Documentation

Friday, May 3, 2024

1:04 PM

## Use on Organizational Patterns

Let’s assume sequence of organizational patterns from the Piecemeal Growth pattern language. Let’s denote names of the patterns in this sequence with first letters of the words in their names:  
  
SST – denotes Self Selecting Team organizational pattern.  
DG - denotes Diverse Groups organizational pattern.

UP - denotes Unity of Purpose organizational pattern.

PR - denotes Patron Role organizational pattern.

FW - denotes Fire Walls organizational pattern.

GK - denotes Gate Keeper organizational pattern.

CS - denotes Compensate Success organizational pattern.

STO - denotes Size the Organization organizational pattern.

PII – denotes Phasing It In organizational pattern.

APP - denotes Apprenticeship organizational pattern.

SV - denotes Solo Virtuoso organizational pattern.

DIP - denotes Developing in Pairs organizational pattern.

HD - denotes Holistic Diversity organizational pattern.

DER - denotes Domain Expertise in Roles organizational pattern.

SBS - denotes Subsystem by Skill organizational pattern.

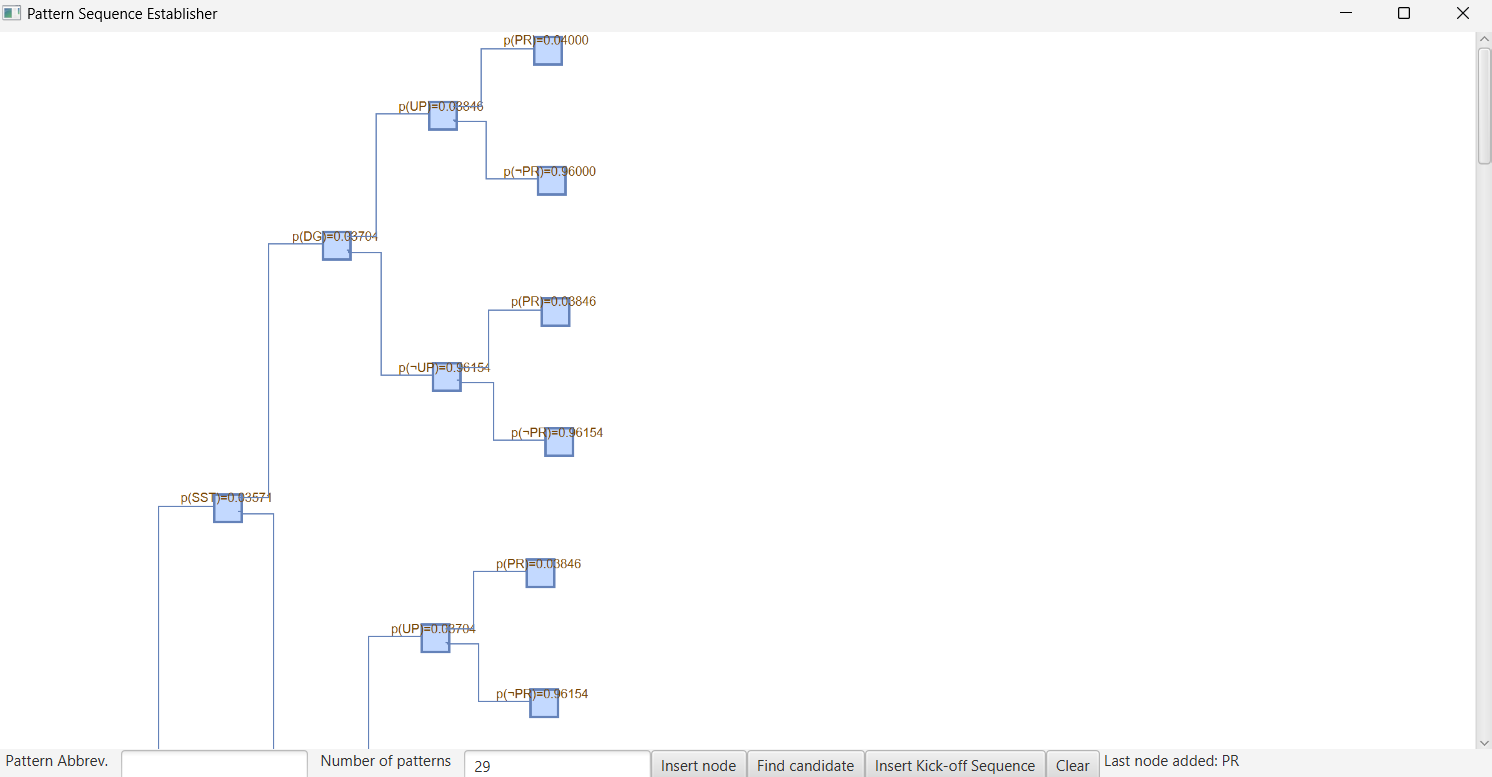
MTN - denotes Moderate Truck Number organizational pattern.

**Kick-off sequence 1:** SW -> SST -> DG -> UP -> PR -> FW -> GK -> CS -> STO -> PII -> APP -> SV -> DIP -> HD -> DER -> SBS -> MTN

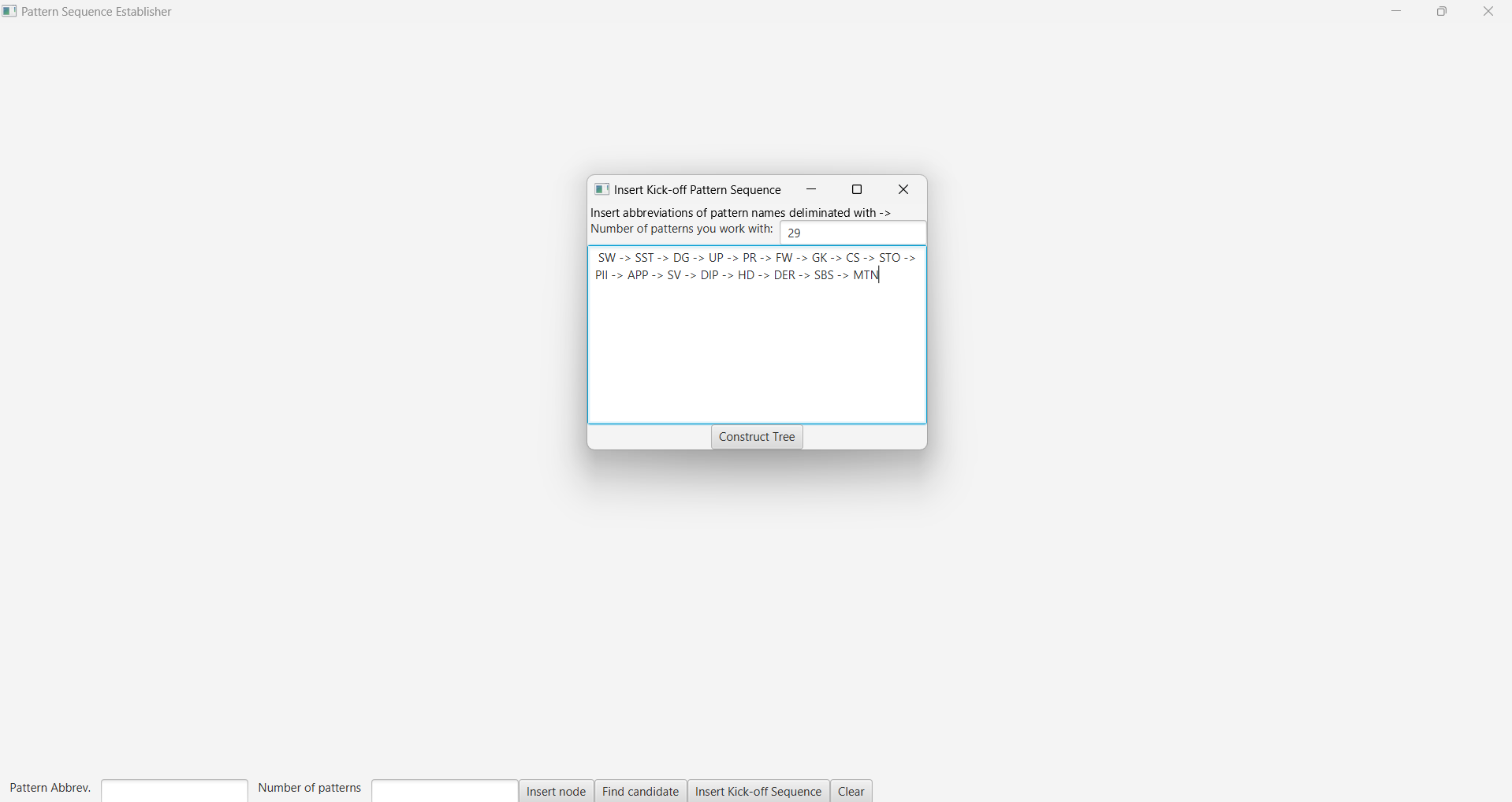
This sequence is meaningful, its use can be explained in pattern story and describes transformation of the small team experimenting with new technology into the wider organization structure after initial success of the software prototype they were responsible for.

262 141 nodes are required to implement stochastic tree for this sequence, if each node has to hold information about all its parents, along with probability of the use of each of these nodes representing pattern sequence.

Users of the application have the option to insert new nodes into the stochastic tree one by one by inserting pattern name abbreviation into the relevant text field labeled “Pattern Abbrev.”. Value into the text field labeled “Number of patterns” must be inserted as well and it’s the number of patterns user works with. Each node is added into the tree after clicking on 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 because non-editable.

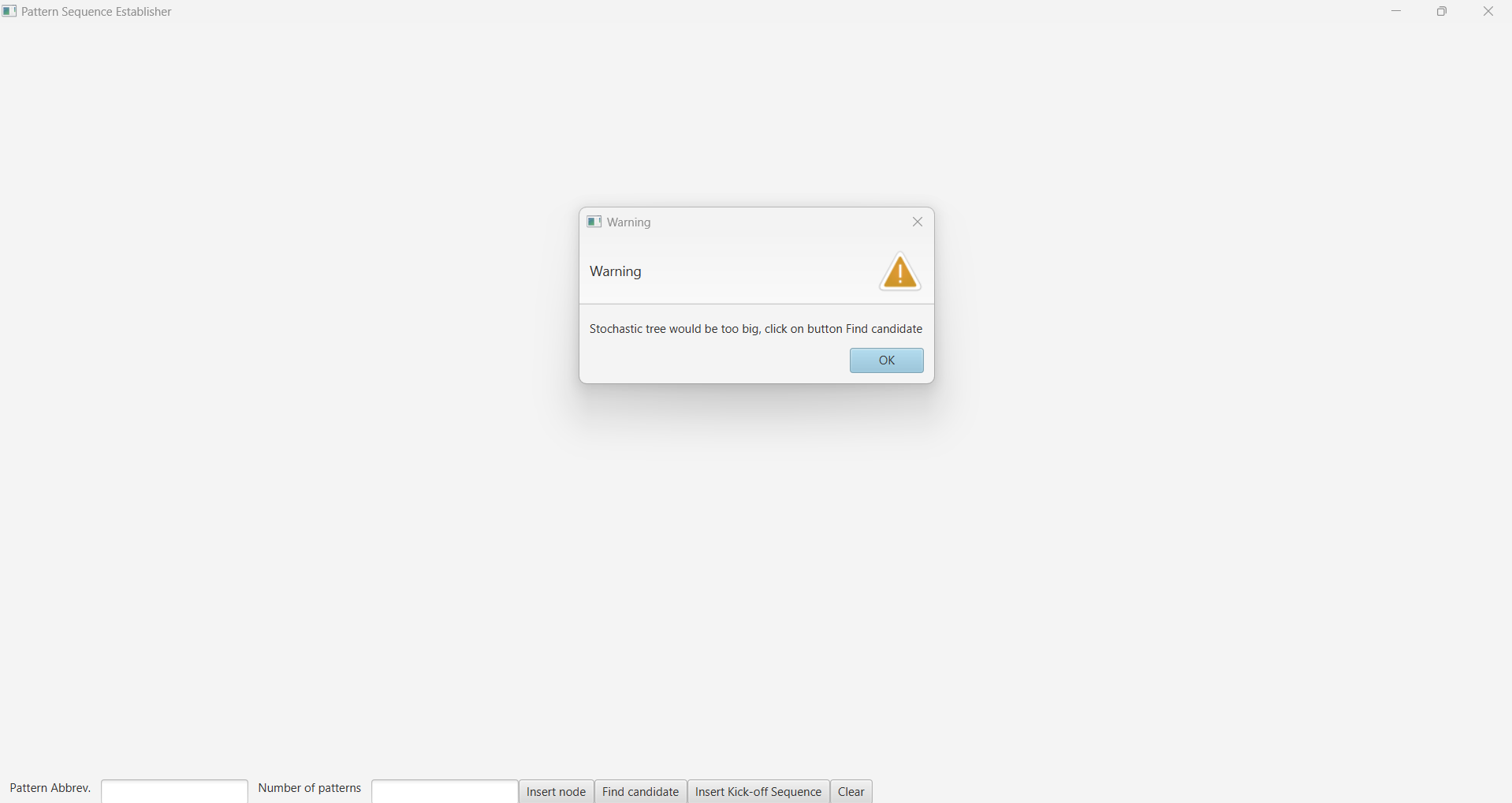


In order to speed up inserting kick-off sequences and constructing stochastic trees on top of them, application provides option to batch upload them through the text area displayed after the click on “Insert Kick-off Sequence” button.

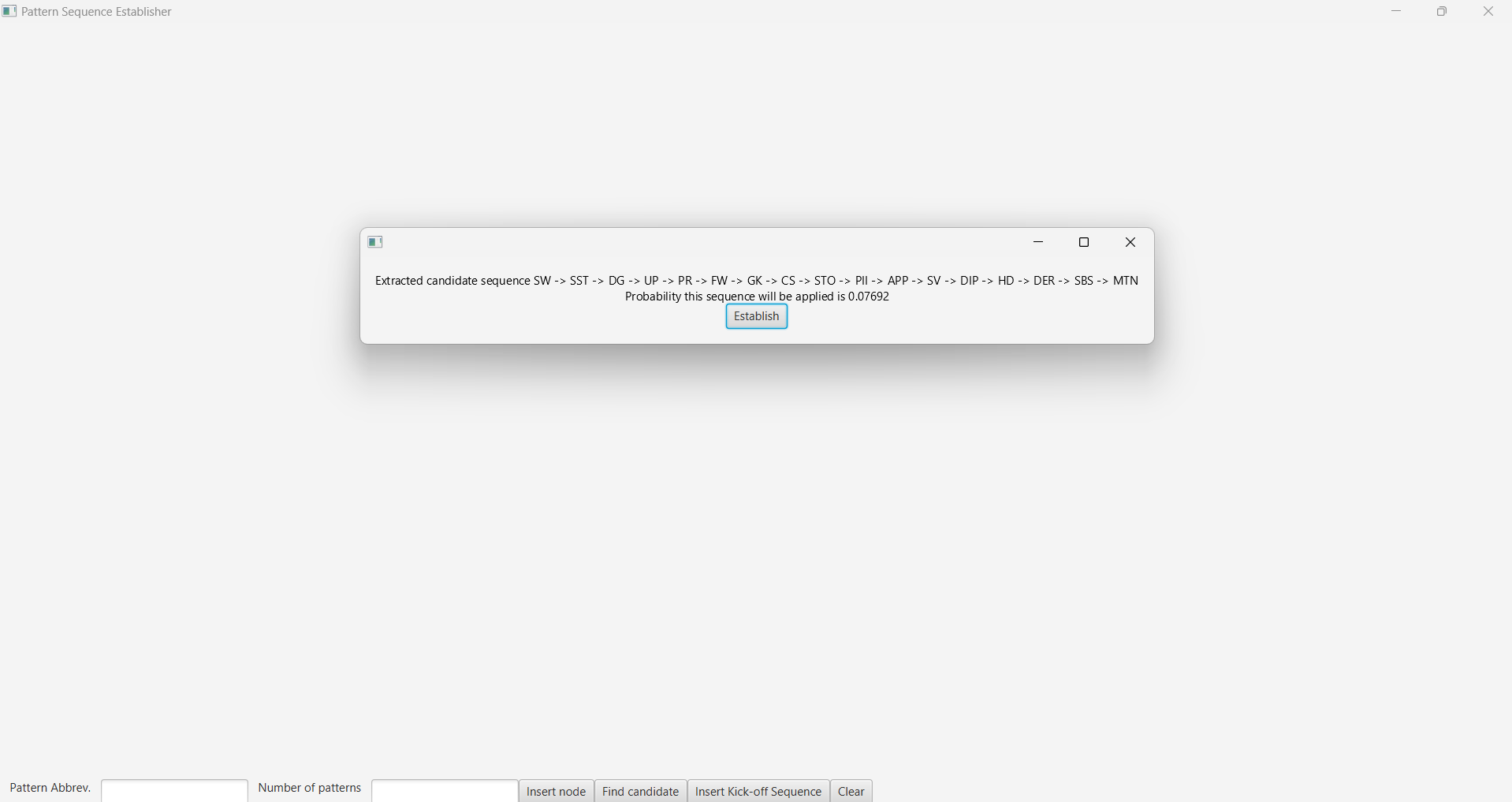


Because our kick-off sequence has 17 patterns, stochastic tree for this sequence would be too big to be displayed with JGraphT graphic library. This library only allows to draw a binary tree and functionality behind the stochastic tree must be implemented by my own. This graphic library is also very slow when it comes to loading a big graph, therefore, other graphic libraries should be explored as well.

Stochastic trees are used only to find an expected pattern sequence candidate. If the visualization of the stochastic tree would be too big to be visualized, application stops to visualize it, provides warning about what happened and let’s user to be shown with the expected pattern sequence candidate, in a matter of seconds. The structure of the stochastic tree is this time constructed but visualization is stopped.

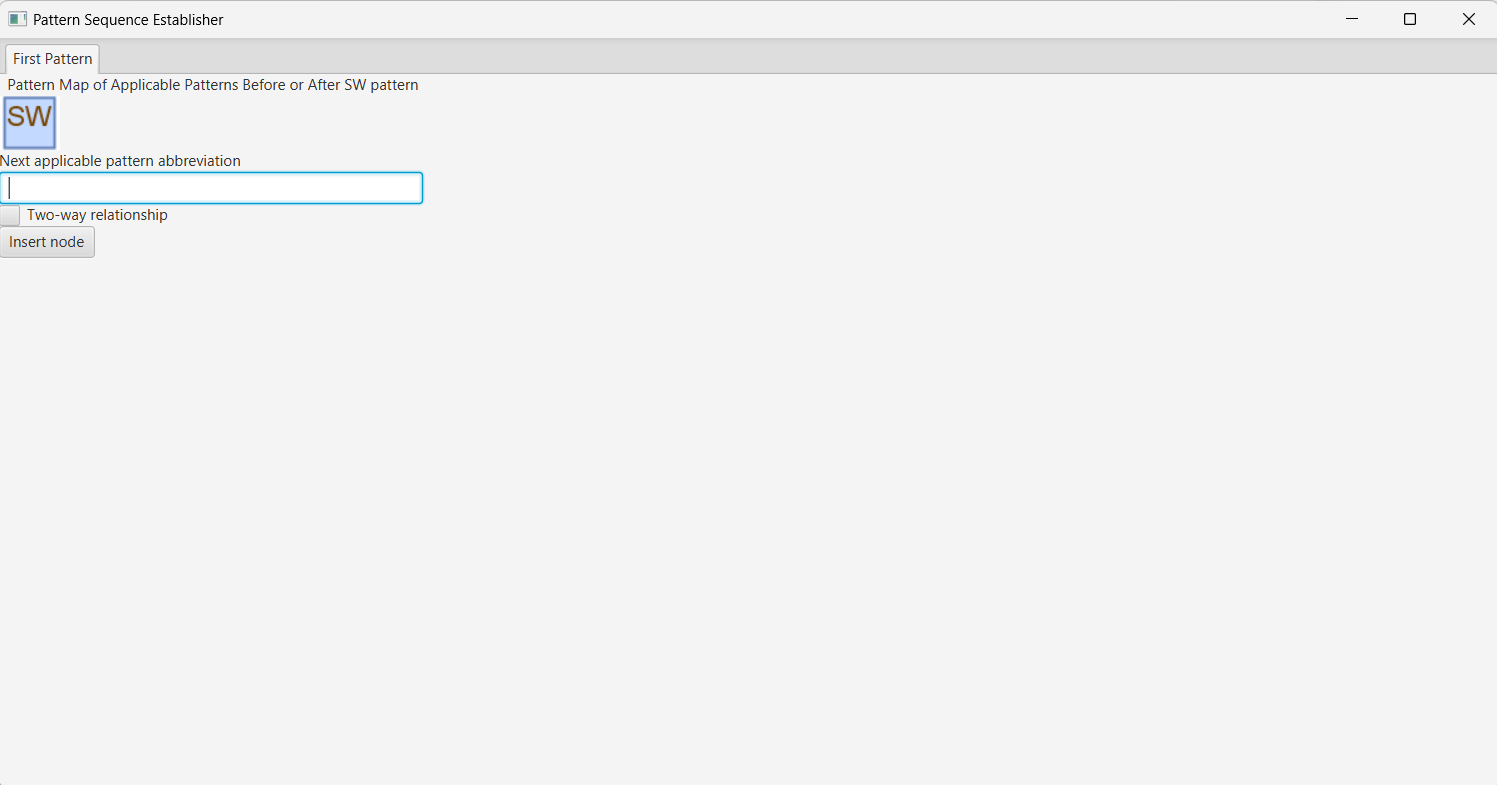


As can be seen on the bottom picture, application displays expected pattern sequence candidate along with the probability of applying it by the assumed domain expert.



After clicking on button “Establish” user is redirected to new tab where he can add additional applicable patterns after the first pattern in the expected pattern sequence candidate into the pattern map of applicable patterns. Abbreviations for these patterns must be inserted into the text field labeled “Next applicable pattern abbreviation”. If this second pattern links to the first pattern in candidate sequence in its text description, checkbox “Two-way relationship” should be checked.

Each relationship between first pattern in expected pattern sequence candidate and additional applicable pattern is added into the pattern map after click on ‘Insert node’ pattern.



After each addition of relationship to pattern map, new row is added to the table with the conditional probabilities and symmetry of relationships. This table is shown on the right side of the tab. Table has four columns:

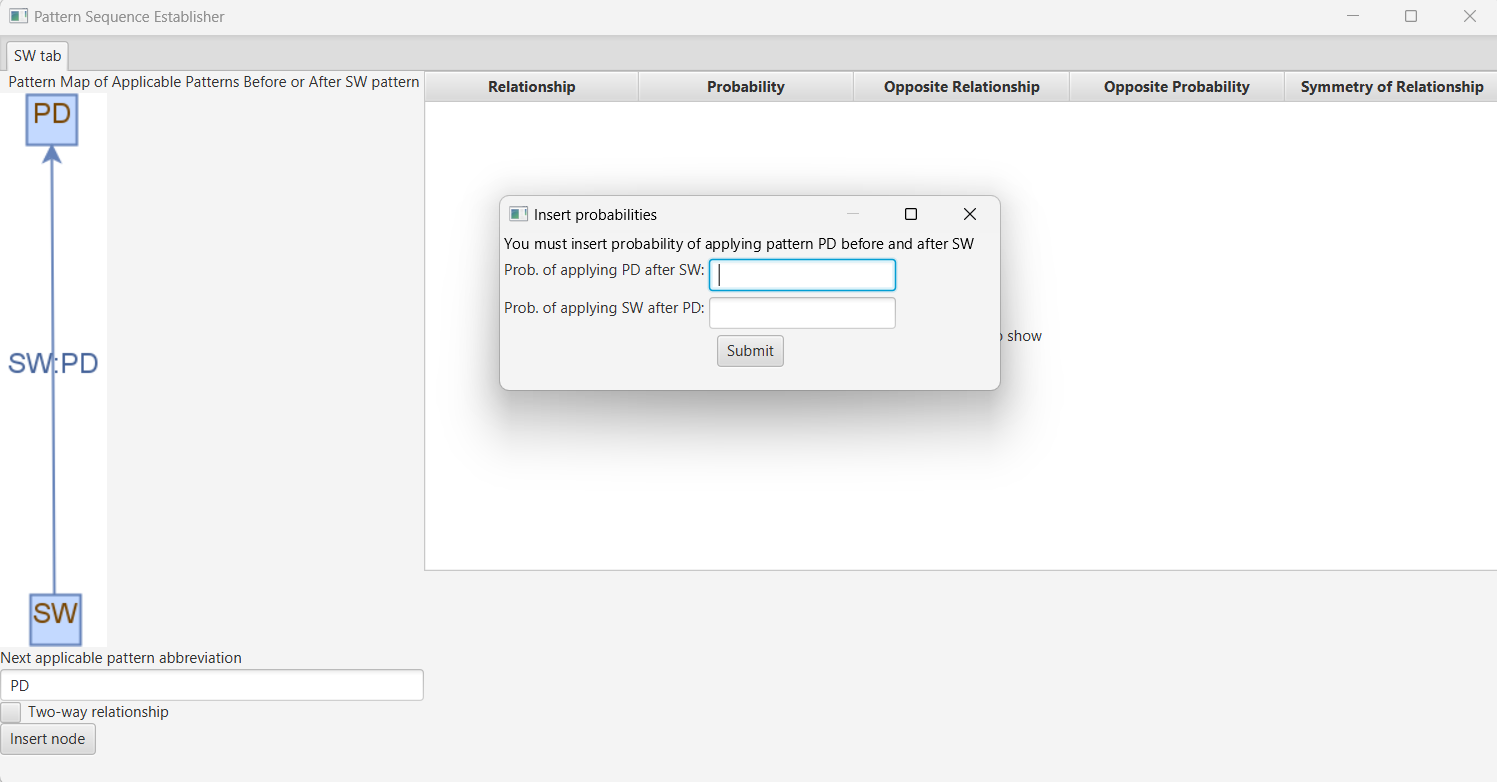
* A column named “Relationship” indicates the first pattern in the expected pattern sequence candidate links to additional pattern provided by user of this application in its text description.
* A column named “Probability” provides conditional probability of applying this additional applicable pattern after the first pattern in expected pattern sequence candidate. This probability is extracted from the stochastic tree.
* A column named “Opposite Relationship” provides inverse probability to the conditional probability calculated with Bayes rule. It is a conditional probability of applying an additional applicable pattern before application of first pattern in expected pattern sequence candidate.
* A column named “Symmetry of Relationship” provides symmetry of relationship between the first pattern in the expected pattern sequence candidate and each additional applicable pattern. This value is calculated as the absolute value of diffference between values in columns “Probability” and “Opposite probability”.

Pattern expected to be applied after the first pattern in the expected pattern sequence candidate is the one having the strongest symmetry (lowest number in the column “Symmetry of relationship”) of relationship with the first pattern in candidate sequence. Name of this next expected pattern is provided to user after the click on button “Find next applicable pattern”.

A screenshot of a computer

Description automatically generated

If the next applicable pattern is not present in the kick-off pattern sequence, conditional probability for column “Probability” cannot be extracted from the stochastic tree and different stochastic tree for different kick-off pattern sequence had to be constructed. Because of that application prompts user to skip this step and provide value for column “Probability” and “Symmetry of Relationships” directly, such that symmetry of relationship can be computed.



After clicking on button “Continue”, user is redirected to another tab where he continues the same. He is asked to provide other applicable patterns but this time, after the pattern that was previously identified as pattern expected to be applied next (after SW, and it was Self Selecting Team). Symmetries of relationships are calculated again and provided in the table on the right-hand side. This time, the next applicable pattern was identified as DG = Diverse Groups. This makes an expected pattern sequence SW -> SST -> DG standing for Skunk Works -> Self-Selecting Team -> Diverse Groups.

A screenshot of a computer

Description automatically generated

User can also get back to previous tabs anytime he wants to see what led to decision to recommend next applicable pattern.

A screenshot of a computer

Description automatically generated

## Use on Security Patterns

Let’s assume sequence of security patterns from the catalog of security patterns. Let’s denote names of the patterns in this sequence with first letters of the words in their names:

ACR – stands for Access Control Requirements.

SAP – stands for Single Access Point.

SS – stands for Security Session.

RBAC – stands for Role-Based Access Control.

AUTH – stands for Authorization.

ACL – stands for Access Control List.

PBAC – Policy-Based Access Control.

**Kick-off sequence 1:** ACR -> SAP -> SS -> RBAC -> AUTH -> ACL -> PBAC

User clicks on the “Insert Kick-off Sequence” button and inserts number of the patterns in catalog of security patterns and kick-off pattern sequence where patterns have abbreviated names. User constructs the stochastic tree for this kick-off pattern sequence by clicking on button “Construct Tree”.

A screenshot of a computer

Description automatically generated

Once the stochastic tree is constructed user can search for the expected pattern sequence candidate on his own or he can speed up this process by clicking on button “Find candidate”.

A screenshot of a computer

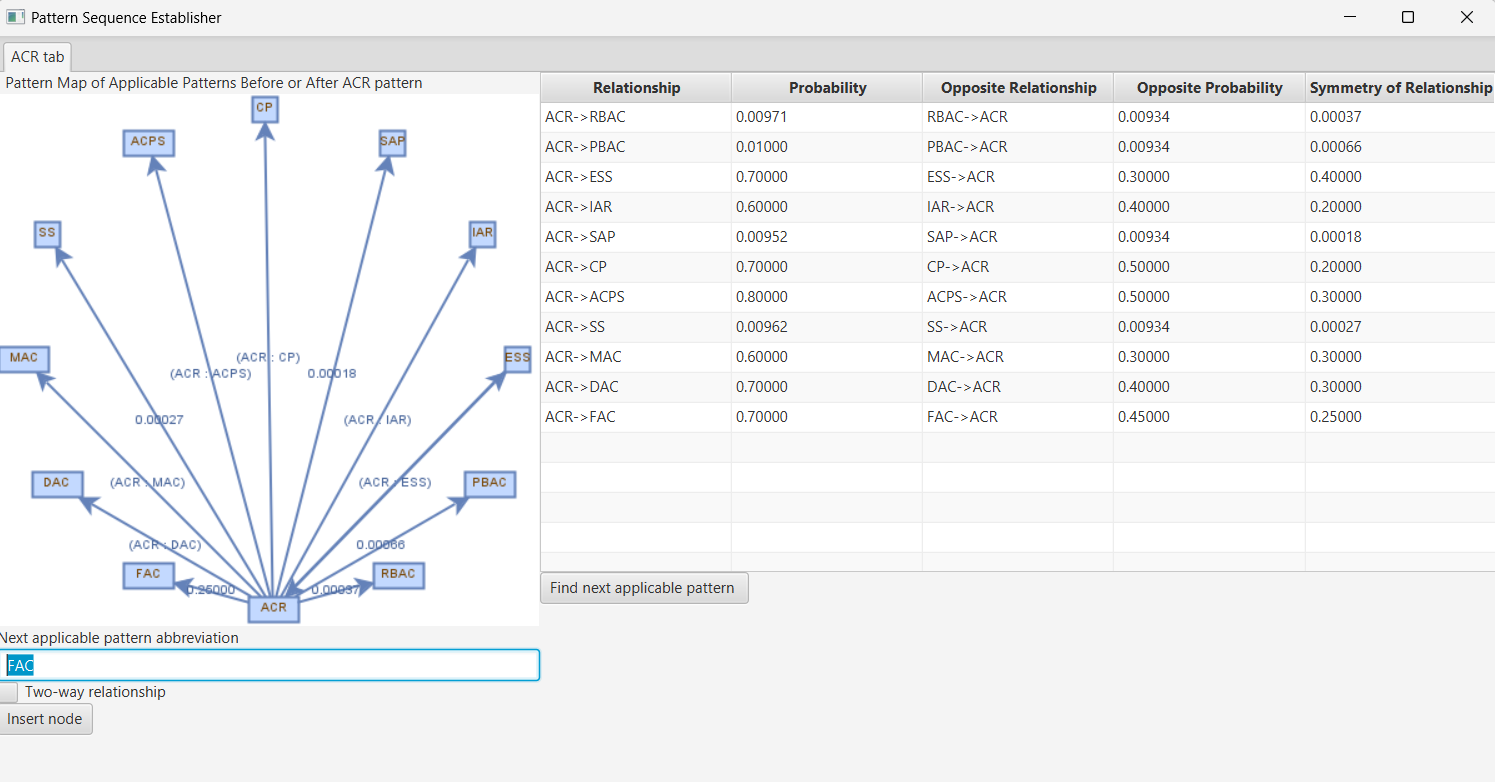
Description automatically generated

After that, the expected pattern sequence candidate is displayed to the user along with probability of applying it. Users can start establishing expected pattern sequence from this sequence by clicking on button “Establish”.

A screenshot of a computer

Description automatically generated

User constructs pattern map of applicable patterns for first pattern in the expected pattern sequence candidate in the first tab. User is provided with pattern expected to be applied next after the first pattern in candidate sequence after clicking on “Find next applicable pattern” button. For patterns that are not present in the kick-off pattern sequence and there were identified in pattern map as applicable, probability of applying them before and after the first pattern in candidate sequence must be provided in modal window, such that symmetry of relationship can be computed.



After clicking on button labeled “Find next applicable pattern”, SAP was shown standing as abbreviation for Single Access Point security pattern that is expected to be applied after the ACR or Access Control Requirements. User can now continue to establish expected pattern sequence by clicking on button “Continue”.

A screenshot of a computer

Description automatically generated

After clicking on button labeled “Find next applicable pattern” another expected pattern can be provided. This time it was security pattern abbreviated as SS that stands for Security Session pattern.

A screenshot of a computer

Description automatically generated

If the next applicable pattern is not in the kick-off pattern sequence, application asks for conditional probabilities:

A screenshot of a computer

Description automatically generated