patterns must be because of the Bayesian belief networks considered as random events in time t from index set (0, N), where N stands for a specific number of security patterns to choose from in pattern catalog of Cordeiro [3]. Random variables represent applications of security patterns in probability space$(\Omega, (\mathcal{F}_t)_{t\in\mathcal{R}}, \mathcal{P})$ of all security patterns Ω, where each application of security pattern from pattern catalog is assigned probability from probability space $\mathcal{P}$ in the stochastic tree.

Hazen's stochastic trees [7] are used here to find the most probable candidate for the most expected pattern sequence by selecting the node with the highest probability from stochastic tree modeling one chosen pattern sequence. We use Bayesian belief networks to identify conditionally dependent patterns. We try to find the most expected combinations between patterns identified as conditionally dependent.

An experiment of establishing sequences of security patterns using stochastic trees and strengths of symmetries of relationships between these patterns is reported in Section 3.1. An experiment of establishing sequences of security patterns using the Bayesian belief network and strengths of symmetries of relationships between these patterns is reported in Section 3.2.

3.1 Establishing Pattern Sequences Based on Stochastic Trees

Hazen's stochastic trees [7] can be constructed on top of any pattern sequence to find the most probable candidate for the most expected pattern sequence by selecting the node with the highest probability. Each pattern in this most expected pattern sequence candidate can be checked to see if the subsequently applied pattern has the strongest symmetric relationship with the previous pattern. The strength of the symmetry of the relationship between two patterns can be calculated similarly to Sousa et al. [11] as the absolute value of the difference of the conditional probability of applying pattern after another pattern and the opposite probability of it calculated with Bayes rule. Patterns with the strongest symmetry of relationship are expected to be used the most in the pattern sequence. Pattern sequences we established are the most expected because they are based on the strongest symmetries of relationships between patterns calculated with conditional probabilities.

Input to the first experiment were text descriptions of security patterns documented by Cordeiro [3]. The experiment resulted in establishing the most expected and unexpected pattern sequences. Unexpected pattern sequences are not meaningful and should be avoided. Unexpected pattern sequences differ from misuse pattern sequences recognized by Fernandez et al. [4].

3.1.1 Construction of a Stochastic Tree. The kick-off pattern sequence from at least two security patterns was established by following explicit links in the pattern catalog of Cordeiro [3] consisting of 106 patterns. If less than two patterns were selected, construction of the stochastic tree would not be possible. The following kick-off pattern sequence describes software development and can be described with the pattern story.[1]

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

[1] pattern story for this sequence can be seen at https://github.com/viktorFIIT/fiit-research-resources/blob/main/stories/kick-off-sequence-pattern-story.png

This kick-off pattern sequence was established using explicit links between two subsequent patterns in this kick-off pattern sequence.

Conditional probabilities of subsequently applying one pattern after another can be calculated with the help of a stochastic tree. Its calculation can be shown in the following example:

> If the pattern user decides to establish a pattern sequence using a finite number of patterns, the probability of use of this pattern sequence can be calculated as $1/M$, where letter M stands for the number of patterns the user works with decreased by the number of patterns that are already applied at the time of calculating this probability.

For example, the variable M important to calculate the probability that Single Access Point would be used after Access Control Requirements in the kick-off pattern sequence can be calculated as $106 - 1$ because one pattern was used in the start-up pattern sequence before the Single Access Point and pattern catalog of Cordeiro consists of 106 patterns.

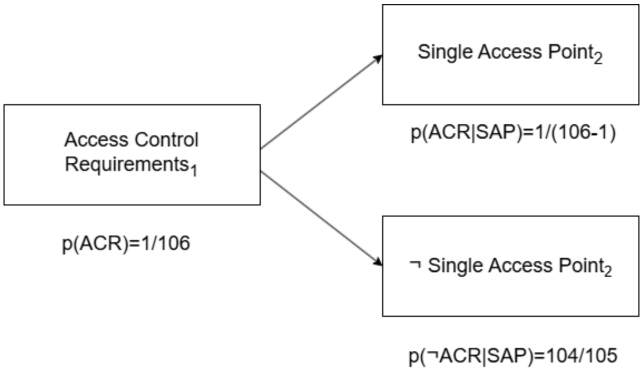

Figure 1: Stochastic tree example constructed for pattern sequence of two patterns

Although multiple nodes with the same highest probability could be selected in the stochastic tree, the stochastic tree had only one node with the highest probability.[2] The node with the highest probability in the stochastic tree constructed for the kick-off pattern sequence that represented an uninterrupted sequence of patterns is the following pattern sequence:

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

It had to be verified if the candidate pattern sequence represents the most expected pattern sequence because there was no guarantee this candidate pattern sequence was established using the strongest symmetries of relationships between patterns.

3.1.2 Symmetric Relationships Between Patterns In Pattern Maps. Another applicable pattern had to be found for each security pattern from the candidate pattern sequence from Section 3.1.1. Mapping relationships of the first pattern in the candidate for the most expected pattern sequence resulted in creating a pattern map of applicable patterns in Figure 2, where a bidirectional arrow between patterns means patterns can be applied in any order.

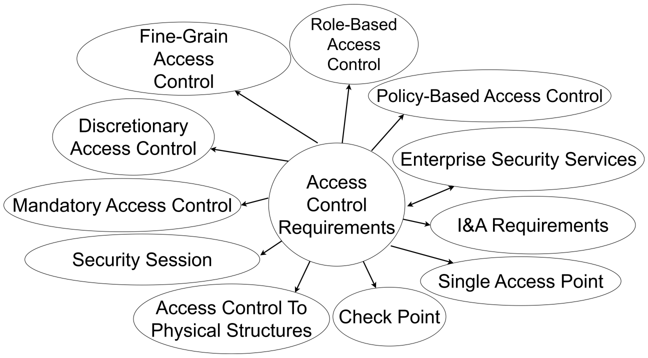
Fine-Grain Access Control
Role-Based Access Control
Policy-Based Access Control
Discretionary Access Control
Enterprise Security Services
Access Control Requirements
Mandatory Access Control
I&A Requirements
Security Session
Single Access Point
Access Control To Physical Structures
Check Point

Figure 2: The Access Control Requirements pattern map of applicable patterns

To identify the pattern expected to be used the most after Access Control Requirements, the strength of the symmetry of the relationship between Access Control Requirements and other linked patterns had to be calculated using the stochastic tree from Section 3.1.1. The strength of symmetry of the relationship between two patterns calculated in this article is similar to symmetry of relationship between patterns used by Sousa et al. [11] with the difference that here it is the *absolute* difference between probability of subsequent use of two patterns A and B and the opposite probability to it $|p(A|B) - p(B|A)|$ to identify patterns expected to be used together the most and patterns expected to be used together less more easily.

The strength of symmetry of the relationship to another pattern could only be calculated between (Access Control Requirements and Single Access Point), (Access Control Requirements and Security Session), and (Access Control Requirements and Role-Based Access Control) because only Single Access Point, Security Session, and Role-Based Access Control from Figure 2 are used in the kick-off pattern sequence. We calculated the strength of symmetry of relationships between pairs of patterns using conditional probabilities calculated during stochastic tree construction in Section 3.1.1. [3]

We calculated the strength of symmetry of the relationship by subtracting conditional probabilities of subsequent use $p(Single\ Access\ Point \mid Access\ Control\ Requirements)$, $p(Security\ Session \mid Access\ Control\ Requirements)$, $p(Role-Based\ Access\ Control \mid Access\ Control\ Requirements)$ from their inverse counterparts calculated with Bayes rule. After applying Access Control Requirements, the conditional probabilities of subsequent use of other applicable patterns in the pattern map in Figure 2 were calculated with the stochastic tree in Section 3.1.1. Because we were interested mainly in the positive strengths of symmetries of relationships between patterns, the absolute value of the difference

[2] whole stochastic tree we used can be seen at: https://github.com/viktorFIIT/fiit-research-resources/blob/main/stochastic-tree/stochastic-tree-security-patterns.pdf

[3] the application we developed simplifies the construction of the stochastic tree https://github.com/viktorFIIT/fiit-research-resources/blob/main/app/Use-On-Security-Patterns.md

between these probabilities was calculated, which resulted in the following:

- Probability of applying Single Access Point after Access Control Requirements $p(Single\ Access\ Point \mid Access\ Control\ Requirements)$ calculated with stochastic tree is 0.16666, and the opposite probability to it calculated using Bayes rule is $p(Access\ Control\ Requirements|Single\ Access\ Point) = 0.16278$. The absolute difference between these two probabilities is 0.00388 representing the strength of symmetry of the relationship between Access Control Requirements and Single Access Point.
- Probability of applying Security Session after Access Control Requirements $p(Security\ Session \mid Access\ Control\ Require-ments)$ calculated with stochastic tree is 0.20, and the opposite probability to it calculated using Bayes rule is $p(Access\ Control\ Requirements|Security\ Session) = 0.16665$. The absolute difference between these two probabilities is 0.03335, representing the strength of symmetry of the relationship between Access Control Requirements and Security Session.
- Probability of applying Role-Based Access Control after Access Control Requirements $p(Role-Based\ Access\ Control \mid Access\ Control\ Requirements)$ extracted from stochastic tree is 0.03355, and the opposite probability to it calculated using Bayes rule is $p(Access\ Control\ Requirements \mid Role - Based\ Access\ Control) = 0.17240$. The absolute difference between these two probabilities is 0.13885, representing the strength of symmetry of the relationship between Access Control Requirements and Role-Based Access Control.

The strength of symmetry of the relationship between Access Control Requirements and a Single Access Point is the strongest because of the smallest absolute value. A Single Access Point is therefore expected to be used the most after the Access Control Requirements. It is less expected that Security Session or Role-Based Access Control would be used directly after Access Control Requirements because of the weaker strengths of symmetries of relationships.

Mapping relationships of the second pattern in the candidate for the most expected pattern sequence to patterns that can be applied after it and linked in its text description resulted in a pattern map of applicable patterns in Figure 3.

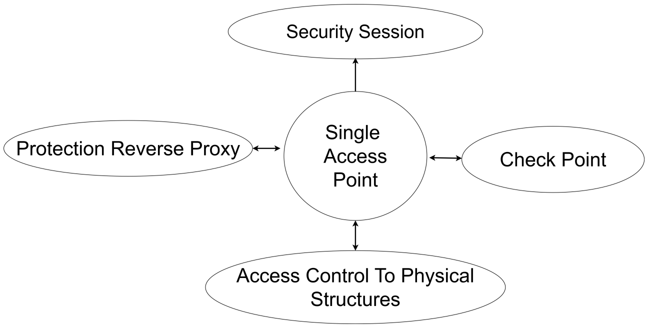

Figure 3: The Single Access Point pattern map of applicable patterns

To identify the pattern expected to be used the most after Single Access Point, the strength of the symmetry of the relationship between Single Access Point and other patterns allowed to be used after it had to be calculated with the stochastic tree from Section 3.1.1. The strength of the symmetry of the relationship with another pattern could only be calculated between Single Access Point and Security Session because only Single Access Point and Security Session from Figure 3 are used in the kick-off pattern sequence. The strength of the symmetry of the relationship between Single Access Point and Security Session is therefore strongest and equal to $|p(Security\ Session|Single\ Access\ Point) - p(Single\ Access\ Point|Security\ Session)| = 0.03334$. A Security Session is expected to be applied after a Single Access Point because a session is established after successful authentication and authorization of the user accessing the system. This system must be protected by the defined security and access policy through Access Control Requirements. Security Session is also applied directly after Single Access Point in the candidate for expected pattern sequence.

Mapping relationships of the third pattern in the candidate for expected pattern sequence to patterns that can be applied after it and linked in its text description resulted in a pattern map of applicable patterns in Figure 4.

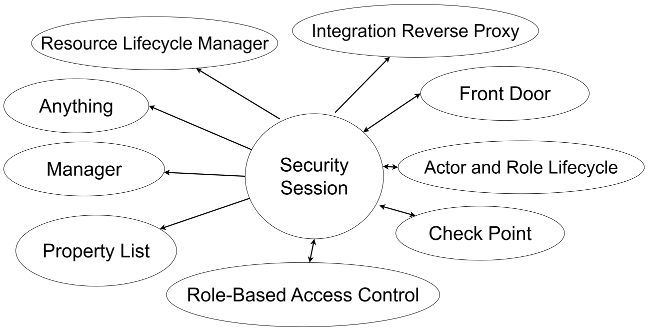

Figure 4: The Security Session pattern map of applicable patterns

To find the pattern expected to be applied after the Security Session, the strength of the symmetry of the relationship between the Security Session and another pattern in Figure 4 had to be calculated. The strength of the symmetry of the relationship with another pattern could only be calculated between Security Session and Role-Based Access Control because only Role-Based Access Control is used in the kick-off pattern sequence. If the strength of the symmetry of the relationship between patterns is calculated using conditional probabilities extracted from the stochastic tree, relationships between Security Session and Manager, Property List, Anything, and Integration Reverse Proxy patterns linked in its text description are ignored because they are not used in the kick-off pattern sequence.

The strength of the symmetry of the relationship between Security Session and Role-Based Access Control is therefore strongest and equal to $|p(Role - Based\ Access\ Control|Security\ Session) - p(Security\ Session|Role - Based\ Access\ Control) = 0.07760$. The Role-Based Access Control pattern is also applied directly after the Security Session in the candidate for the expected pattern sequence.

Mapping relationships of the fourth pattern in the candidate for expected pattern sequence to patterns that can be applied after it and linked in its text description resulted in creating a pattern map of applicable patterns in Figure 5.

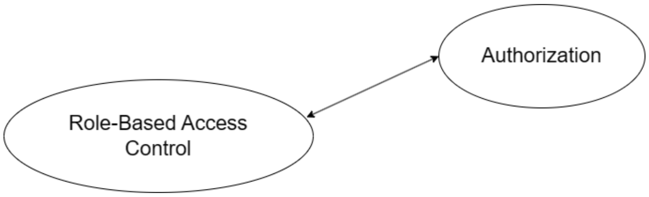

Figure 5: The Role-Based Access Control pattern map of applicable patterns

Mapping relationships of the fifth pattern in the candidate for expected pattern sequence to patterns that can be applied after it and that are linked in its text description resulted in creating a pattern map of applicable patterns in Figure 6.

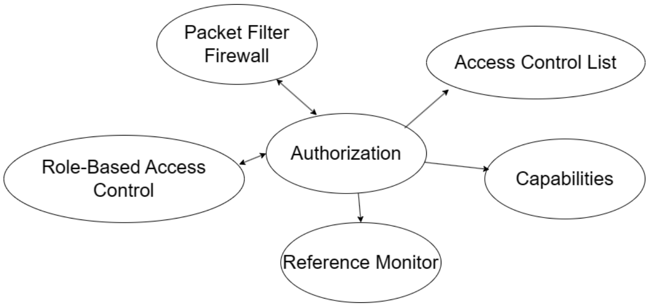

Figure 6: The Authorization pattern map of applicable patterns

The strongest symmetry of the relationship was found between Authorization and Role-Based Access Control, which points to a meaningful relationship. Because of the circular strongest symmetry of the relationship between Role-Based Access Control and Authorization, the expected pattern sequence candidate cannot continue with the application of another security pattern.

Pattern sequence Access Control Requirements → Single Access Point → Security Session → Role-Based Access Control → Authorization can be described in a pattern story.[4] Roles are sets of user permissions. They reflect the job duties of employees that are assigned to them by applying Role-Based Access Control. Roles are assigned for user accounts because of the previously defined security policy with the application of Access Control Requirements. Web systems are accessed by users with accounts with permissions, which are interpreted for successful authentication and authorization through a Single Access Point. Successful authentication results in the establishment of a Security Session.

3.1.3 *Verifying Usefulness of the Patterns in Candidate Pattern Sequence With Maximum-Likelihood Function.*

Before declaring the candidate for the expected pattern sequence as a truly expected pattern sequence, the usefulness of all patterns in this sequence had to be verified. Verification of the usefulness of a security pattern from the expected pattern sequence candidate was performed by calculating the likelihood of its usefulness with the maximum-likelihood function and comparing the output value from this function to the numerically represented confidence level inspired by confidence levels provided for organizational patterns by Coplien and Harrison [2].[5]

The usefulness of the Access Control Requirements can be verified if the output value from the Maximum likelihood function is equal to or above the numerically represented confidence level one asterisk (equalling 0.66666). Access Control Requirements from the candidate pattern sequence was the first pattern that needed to be verified. This pattern advises organizations to define effective and non-conflicting security and access policies. To verify the usefulness of the pattern, probabilities of the subsequent use of patterns it relates to in the pattern map depicted in Figure 2 had to be calculated with the stochastic tree in Section 3.1.1. The opposite values to these conditional probabilities served as input parameters θ into the maximum likelihood function.

The likelihood of the usefulness of the Access Control Requirements can be considered high enough because output value 0.64430 is near the numerically represented one asterisk of the confidence level scale in the work of Coplien and Harrison [2]. Because of the missing confidence level for each pattern in the pattern catalog of Cordeiro [3], the output value from the maximum likelihood function could not be compared to the numerically represented confidence level. Confidence level "*" can only be recommended to be stated after the name of Access Control Requirements.

$$\begin{aligned}
&p(Access\ Control\ Requirements \mid \theta_1, \theta_2, \theta_3) = \\
&p(\neg Role - Based\ Access\ Control \mid \theta_1) * \\
&p(\neg Single\ Access\ Point \mid \theta_2) * \\
&p(\neg Security\ Session \mid \theta_3) = \\
&(1 - \theta_1) * (1 - \theta_2) * (1 - \theta_3) = \\
&(1 - 0.03355) * (1 - 0.16666) * (1 - 0.20) = 0.6443051544
\end{aligned}$$

Applying the same maximum likelihood function, verification of usefulness of Access Control Requirements (likelihood 0.64430 < 0.66666), Single Access Point (likelihood 0.8 > 0.66666), Security Session (likelihood 0.75 > 0.66666), Role-Based Access Control (likelihood 0.66667 > 0.66666), and Authorization (likelihood 0.57273 < 0.66666) was successful.

Because of the number of security patterns in the pattern catalog of Cordeiro [3] (106 patterns), the probability of the subsequent use 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 small number and multiplying these probabilities together produces even smaller number. This was the reason behind accepting Access Control Requirements and Authorization in the expected pattern sequence even if the likelihood of usefulness of Access Control Requirements and Authorization is less than the numerically represented confidence level with one asterisk. Security patterns

[4] pattern story for this sequence can be seen at: https://github.com/viktorFIIT/fiit-research-resources/blob/main/stories/first-expected-pattern-sequence-story.png

[5] The maximum-likelihood function can be seen formulated at https://github.com/viktorFIIT/fiit-research-resources/tree/main/appendices

Access Control Requirements, Single Access Point, Security Session, Role-Based Access Control, and Authorization can be part of the true expected pattern sequence because the likelihood of their usefulness is sufficiently high.

3.1.4 Establishing Expected Pattern Sequences. Following the strength of the symmetry of the relationship between security patterns resulted in establishing a meaningful pattern sequence from the kick-off pattern sequence. The pattern story was created to check the meaningfulness of the established pattern sequence.

All sequences of patterns established this way had to be evaluated first concerning the following properties before they could be declared as expected:

- All patterns used in this sequence are found to be useful during verification with the maximum likelihood estimation method [14].
- 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 this sequence would be used outside of the sequence, calculated as $p(X_1) * p(X_2) * ... * p(X_N)$
- All patterns in the sequence have the strongest symmetries of relationships between each other
- This pattern sequence can be described in a pattern story.

The probability of applying a pattern sequence Access Control Requirements → Single Access Point → Security Session → Role-Based Access Control can be calculated as $p(\mathit{Single\ Access\ Point} \mid \mathit{Access\ Control\ Requirements}) * p(\mathit{Security\ Session} \mid \mathit{Single\ Access\ Point}) * p(\mathit{Role{-}Based\ Access\ Control} \mid \mathit{Security\ Session}) * (\mathit{Authorization} \mid \mathit{Role} - \mathit{Based\ Access\ Control}) = 0.16666 * 0.2 * 0.25 * 0.33333 = 0.00277$. The probability of applying patterns Access Control Requirements, Single Access Point, Security Session, Role-Based Access Control, and Authorization outside of the pattern sequence is $p(\mathit{Access\ Control\ Requirements}) * p(\mathit{Single\ Access\ Point}) * p(\mathit{Security\ Session}) * p(\mathit{Role{-}Based\ Access\ Control}) * p(\mathit{Autho{-}rization}) = (1/106) * (1/106) * (1/106) * (1/106) * (1/106) = 0.0000000000747258173$. The probability of using this sequence is higher than the probability of using its patterns outside the sequence.

Based on the strongest symmetries of the relationships between patterns in the expected pattern sequence, the first the most expected pattern sequence in the pattern catalog of Cordeiro [3] is Access Control Requirements → Single Access Point → Security Session → Role-Based Access Control → Authorization.

Applying the same steps as in Section 3.1.1–Section 3.1.2 for the kick-off pattern sequence Access Control Requirements → Single Access Point → Security Session → Actor and Role Lifecycle → Actor-Based Access Rights → Administrator Object another meaningful expected pattern sequence was established: Access Control Requirements → Single Access Point → Security Session → Actor and Role Lifecycle.

Because during each identification of the strongest symmetry of relationship between a pattern from the expected pattern sequence candidate and another pattern links to weaker strengths of symmetries are identified, these weaker strengths of symmetries of relationships can be used to establish less meaningful unexpected pattern sequences. Applying the same steps as in Section 3.1.1–Section 3.1.2, the number of unexpected pattern sequences will be higher than the number of expected pattern sequences established from the same set of patterns.

Because a stochastic tree cannot be constructed for less than two patterns, applying the same steps as in Section 3.1.1-Section 3.1.2 leads to establishing expected pattern sequences consisting of at least two patterns.

3.2 Establishing Pattern Sequences Based on Bayesian Networks

Another way to establish pattern sequences is with the help of the Bayesian belief network from Barber [1]. The Bayesian belief network can be used to identify conditionally dependent and independent patterns in pattern sequences. The Bayesian belief network can be used to identify conditional dependence between more than two patterns. This conditional dependence can be used to identify patterns expected to have the strongest symmetry of relationships between. It is also assumed that conditionally independent patterns are expected to be used individually or avoided in the unexpected pattern sequences. Although Bayesian belief networks cannot express all conditional dependence and independence relationships between events such as the application of patterns, almost all graphical models according to Studeny [12] suffer from this disadvantage.

According to Barber [1] Bayesian belief networks are graphical models that can be used to model probability distributions. We can use the Bayesian belief network to identify candidate sequences expected to be used the most from the probability model of any pattern sequence.

The input to the second experiment with the Bayesian belief network was a set of text descriptions of patterns from a catalog of Cordeiro [3] that relate to each other in their text descriptions to establish meaningful kick-off pattern sequences. Text descriptions of patterns used in this second experiment are documented using the same pattern form provided by Cordeiro. The second experiment started by establishing a kick-off pattern sequence.

3.2.1 Establishing Kick-off Pattern Sequence. The following kick-off pattern sequence was extracted from the pattern story seen in the real fin-tech software company. This kick-off pattern sequence describes the software development process in this software company. Text descriptions of all patterns in this kick-off pattern sequence are present in the catalog of security patterns from Cordeiro [3]. During the second experiment with the Bayesian belief network, this kick-off pattern sequence was used to establish the most expected and unexpected pattern sequences.

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

It was found in Section 3.1, that Role-Based Access Control has the strongest symmetric relationship with the Authorization pattern, and because of this, the Authorization pattern is expected to be used the most after Role-Based Access Control. The problem with the kick-off pattern sequence is that Authorization has the strongest symmetric relationship with Role-Based Access Control.

Thus, no other pattern is expected to be used the most after these two. Because of this problem, another variable X can be introduced into the Bayesian belief network in Figure 7 describing this kick-off pattern sequence. The Bayesian belief network can be used to answer whether this kick-off pattern sequence is truly the most expected or if Role-Based Access Control is expected to be substituted with another pattern X in Figure 7 to remove this circular dependency.

The previous kick-off pattern sequence corresponds to the following probability distribution of the joint set of variables standing for patterns in the kick-off pattern sequence:

$$p(\mathit{Access\ Control\ Requirements}, \mathit{Single\ Access\ Point}, \mathit{Security\ Session}, X, \mathit{Role{-}Based\ Access\ Control}, \mathit{Autho{-}rization}) = p(\mathit{Authorization} \mid \mathit{Role{-}Based\ Access\ Control}, X) * p(\mathit{Role{-}Based\ Access\ Control} \mid \mathit{Security\ Session}) * p(\mathit{Security\ Session} \mid \mathit{Single\ Access\ Point}) * p(\mathit{Single\ Access\ Point} \mid \mathit{Access\ Control\ Requirements})$$

The model of this kick-off pattern sequence corresponds to the Bayesian belief network in Figure 7, where the source node of the directed arrow depicts the use of the security pattern from the kick-off pattern sequence before the use of another pattern from the kick-off pattern sequence depicted by target node pointed by this arrow. Because no security pattern was used twice in this kick-off pattern sequence, the Bayesian belief network in Figure 7 contains only unidirectional arrows. Node labeled as “Patterns X” stands for Anything, Property List, Manager, Check Point, Actor and Role Lifecycle, Front Door, Integration Reverse Proxy, Resource Lifecycle Manager security pattern that is mentioned in description of Security Session in pattern catalog of Cordeiro [3] which use can break a circular dependency between Role-Based Access Control and Authorization.

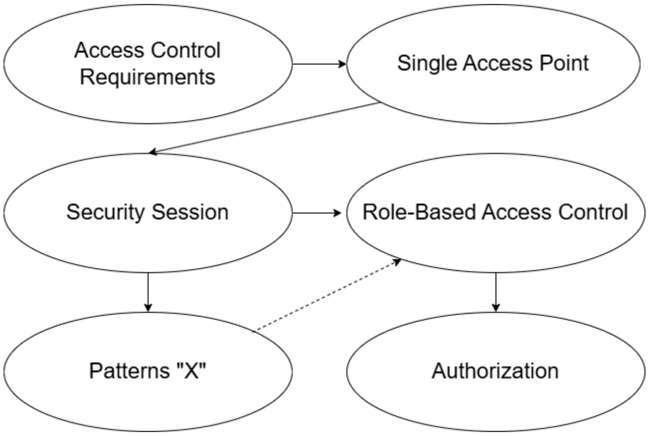

Figure 7: The Bayesian belief network for the kick-off pattern sequence

3.2.2 Establishing Expected Pattern Sequence. To break the circular dependency between Role-Based Access Control and Authorization, a pattern linked by Security Session or by Authorization has to be found if it is to replace Role-Based Access Control in the most expected pattern sequence established in Section 3.1 Apart from Role-Based Access Control, pattern Security Session links to 8 other patterns: Anything, Property List, Manager, Check Point, Actor and Role Lifecycle, Front Door, Integration Reverse Proxy, and Resource Lifecycle Manager. Not all of these patterns have their text descriptions present in the catalog of security patterns. Because none of these patterns are used in the kick-off pattern sequence, we cannot use the stochastic tree constructed in Section 3.1.1 to calculate the strength of symmetry of the relationship between the Security Session and another pattern from this list of patterns. Another stochastic tree for another kick-off pattern sequence would have to be constructed.

To test that Access Control Requirements → Single Access Point → Security Session → Role-Based Access Control → Authorization → Role-Based Access Control is the most expected pattern sequence composing these patterns (and not sequence from Section 3.1), probability of existence of relationship between pattern X (standing for one pattern from the list of 8 patterns) and pattern Authorization, given its existence results in applying one of these 8 patterns, needs to be calculated. If this probability is zero, Role-Based Access Control is not expected to be substituted between Security Session and Authorization. This probability can be calculated as:

$$p(\mathit{Authorization} = 1 \mid \mathit{Access\ Control\ Requirements},\ \mathit{Single\ Access\ Point},\ \mathit{Security\ Session} = 1, \mathit{Role{-}Based\ Access\ Control} = 0, \mathit{PatternX} = 1) = p(\mathit{Authorization} = 1,\ \mathit{Access\ Control\ Requirements},\ \mathit{Single\ Access\ Point}, \mathit{Security\ Session} = 1, \mathit{Role - Based\ Access\ Control} = 0,\ \mathit{PatternX} = 1) / p(\mathit{Authorization}, \mathit{Access\ Control\ Requirements}, \mathit{Single\ Access\ Point},\ \mathit{Security\ Session} = 1,\ \mathit{Role - Based\ Access\ Control} = 0, \mathit{PatternX} = 1)$$

This probability can be factorized as in Figure 8.

$$p(AUTH = 1 \mid ACR, SAP, SS = 1, RBAC = 0, X = 1) =$$

$$= \frac{p(AUTH{=}1,ACR,SAP,SS{=}1,RBAC{=}0,X{=}1)}{p(AUTH,ACR,SAP,SS{=}1,RBAC{=}0,X{=}1)} =$$

$$= \frac{p(AUTH = 1 \mid RBAC{=}0,X{=}1)*p(RBAC{=}0 \mid SS{=}1) *p(SS{=}1|SAP)*p(SAP|ACR)}{p(AUTH \mid RBAC{=}0,X{=}1)*p(RBAC{=}0 \mid SS{=}1) *p(SS{=}1|SAP)*p(SAP|ACR)} = 0$$

Figure 8: The probability of implicit relationship between pattern X and Authorization

The numerator of the last fraction is the factorization of four probability distributions. After calculating for $\mathit{Single\ Access\ Point} = 1$, $\mathit{Single\ Access\ Point} = 0$, $\mathit{Access\ Control\ Requirements} = 1$, and $\mathit{Access\ Control\ Requirements} = 0$, the resulting probability is zero because $p(\mathit{Authorization} = 1 | \mathit{Role - Based\ Access\ Control} = 0, X = 1)$ cannot be calculated if patterns X are not present in the kick-off pattern sequence. The value of the denominator does not matter because the numerator is zero. Because the probability of the existence of an implicit relationship between Role-Based Access Control and pattern X is zero, pattern sequence Access Control Requirements → Single Access Point → Security Session → Pattern X → Authorization is not the most expected pattern sequence.

Sequence Role-Based Access Control ⟶ Pattern X ⟶ Authorization renders eight unexpected pattern sequences.

3.3 The First Pattern Sequence

Pattern Role-Based Access Control in the Bayesian belief network in Figure 7 is a collider. The question is whether patterns Security Session and pattern X are conditionally dependent on pattern Role-Based Access Control, that is if $p(Security\ Session, X|Role - Based\ Access\ Control) = p(Security\ Session\ |\ Role{-}Based\ Access\ Control)p(X|Role - Based\ Access\ Control)$ applies. Patterns Security Session and X are conditionally dependent because Role-Based Access Control is a collider; conditional independence does not apply $p(Security\ Session,\ X|Role - Based\ Access\ Control) \propto p(Role - Based\ Access\ Control\ |\ Security\ Session, X)p(Security\ Session)p(X)$, and Role-Based Access Control is in conditioning set. Because of this, the Security Session is conditionally dependent on pattern X, given Role-Based Access Control.

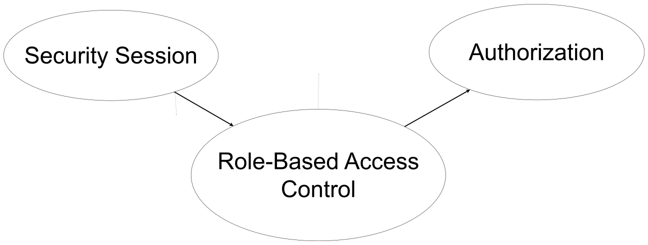

Figure 9: The first expected pattern sequence

Pattern Security Session is conditionally dependent on pattern X, given the subsequent application of pattern Role-Based Access Control. The existence of Security Session and Role-Based Access Control gives us additional information about pattern X, which is not expected to be applied between Security Session and Authorization because a symmetry of the relationship between Security Session and the other eight patterns linked by it cannot be calculated. The Security Session and pattern X are conditionally dependent, given Authorization in the kick-off pattern sequence. Security Session ⟶ Role-Based Access Control ⟶ Authorization in Figure 9 is the first the most expected pattern sequence established from the kick-off pattern sequence using the Bayesian belief network. This sequence can be described in pattern story.[6]

3.4 The Second Pattern Sequence

Pattern Single Access Point from Figure 7 is expected to be applied directly after Access Control Requirements and before Security Session because Access Control Requirements is conditionally dependent of Security Session, given Single Access Point. This is because $p(Access\ Control\ Requirements, Security\ Session|Single\ Access\ Point) \neq p(Access\ Control\ Requirements|Single\ Access\ Point) * p(Security\ Session|Single\ Access\ Point)$, and thus conditioning on a Single Access Point does not block the ability of Access Control Requirements to influence the application of Security Session after the Single Access Point. Another most expected pattern sequence in Figure 10 can be established: Access Control Requirements ⟶ Single Access Point ⟶ Security Session.

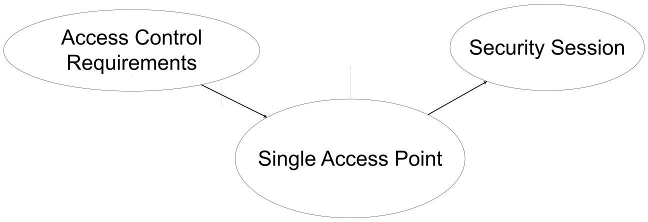

Figure 10: The second expected pattern sequence

The Access Control Requirements ⟶ Single Access Point ⟶ Security Session ⟶ Role-Based Access Control ⟶ Authorization is the expected pattern sequence composed of two shorter expected pattern sequences. This pattern sequence describes establishing a secure session before authorization and can be described in a pattern story.[7]

3.5 The Third Pattern Sequence

Pattern sequence Access Control Requirements ⟶ Single Access Point ⟶ Security Session ⟶ Role-Based Access Control ⟶ Authorization in Figure 11 is a pairwise Markov network with potentials (factors) defined over two cliques. This pattern sequence is the most expected, is meaningful and consists of two shorter expected pattern sequences established in Sections 3.3–3.4.

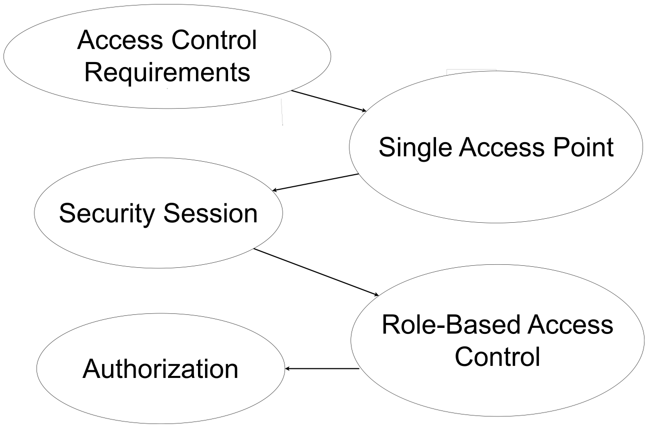

Figure 11: The third expected pattern sequence

[6] pattern story for this sequence can be seen at: https://github.com/viktorFIIT/fiit-research-resources/blob/main/stories/first-bayes-expected-pattern-sequence-story.png

[7] pattern story for this sequence can be seen at https://github.com/viktorFIIT/fiit-research-resources/blob/main/stories/second-bayes-expected-pattern-sequence-story.png

4 DISCUSSION

If that Policy-Based Access Control was part of the kick-off pattern sequence, then a different candidate of the expected pattern sequence would be selected, and a different expected pattern sequence would be established from it because the strength of symmetry of the relationship between Access Control Requirements and Policy-Based Access Control is $|0.01 - 0.0095| = 0.0005$. This value is smaller than the strength of symmetry of the relationship between Access Control Requirements and Single Access Point and therefore Policy-Based Access Control would be expected to be used next after Access Control Requirements, instead of Single Access Point.

Probabilities of subsequent use of patterns can also be provided by pattern users reflecting their actual use. The symmetry calculation of the relationship we used to establish the most expected pattern sequences is based on explicit pattern relationships. Another way how to calculate this strength of symmetry of the relationship between implicitly related patterns must be found.

5 EVALUATION

The three selected and most expected pattern sequences were evaluated successfully to check if there is only one way to establish these sequences using security patterns. If we were successful in the evaluation and could say that patterns in pattern sequences are not expected to be reordered, we proved the immutability property of the most expected pattern sequence. The three most expected pattern sequences were successfully evaluated to see if the transitive dependencies between patterns in these sequences exist. If we were successful in the evaluation and could say that patterns in pattern sequences have transitive dependencies, we said we proved the transitivity property of the expected pattern sequence.

5.1 Immutability of Expected Pattern Sequence

Successful evaluation of whether the established the most expected pattern sequences in the previous two experiments is the only way patterns in these sequences can be established was performed.

The problem with the circular path between Role-Based Access Control and Authorization cannot be solved by alternative pattern sequence Access Control Requirements → Single Access Point → Security Session → pattern X → Authorization, because Security Session does not link to any other pattern which would link to or would be linked by Authorization pattern. Bayesian belief network for:

- Access Control Requirements → Single Access Point → Security Session → Pattern X → Authorization contains two conditional independence statements.
- Access Control Requirements → Single Access Point → Security Session → Role-Based Access Control → Authorization does not contain conditional independence statement.

The Bayesian belief networks for these two pattern sequences are not Markov equivalent because these sequences do not represent the same sets of conditional independence statements. Because these sequences do not represent the same sets of conditional independence statements they also do not represent the most expected pattern sequences. Pattern sequence Access Control Requirements → Single Access Point → Security Session → Role-Based Access Control → Authorization is the most expected pattern sequence, while the second one with pattern X is not.

5.2 Transitivity of Expected Pattern Sequence

One chosen the most expected pattern sequence established using the Bayesian belief network: Access Control Requirements → Single Access Point → Security Session was successfully checked if it contains a transitive dependency between Access Control Requirements and Security Session. Proving the existence of the transitive dependency would result in finding that logical relationships between patterns in this expected pattern sequence are present.

If the application of Single Access Point after Access Control Requirements ACR → SAP and application of Security Session after Single Access Point SAP → SS is the most expected, then application of Access Control Requirements must be the precondition for Security Session. These two hypotheses can be tested using the Modus Ponens rule.

Modus Ponens rule [13], together with Bayesian statistics [1], can be used to answer the question of whther the application of an Access Control Requirements security pattern is a precondition for the application of a Security Session. According to Modus Ponens logical inference rule [13] if the following hypotheses are true, then the conclusion about transitive dependency between Access Control Requirements and Security Session must also be true:

- Access Control Requirements ⟹ Single Access Point corresponds to $p(SAP = tr|ACR = tr) = 1$ is the most expected because the Access Control Requirements has the strongest symmetric relationship with Single Access Point.
- Single Access Point ⟹ Security Session corresponds to $p(SS = tr|SAP = tr) = 1$ is the most expected because Single Access Point has the strongest symmetric relationship with Security Session.

If an application of Access Control Requirements is a precondition for the application of Security Session, then conclusion $(Security\ Session = true|Access\ Control\ Requirements = true) = 1$ must be true and $p(Security\ Session = true \mid Access\ Control\ Requirements = false) = 0$ must also be true. Probability $p(Security\ Session = true \mid Access\ Control\ Requirements = false) = 0$ can be calculated as follows:

p(Security Session = true | Access Control Requirements = false) = p(Security Session = true, Access Control Requirements = false) = p(Security Session = true, Access Control Requirements = false, Single Access Point=true) + p(Security Session = true, Access Control Requirements = false, Single Access Point=false)

A prerequisite for applying a Security Session after the Single Access Point is to have a basic security policy defined through Access Control Requirements. Because of this $p(Security\ Session = true, Access\ Control\ Requirements = false, Single\ Access\ Point = true) < p(Single\ Access\ Point = true, Access\ Control\ Requirements = false)$ and therefore $p(Security\ Session = true, Access\ Control\ Requirements = false, Single\ Access\ Point = true) = 0$. Subsequently, $p(Security\ Session = true, Access\ Control\ Requirements = false, Single\ Access\ Point = false) = 0$ because Single Access Point has the strongest symmetric relationship with Security

Session, meaning these two patterns are expected to be applied together. Because $p(Security\ Session = true \mid Access\ Control\ Requirements = false) = 0$, Access Control Requirements is the precondition for the application of Security Session. In other words, it can be implied that $p(Security\ Session = true \mid Access\ Control\ Requirements = true) = 1$.

Modus Tollens: The Modus Tollens [13] logical inference rule states that if the hypothesis is not true and an implication is true, the conclusion cannot be true. We have two hypotheses that are false statements:

- application of Access Control Requirements is not a precondition for application of Single Access Point. But it was found that $p(Single\ Access\ Point = true \mid Access\ Control\ Requirements = true) = 1$.
- application of Single Access Point is not a precondition for application of Security Session. But it was found that $p(Security\ Session = true | Single\ Access\ Point = true) = 1$.

The conclusion that the application of Access Control Requirements is the precondition for the application of Security Session must be true because according to Modus Tollens, two hypotheses are true. This implies Access Control Requirements ⟶ Single Access Point ⟶ Security Session is the most expected pattern sequence.

6 CONCLUSIONS AND FURTHER WORK

The two expected pattern sequences were established using the stochastic trees. Three expected pattern sequences were established using the Bayesian belief network. All six expected pattern sequences were found to be meaningful. All six expected pattern sequences could be described in a pattern story and were found to be used in the software company.[8] Eight unexpected pattern sequences were established using the Bayesian belief network and were not found to be meaningful. Stochastic trees and Bayesian networks can also be used to establish expected pattern sequences of any patterns, such as patterns for engineering software for the cloud documented by Sousa et al. [11].

The construction of stochastic trees and Bayesian belief networks will be automated using programming language to establish the most expected and unexpected pattern sequences. Established expected and unexpected pattern sequences will be grouped inside the same pattern language. In future work, a catalog of expected pattern sequences needs to be created to publish the outputs of these computations to a wider audience.

ACKNOWLEDGMENTS

We thank our EuroPLoP 2024 shepherd, Mary Tedeschi from City Tech - New York City College of Technology - CUNY, for valuable feedback.

The work reported here was supported by the Cultural and Educational Grant Agency of Slovak Republic (KEGA) under grant No. KG 014STU-4/2024.

REFERENCES

[1] D. Barber. 2011. *Bayesian Reasoning and Machine Learning.* Cambridge University Press, Cambridge, UK.

[2] James O. Coplien and Neil B. Harrison. 2004. *Organizational Patterns of Agile Software Development.* Prentice-Hall, Inc., USA.

[3] André Cordeiro, André Vasconcelos, and Miguel Correia. 2022. A Catalog of Security Patterns. https://hillside.net/plop/2022/papers/G1_P1.pdf To appear..

[4] Eduardo B. Fernandez, Nobukazu Yoshioka, Hironori Washizaki, and Michael VanHilst. 2010. Measuring the Level of Security Introduced by Security Patterns. In *2010 International Conference on Availability, Reliability and Security, (ARES).* IEEE, Prague, Czech Republic, 565–568.

[5] Erich Gamma, Richard Helm, Ralph Johnson, and John M. Vlissides. 1994. *Design Patterns: Elements of Reusable Object-Oriented Software.* Addison-Wesley Professional, USA.

[6] Hui Guan, Hongji Yang, and Jun Wang. 2016. An Ontology-Based Approach to Security Pattern Selection. *Int. J. Autom. Comput.* 13, 2 (2016), 168–182.

[7] Gordon Hazen. 1992. Stochastic Trees: A New Technique for Temporal Medical Decision Modeling. *Medical decision making : an international journal of the Society for Medical Decision Making* 12 (08 1992), 163–78.

[8] A. Kubo, H. Washizaki, A. Takasu, and Y. Fukazawa. 2005. Analyzing relations among software patterns based on document similarity. In *International Conference on Information Technology: Coding and Computing (ITCC'05).* IEEE, Washington, USA, 298–303 Vol. 2.

[9] Anas Motii, Brahim Hamid, Agnès Lanusse, and Jean-Michel Bruel. 2015. Guiding the selection of security patterns based on security requirements and pattern classification. In *Proceedings of the 20th European Conference on Pattern Languages of Programs* (Kaufbeuren, Germany) *(EuroPLoP '15).* Association for Computing Machinery, New York, NY, USA, Article 10, 17 pages.

[10] Alireza Shameli-Sendi, Yosr Jarraya, Makan Pourzandi, and Mohamed Cheriet. 2019. Efficient Provisioning of Security Service Function Chaining Using Network Security Defense Patterns. *IEEE Transactions on Services Computing* 12, 4 (2019), 534–549.

[11] Tiago Boldt Sousa, Hugo Sereno Ferreira, and Filipe Figueiredo Correia. 2022. A Survey on the Adoption of Patterns for Engineering Software for the Cloud. *IEEE Transactions on Software Engineering* 48, 6 (2022), 2128–2140.

[12] Milan Studeny. 2001. On non-graphical description of models of conditional independence structure.

[13] Indiana The Purdue University, Department of Computer Science. 2007. Proof Techniques. https://www.cs.purdue.edu/homes/spa/courses/cs182/mod3.pdf

[14] The Pennsylvania State University. 2023. Maximum Likelihood Estimation. https://online.stat.psu.edu/stat415/lesson/1/1.2

[8] pattern stories for all established expected pattern sequences can be seen at https://github.com/viktorFIIT/fiit-research-resources/tree/main/stories