

SVIPEX – Tutorial

Florian Stertz¹, Karolin Winter¹, Stefanie Rinderle-Ma^{1,2}

¹Research Group Workflow Systems and Technology, Faculty of Computer Science,

²Data Science@Uni Vienna, University of Vienna, Vienna, Austria
{firstname.lastname}@univie.ac.at

Note that the results of ISC discovery remain visible to other users after the analysis and are deleted on a daily basis. Do not upload data that is subject to privacy.

SVIPEX is a lightweight web service implementing the four ISC discovery algorithms presented in [2] and available at

<http://svipex.wst.univie.ac.at>

A screencast is available at

<https://ucloud.univie.ac.at/index.php/s/p00h1lwuRIS3xsT>

SVIPEX takes process execution logs in XES format¹ as input. Each ISC discovery algorithm returns a set of ISC candidates which are depicted as red lines within or across the underlying process models which were discovered simultaneously using the Heuristics Miner (cf. [1]). The algorithms were developed based on the following ISC categorization.

- I simultaneous execution of events \mapsto Algorithm 1
- II constrained event execution based on data elements, time constraints or regularities \mapsto Algorithm 2
- III order of events \mapsto Algorithm 3
- IV non concurrent execution of events. \mapsto Algorithm 4

SVIPEX user interface consists of two pages, one for displaying the results and a second page for uploading new datasets.

1 Main Page

The main page, depicted in Fig. 1, is displayed when accessing the website, i.e., serves as start page. It contains information on, within a previous publication [2], analyzed datasets (*Manufacturing*, *Printer* and *HigherEducationDomain*), displayed on the left. If a user adds a new dataset it will be displayed below *HigherEducationDomain*.

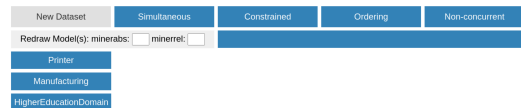


Fig. 1: Screenshot – Main Page on Initial Call

¹ <http://www.xes-standard.org>

The figure shows four sequential screenshots of the 'Upload a new dataset' interface:

- Step 1: File Upload**: Shows a 'Dataset Name' input field, a 'Logs in XES Format' section with a 'Dateien auswählen' button and 'Keine ausgewählt' text, and a 'Mergeattribute' dropdown set to 'optional'. 'Previous' and 'Next' buttons are at the bottom right.
- Step 2: Algorithm Selection**: Shows four checkboxes for 'Algorithm 1', 'Algorithm 2', 'Algorithm 3', and 'Algorithm 4'. 'Algorithm 1' and 'Algorithm 3' are checked. 'Previous' and 'Next' buttons are at the bottom right.
- Step 3: Parameter (ISC Mining)**: Shows input fields for 'Gamma 1' (0,95), 'Epsilon 1' (0,1), 'Gamma 2' (0,95), 'Limit 2' (optional), 'Gamma 3' (0,95), 'Kappa' (0,05), and 'Epsilon 4' (0). 'Previous' and 'Next' buttons are at the bottom right.
- Step 4: Parameter (Heuristics Mining)**: Shows input fields for 'Absolute Number of Edges' (1) and 'Relative Number of Edges' (0,3). A red warning note states: 'Note that the results of ISC discovery remain visible to other users after the analysis and are deleted on a daily basis. Do not upload data that is subject to privacy.' 'Previous' and 'Submit' buttons are at the bottom right.

Fig. 3: Screenshots – Upload Page

- Whenever ISC span multiple process types, traces or events should have a *unique case identifier*, i.e., an attribute that enables the merging of traces stemming from different processes, e.g., a customer id (called merge attribute).

The latter is called mergeattribute and is optional. If more than one XES file is given, a mergeattribute is automatically detected by SUIPEX under the assumption that there exists an event attribute that is unique in terms of values, i.e., the last requirement imposed on the log files is fulfilled. The source code as well as example data sets can be downloaded at

<http://gruppe.wst.univie.ac.at/projects/crisp/index.php?t=discovery>

Step 2 – Algorithm Selection

Within this step a user can select which algorithms shall be applied whereas at least one of them must be chosen. Algorithm 3 and 4 are only applicable for a multiple process setting, i.e., more than one XES file must be uploaded.

Step 3 – Parameter (ISC Mining)

Within this step a user can set the Parameters for the ISC mining algorithms. In particular, for

- Algorithm 1: $\gamma_1 \in [0, 1]$ and $\varepsilon_1 \in \mathbb{R}_0^+$ in seconds

- Algorithm 2: $\gamma_2 \in [0, 1]$ and $\varepsilon_2 \in \mathbb{R}_0^+$ in seconds
- Algorithm 3: $\gamma_3 \in [0, 1]$ and $\kappa \in [0, 0.5)$
- Algorithm 4: $\varepsilon_4 \in \mathbb{R}_0^+$ in seconds.

All γ parameters serve for filtering relative occurrences, e.g., if the parameter is set to 0.9, the observation must have been made for at least 90% of all cases. The $\varepsilon_j, j = 1, 4$ parameters can be used to compensate logging errors in seconds whereas ε_2 represents an activity execution time limit. Parameter κ represents the number of inverse orders that may occur. For further details the reader is referred to [2].

Note that each of these parameters has a default value based on experience from previous experiments in order to assist users who are not yet familiar with SUIPEX.

Step 4 – Parameter (Heuristics Miner)

Since SUIPEX uses the Heuristics Miner [1] for discovering process models, two parameters need to be set by the user. On the one hand a threshold for the absolute number of occurrences of an edge, on the other a threshold for the relative number of occurrences of an edge.

Step 5 – Submit

At the end of step 4 by clicking onto the blue submit button the mining process is started and the user is redirected to the main page. As soon as the mining process is finished, the results can be displayed by refreshing the website. Have fun discovering ISC in your process execution logs!

Acknowledgment

This work has been partly funded by the Austrian Research Promotion Agency (FFG) via the “Austrian Competence Center for Digital Production” (CDP) under the contract number 854187. This work has been supported by the Pilot Factory Industry 4.0, Seestadtstrasse 27, Vienna, Austria.

References

1. Weijters, A., van Der Aalst, W.M., De Medeiros, A.A.: Process mining with the heuristics miner-algorithm. Technische Universiteit Eindhoven, Tech. Rep. WP **166**, 1–34 (2006)
2. Winter, K., Stertz, F., Rinderle-Ma, S.: Discovering instance and process spanning constraints from process execution logs. *Inf. Syst.* **89**, 101484 (2020). <https://doi.org/10.1016/j.is.2019.101484>