SMARTRPA
USER MANUAL

Web Documentation

OUTLINE

Architecture

Action Logger

Dialog Windows

Bot Execution

Robotic Process Automation (RPA) is a technology which automates mouse and keyboard interactions by means of a software (SW) robot to remove intensive routines. The current generation of RPA tools is driven by predefined rules and manual configurations made by expert users rather than automated techniques.

SmartRPA is a cross-platform tool that tackles such issues. It allows to easily record event logs and to automatically generating executable RPA scripts that will drive a SW robots in emulating an observed user behavior (previously recorded in dedicated UI logs) during the enactment of a routine of interest.

SMARTRPA ARCHITECTURE

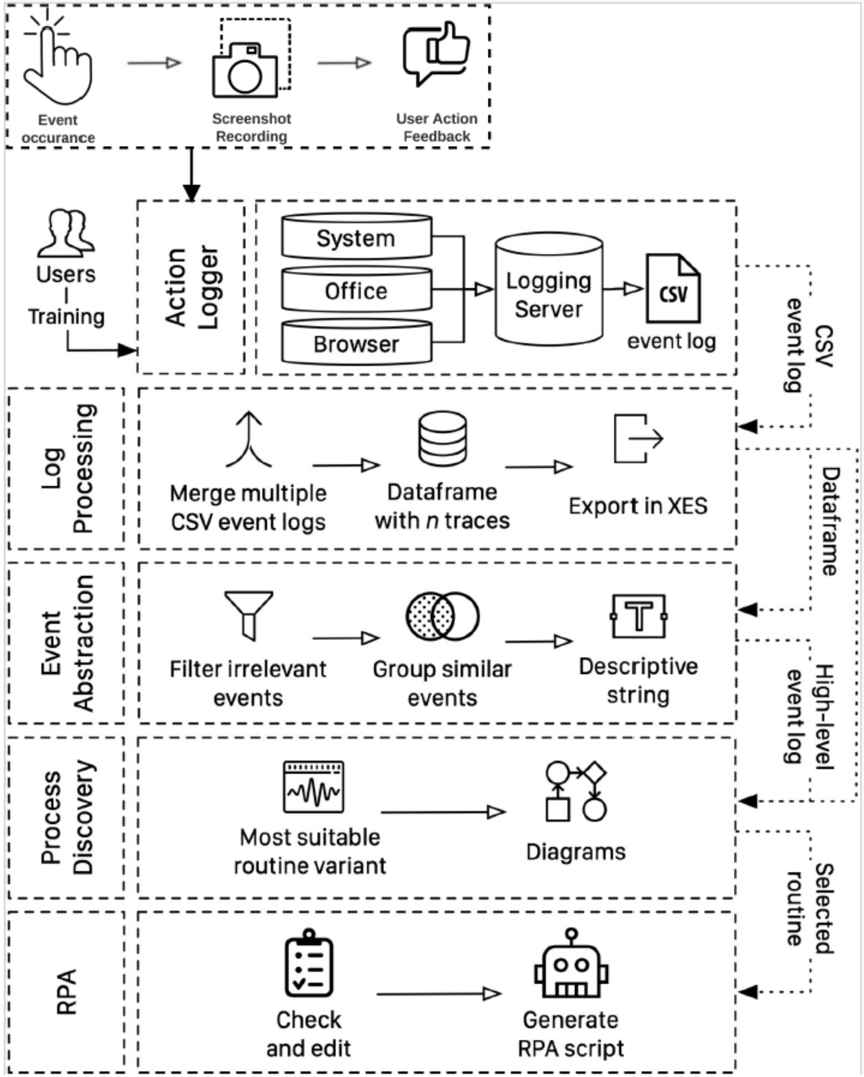
• **Action Logger**, log user behaviour, take screenshots, tag actions, supports wide range of applications, cross-platform;

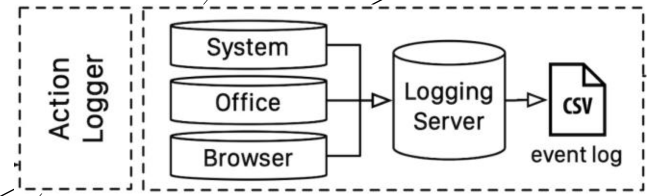
• **Log Processing**, generates both CSV and XES event log;

• **Event abstraction**, abstracts events to a higher level;

• **Process Discovery**, selects the most suitable routine variant to automate and generates high-level flowchart diagram, thus skipping completely the manual modeling activity;
•**Decision Points**, discover differencies between multiple traces in a process and build a new routine based on user decisions;

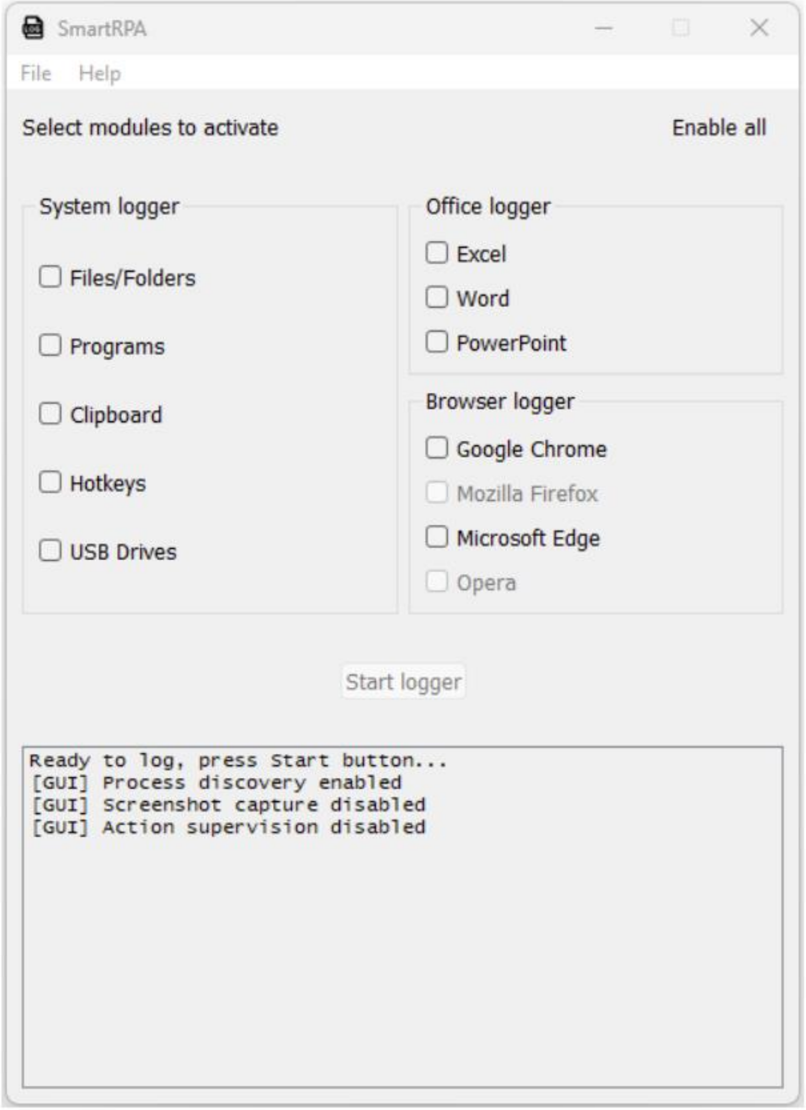
• **RPA**, implements and enacts a SW robot emulating a routine reflecting the observed behavior (either the most frequent one or the one based on decision points). Available both as a cross-platform Python script and as a UiPath project.





USER INTERFACE

- FileMenu : Opening the menu to set the settings or choose additional functionality of SmartRPA.
- Help Menu: Shortcut to get to the SmartRPA GitHub documentation.
- Enable All: Selects all possible modules to be activated.
- Start Logger Button: Can be clicked once at least
  one module has been activated.
- SmartRPA notification panel: Displayes messages about Logging Server events



Select modules to activate

System logger

□ Files/Folders

□ Programs

□ Clipboard

□ Hotkeys

□ USB Drives

Office logger

□ Excel

□ Word

□ PowerPoint

Browser logger

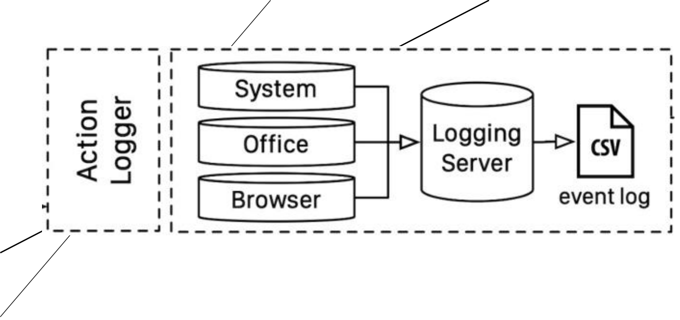
□ Google Chrome

□ Mozilla Firefox

□ Microsoft Edge

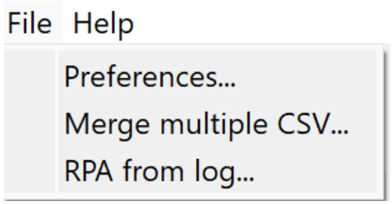
□ Opera

```
Ready to log, press Start button...
[GUI] Process discovery enabled
[GUI] Screenshot capture disabled
[GUI] Action supervision disabled
```



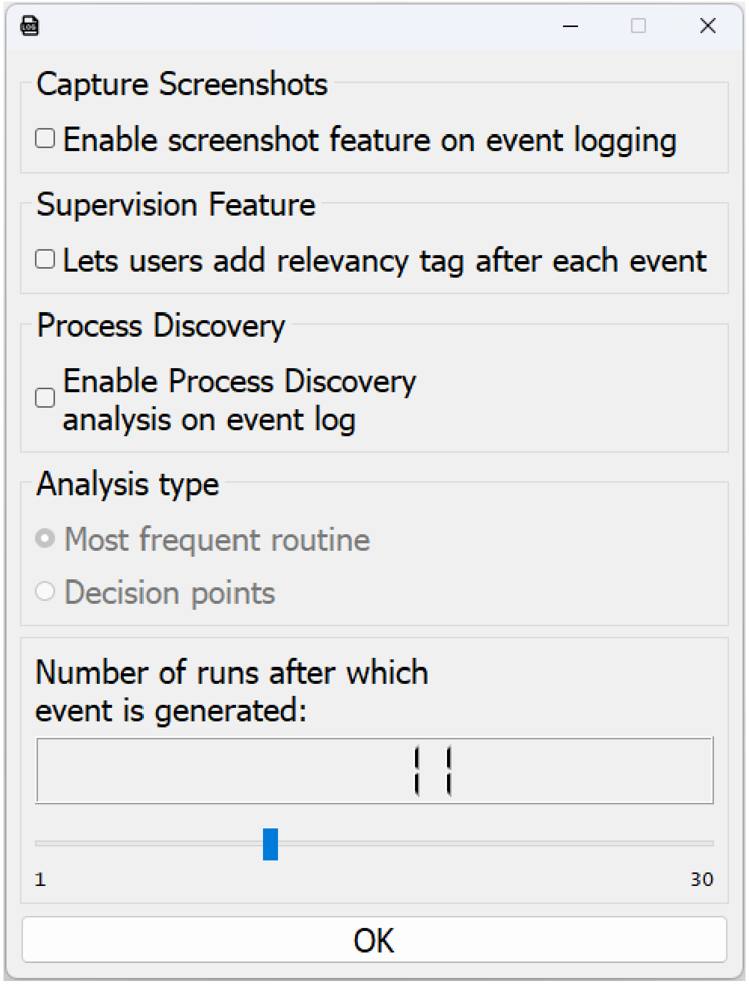
FILE MENU

- **Preferences**: Setting the recording and discovery preferences for the logging and process model discovery process.



- **Merge multiple CSV**: Allows the user to select multiple CSV files in a folder that will be merged into a single CSV file. Once the logs are selected the process analysis will be conducted and the process mining is started.
- **RPA from Log**: The user can select a single log file, which is used for process discovery (segmentation and variant discovery). The RPA bot will be created as Python file and UiPath file in the folder "RPA"

PREFERENCES



Capture Screenshots

□ Enable screenshot feature on event logging

Supervision Feature

□ Lets users add relevancy tag after each event

Process Discovery

□ Enable Process Discovery analysis on event log

Analysis type

● Most frequent routine

○ Decision points

Number of runs after which event is generated:

11

1 30

OK

1. Capture Screenshots: Activating this feature allows the user to capture a screenshot once an event is triggered. The screenshots for each event will be stored in the folder "screenshots" with the event log name.

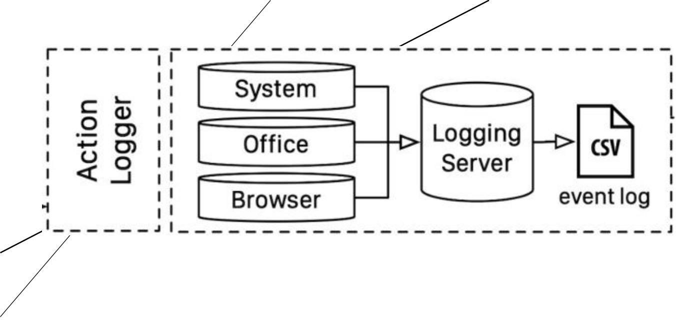
2. Supervision Feature: Activating this feature allows the user to capture a notation of necessity of the conducted event. If this feature is activated after each click a pop up opens asking the user the question show in figure TBD.

3. **Process Discovery**: The process discovery does start the Directly Follows Graph (DFG) discovery and does produce a PDF file containing the mined DFG into the folder "RPA/[filename]/process_discovery/".

4. **Analysis type**: The user can select the type of automation to be emulated into the bot. This happens after the number of runs were executed:

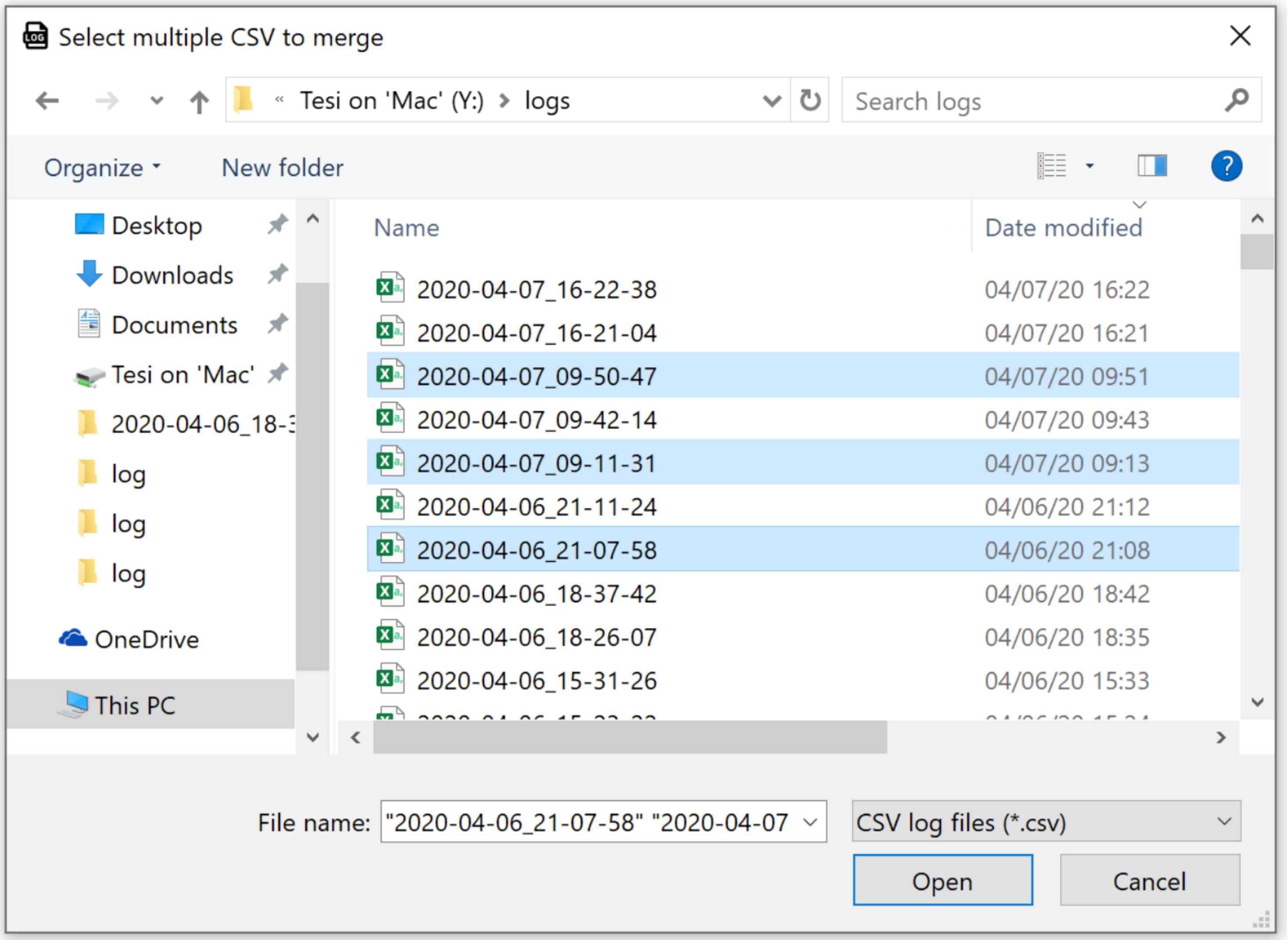
1. Most frequent Routine: Selects the process variant that has been executed the most times in the log.
2. Variation points: Lets the user select the variants after the number of runs were executed.

5. **Number of runs after which the event log is generated**: This setting controls the number of runs, which have to be recorded before the process discovery and process automation are started. An information about the number of runs is presented after each recording run



FILE DIALOG

The dialog is used to select event logs to merge or to run. It supports both single and multiple selection, and it is possible to specify the file types that can be selected, in this case only CSV log files. It returns a list of strings representing the path of each file selected.



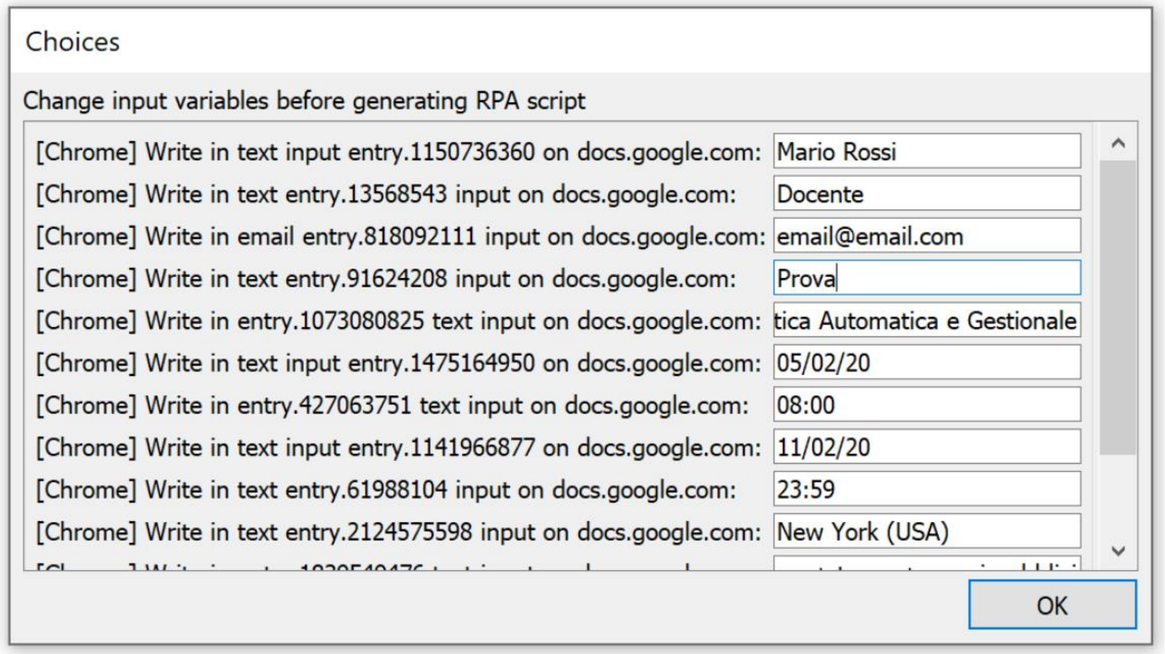
Select multiple CSV to merge

« Tesi on 'Mac' (Y:) › logs

Name	Date modified
2020-04-07_16-22-38	04/07/20 16:22
2020-04-07_16-21-04	04/07/20 16:21
2020-04-07_09-50-47	04/07/20 09:51
2020-04-07_09-42-14	04/07/20 09:43
2020-04-07_09-11-31	04/07/20 09:13
2020-04-06_21-11-24	04/06/20 21:12
2020-04-06_21-07-58	04/06/20 21:08
2020-04-06_18-37-42	04/06/20 18:42
2020-04-06_18-26-07	04/06/20 18:35
2020-04-06_15-31-26	04/06/20 15:33

File name: "2020-04-06_21-07-58" "2020-04-07

CSV log files (*.csv)



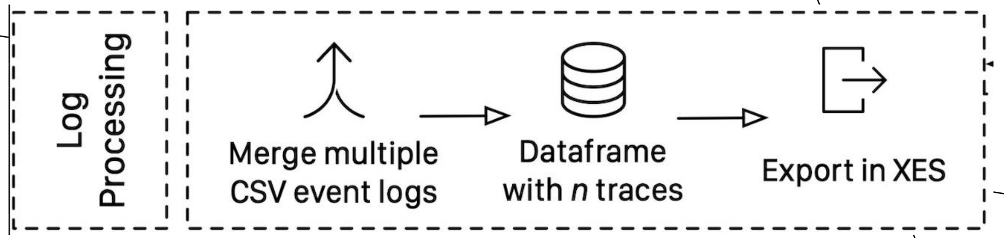
Choices

Change input variables before generating RPA script

[Chrome] Write in text input entry.1150736360 on docs.google.com:	Mario Rossi
[Chrome] Write in text entry.13568543 input on docs.google.com:	Docente
[Chrome] Write in email entry.818092111 input on docs.google.com:	email@email.com
[Chrome] Write in text entry.91624208 input on docs.google.com:	Prova
[Chrome] Write in entry.1073080825 text input on docs.google.com:	tica Automatica e Gestionale
[Chrome] Write in text input entry.1475164950 on docs.google.com:	05/02/20
[Chrome] Write in entry.427063751 text input on docs.google.com:	08:00
[Chrome] Write in text input entry.1141966877 on docs.google.com:	11/02/20
[Chrome] Write in text entry.61988104 input on docs.google.com:	23:59
[Chrome] Write in text entry.2124575598 input on docs.google.com:	New York (USA)

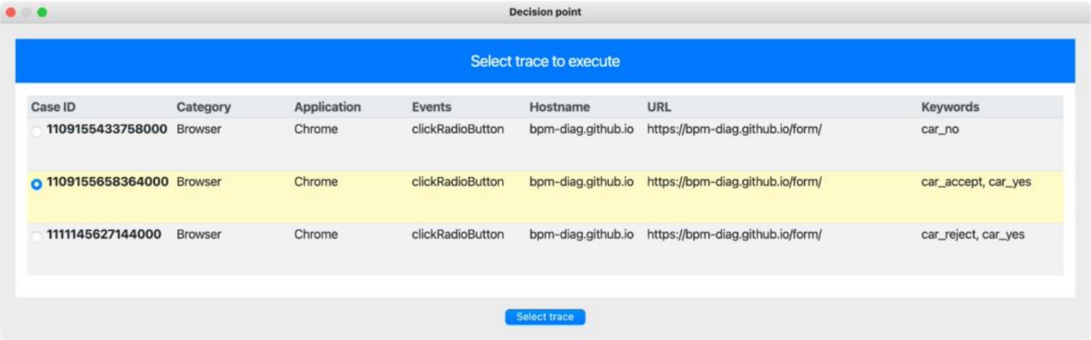
CHOICES DIALOG

Once the routine to automatize is selected, the user can customize its editable fields through a custom dialog window. The program automatically recognises which fields are editable, such as typing something in a web page, renaming a file or pasting a text, and dynamically builds the GUI to let the user edit them. After confirmation, the dataframe is updated.



DECISIONS DIALOG

The user can choose which user actions enact by means of a decision dialog, which displays data from the decided dataframe, that is used to group together all the rows of a routine trace belonging to a routine fragment into a single line.



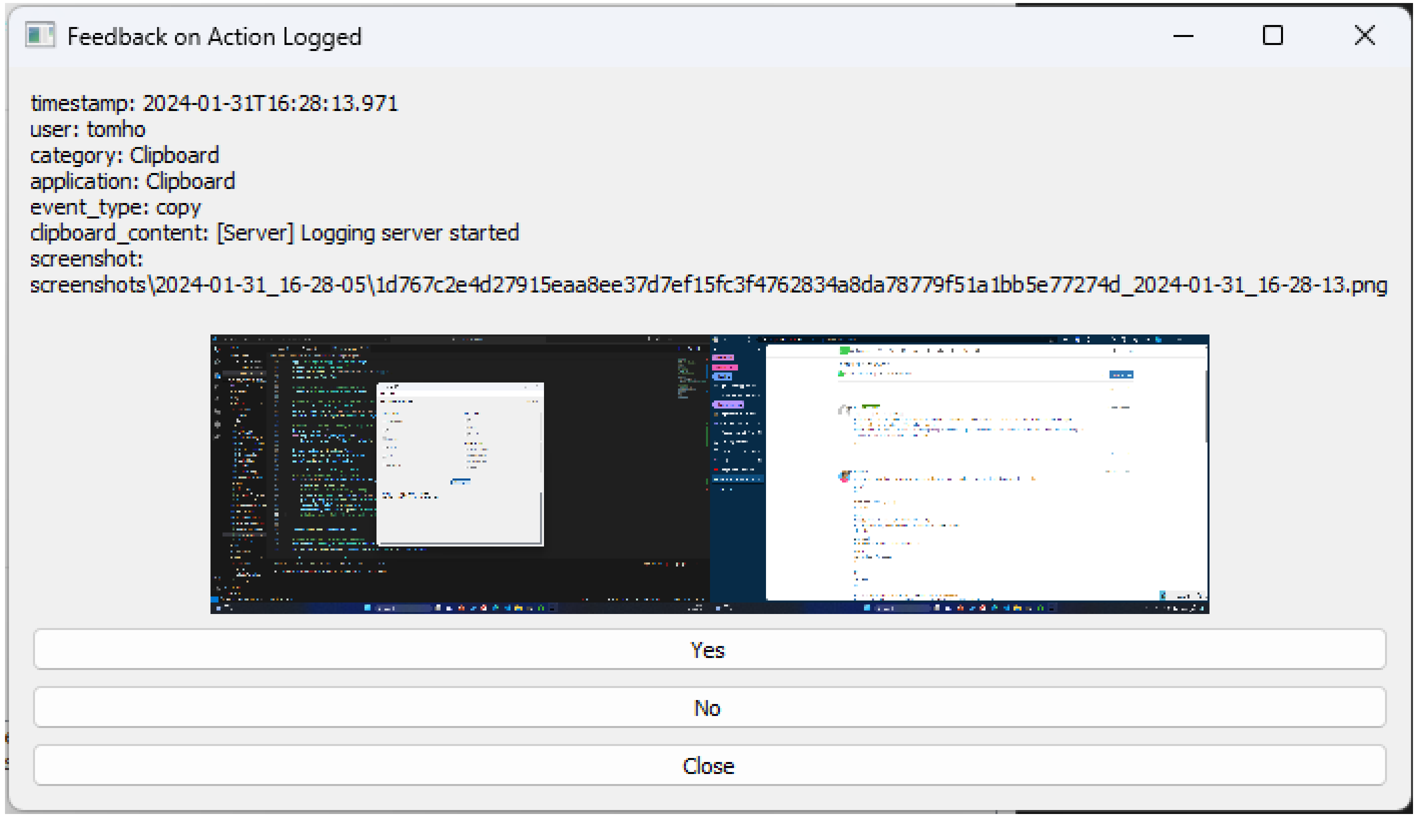
Decision point

Select trace to execute

Case ID	Category	Application	Events	Hostname	URL	Keywords
○ 1109155433758000	Browser	Chrome	clickRadioButton	bpm-diag.github.io	https://bpm-diag.github.io/form/	car_no
● 1109155658364000	Browser	Chrome	clickRadioButton	bpm-diag.github.io	https://bpm-diag.github.io/form/	car_accept, car_yes
○ 1111145627144000	Browser	Chrome	clickRadioButton	bpm-diag.github.io	https://bpm-diag.github.io/form/	car_reject, car_yes

SUPERVISION DIALOG

The user can annotate the actions he/she has done using the supervision feature enabled in the preferences window. The selection currently does annotate the event log, but no further action is taken on the data. It is useful for future supervised learning and noise detection.



```
timestamp: 2024-01-31T16:28:13.971
user: tomho
category: Clipboard
application: Clipboard
event_type: copy
clipboard_content: [Server] Logging server started
screenshot:
screenshots\2024-01-31_16-28-05\1d767c2e4d27915eaa8ee37d7ef15fc3f4762834a8da78779f51a1bb5e77274d_2024-01-31_16-28-13.png
```

RUNNING A BOT IN PYTHON

1. CMD Terminal: Navigate to the folder with the .py bot file using *cd.*

2. Run Bot with python “filename.py”: The bot starts executing the emulated routine

RUNNING A BOT IN UIPATH

1. **Start UiPath Studio**
2. **Open a local project:** In the RPA folder select the SW_Robot/UiPath/ subdirectory and import the project.json
    1. UiPath will download all necessary dependencies and create a new Workspace