

# Work Tagger: A Labelling Companion

Manuel Resinas<sup>1,\*</sup>, Rocío Goñi-Medina<sup>1</sup>, Iris Beerepoot<sup>2,\*</sup>, Adela del-Río-Ortega<sup>1</sup> and Hajo A. Reijers<sup>2</sup>

<sup>1</sup>Universidad de Sevilla, Avda. Reina Mercedes s/n, 41012 Seville, Spain

<sup>2</sup>Utrecht University, Heidelberglaan 8, 3584 CS Utrecht, The Netherlands

## Abstract

In settings where data is recorded at a fine-granular level, it needs to be abstracted to enable process mining. While several event abstraction techniques exist, the majority are supervised and require manually labelled datasets, a process that is both time-consuming and critical for developing new methods. To streamline this process, we introduce a tool designed to facilitate the tagging of fine-granular data using predefined activities, with a specific focus on Active Window Tracking (AWT) data. The tool offers features such as data visualization, filtering, and automatic classification based on GPT, which can be adjusted by the user. Our evaluation, involving four researchers tagging their AWT data, demonstrates that increased experience with the tool leads to faster tagging, and we discuss potential future enhancements.

## Keywords

process mining, event abstraction, active window tracking, task classification

Metadata description	Value
Tool name	Work Tagger
Legal code license	Apache 2.0
Languages, tools and services used	Python, Streamlit, Open AI GPT API
Supported operating environment	Microsoft Windows, GNU/Linux, Mac
Download/Demo URL	<a href="https://worktagger.streamlit.app/">https://worktagger.streamlit.app/</a>
Source code repository	<a href="https://github.com/project-pivot/worktagger">https://github.com/project-pivot/worktagger</a>
Screencast video	<a href="https://www.youtube.com/watch?v=ulVh63TyR6k">https://www.youtube.com/watch?v=ulVh63TyR6k</a>

## 1. Introduction

One of the core requirements for process mining is the recording of process activities [1]. However, process behavior is not always recorded at the right granularity level [2]. In settings where data is recorded at a very fine-granular level, groups of events may need to be abstracted to a higher-level activity [3]. Several event abstraction techniques have been proposed, the majority of which are supervised techniques [4]. Supervised techniques require additional

ICPM 2024 Tool Demonstration Track, October 14-18, 2024, Kongens Lyngby, Denmark

\*Corresponding author.

✉ resinas@us.es (M. Resinas); rgoni@us.es (R. Goñi-Medina); i.m.beerepoot@uu.nl (I. Beerepoot); adeladelrio@us.es (A. del-Río-Ortega); h.a.reijers@uu.nl (H. A. Reijers)

🆔 0000-0003-1575-406X (M. Resinas); 0000-0002-6301-9329 (I. Beerepoot); 0000-0003-3089-4431 (A. del-Río-Ortega); 0000-0001-9634-5852 (H. A. Reijers)



© 2024 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

information, typically a subset of the data that is manually labelled by a domain expert. Although this is a laborious process, the importance of labelled datasets for the development of new techniques cannot be overstated. As such, it is vital that this labelling needs to be made as quick and easy as possible.

In this paper, we present a tool that aims to support the tagging of fine-granular data using a set of predefined activities. The basic features of the tool are applicable to different types of datasets, but this particular implementation focuses on the labelling of so-called Active Window Tracking (AWT) data. The opportunities of this data for mining work practices are described in [5]. AWT data contains information about a person’s computer behavior in the form of start and end times of each active application and window. It allows for the use of different views on the data, the application of filters, and the modification of visualizations. Additionally, it allows a user to start from an automatic classification based on GPT and modify the labels. Through an evaluation by four researchers tagging their own AWT data, we demonstrate how more experience with the tool results in faster tagging, and reflect on future improvements.

## 2. Tool Description

Work Tagger is a web-based tool that facilitates the classification of AWT data. Work Tagger has been developed using Streamlit, which is an open-source Python framework to create interactive web-applications, whose main characteristic is that it integrates the development of both a web-based frontend and backend into a single Python code base.

Work Tagger is designed to use the AWT data collected by an application called Tockler<sup>1</sup>, which records all active windows on the computer while the application is installed and running. We have chosen Tockler because it is open source and runs locally, which helps to avoid privacy concerns. However, Work Tagger is designed to easily integrate data coming from other similar tools. The web-based user interface of Work Tagger allows users to upload files, select data for classification, and visualize the results using different views. The user interface is designed to be user-friendly and interactive, featuring dynamic UI elements such as buttons, select boxes and sliders for ease of use. Streamlit’s interactive widgets enhance user experience by providing responsive and intuitive controls.

Once users upload their AWT logs (in csv format) through the user interface, the backend processes these files, converting them into a dataframe. During this process, columns are prepared with the necessary formats for efficient data manipulation. In contrast to other web-based tools, Work Tagger does not use a database for data storage. Instead, Work Tagger relies on session state variables to store data temporarily. This approach ensures that each user’s data is isolated and managed independently, preventing conflicts in a multi-user environment. These session state variables are maintained for the duration of the user’s interaction with the application, ensuring a personalized and consistent experience. Additionally, this decision is related to privacy concerns, we do not store records of individuals’ computer usage, thus protecting users’ personal information and ensuring their privacy.

When the AWT log is loaded in the application, Work Tagger displays the AWT events in a table and allows the user to label the events with the activity and case the user was performing

---

<sup>1</sup><https://maygo.github.io/tockler/>

at that moment. For activities, the user may opt to undertake the classification process either manually or automatically. In the former case, the user has to choose the activity from a predefined list of activities and subactivities based on the one used in [5] for academic work activities. However, Work Tagger is designed so that the set of activities can be easily modified<sup>2</sup>. In the latter case, once automatic classification is initiated, the backend logic sends the data to the classification core to interact with the OpenAI API to perform zero-shot classification using the GPT-4o model based on the same set of activities and subactivities used in the manual classification. We opted for this approach to provide a highly flexible and adaptive classification system that does not require training the model beforehand.

Concerning the labeling of cases, only manual labeling is possible. Moreover, unlike activities, the set of cases are open and users can pick from case labels already used in the dataset or can enter new case labels. The reason for following a different approach for cases is because, unlike activities, they are very specific to a particular person and a particular moment in time.

### 3. Tool Functionality

In this section, we describe the different functionalities Work Tagger has:

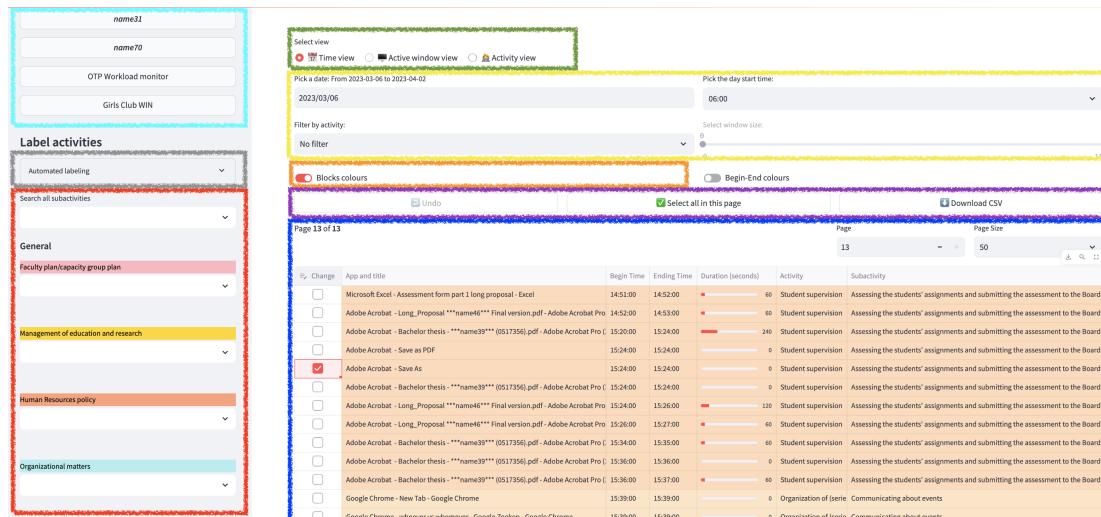
**AWT Event Log Upload.** To start using Work Tagger, the user must upload an event log from Tockler. The accepted file type is CSV with a size limit of 200 MB, although it can be easily extended. By default, the labels of *Activity* and *Subactivity* will be “No work-related” and “Unspecified No work-related.” The user can also upload a CSV file that has been previously labeled in the application, or load a publicly available sample dataset [6].

**AWT Data Visualization.** Once the AWT event log is uploaded, the data is displayed by the application in a table (see blue box in Fig. 1). Work Tagger uses *pagination* in the table to display the classify data in manageable chunks allowing users to navigate through the data by selecting the page they want to view and also to modify the size of the page they are visualizing. Users can personalize the visualization of the data in the table by choosing between three different views (green box in Fig. 1):

- **Time view.** In this view, each row of the table is an event in the AWT. The rows are ordered by timestamp. Several aspects can be configured in this view using the controls in the yellow box of Fig. 1. Using a calendar, users can select the date for which the data should be displayed. The calendar allows selection from the earliest to the most recent entry in the uploaded event log. Users can also select the start time of their day to adapt to people that have different schedules, e.g., night owl workers. It is also possible to filter events by activity possibly showing a window of events before or after them of configurable size. Finally, when *Begin-End colouring* is enabled, if the time difference between the End Time of one row and the Begin Time of the following row exceeds the number of minutes selected in the slider, the row will be marked in gray.

---

<sup>2</sup>More information on how to do it can be found at <https://github.com/project-pivot/worktagger>, in the README file. Still, we plan to extend the application’s flexibility by allowing users to upload their own custom list of activities, enabling customization and adaptation to various domains.



**Figure 1:** User Interface of Work Tagger with an uploaded file in the *Time view*

- **Active window view.** In this view, the rows of the table are the different active window titles that appear in the log. The view is sorted by the number of times the title appears in the log, although it can also be sorted by duration. In this view, users can filter by application, so that only the active window titles of a certain application are shown, and by window title, so that only the titles that contain the words entered by the user are shown. This view is useful to label some activities that are clearly related to a certain window title. For instance, if the window title contains *Overleaf*, then it is likely that the activity is related to *Write research papers*.
- **Activity view.** This view is similar to the *Time view*, but events are grouped by *subactivity*. Like the time view, this view is sorted by timestamp, but it can also be sorted by duration of the subactivity and number of events included in a subactivity. Furthermore, users can also filter by activity. This view is useful to provide an overview of the activity labels applied and to facilitate case labeling.

Finally, users can also enable the *Blocks colours* option depicted in the orange box of Fig. 1. When this option is enabled, each row will be colored differently based on its *Activity* value.

**Manual Classification.** It is performed by selecting one or more rows in the table using the checkboxes and then applying the labels that appear in the left sidebar of the application. Users can modify the subactivity value using the sidebar (red box in Fig. 1), in three different ways:

1. Selecting from a comprehensive list of all activities in the first select box.
2. Clicking on one of the buttons that display the last three used subactivities.
3. Using a select box that categorizes subactivities by activity.

To label cases, users can choose from the cases that have already been used in the dataset by clicking in the corresponding button (cyan box in Fig. 1), or can add a new case label using the corresponding textbox.

**Automatic Classification.** It is performed using an expandable box that appears in the sidebar (grey box in Fig. 1). Once expanded, a form will appear, allowing the user to enter their OpenAI key and organization details, and select the data the user wishes to classify: all data, only selected rows, or only data from a selected date. Once the necessary information is provided, the user clicks a button to start the automatic classification process.

**Undo and Download CSV Buttons** By clicking these buttons (see purple box in Fig. 1), the user can undo the last change made, and download the updated data with all modifications made, respectively. by clicking the “Download CSV” button.

## 4. Tool Maturity

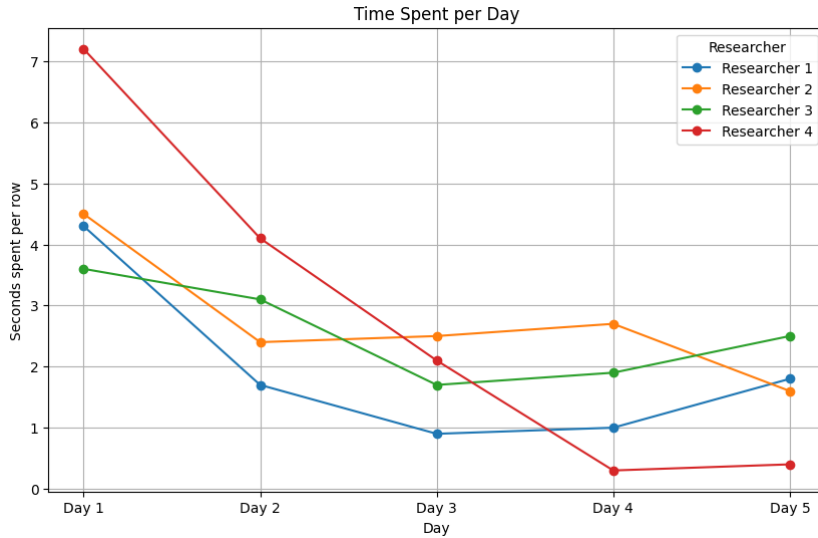
In order to evaluate and improve the tool, four authors of this paper collected data using Tockler and used Work Tagger to label a week’s worth of data. While doing so, they recorded the time they spent on labelling each day of that week and the number of rows labelled. The results are depicted in Table 1. Generally speaking, the time it takes to tag a row in the dataset strongly decreases the more time the user spends in the tool, as can be seen in Fig. 2. This is especially striking for researcher 4, who went from taking 7.2 seconds per row to 0.3 and 0.4 seconds per row. For researcher 1, a decrease between day 1 and 5 can also be seen, but it is less linear. This may be due to the fact that researcher 1 tagged days 1 through 4 within 8 days of each other, while day 5 was tagged 11 days later, when the researcher had to get back into the swing.

After tagging days 1 and 2 simultaneously, the researchers discussed their experiences and proposed changes to the tool. This resulted in the addition of a list of the subactivities last used and a pagination button on the bottom of the page. When researchers 1 through 3 had finished tagging all days, there was another round of changes to the tool. We added the option of uploading sample data, active window and activity views, a visualization depicting the duration, and an ‘undo’ button. Researcher 4 completed the final three days using the current version of the tool, resulting in the fastest tagging times observed.

**Table 1**

Overview of time spent, rows tagged, and seconds spent per event tagged.

Researcher	Metric	Day 1	Day 2	Day 3	Day 4	Day 5
Researcher 1	Time spent (in minutes)	70	35	24	25	7
	# rows tagged	968	1250	1694	1472	238
	Seconds spent per row tagged	4.3	1.7	0.9	1	1.8
Researcher 2	Time spent (in minutes)	60	22	22	23	12
	# rows tagged	804	561	536	514	442
	Seconds spent per row tagged	4.5	2.4	2.5	2.7	1.6
Researcher 3	Time spent (in minutes)	37	17	17	15	3
	# rows tagged	616	329	614	476	72
	Seconds spent per row tagged	3.6	3.1	1.7	1.9	2.5
Researcher 4	Time spent (in minutes)	52	59	33	3	0.5
	# rows tagged	433	862	924	550	69
	Seconds spent per row tagged	7.2	4.1	2.1	0.3	0.4



**Figure 2:** Overview of Time Spent per Researcher.

## Acknowledgments

This work has been partially supported by grants PID2021-126227NB-C21 and PID2022-140221NB-I00 funded by MCIN/AEI/10.13039/501100011033/FEDER, EU, and TED2021-131023B-C22 funded by MCIN/AEI/10.13039/501100011033 and by the European Union “NextGenerationEU”/PRTR.

## References

- [1] W. Van Der Aalst, Process mining: Overview and opportunities, *ACM Transactions on Management Information Systems (TMIS)* 3 (2012) 1–17.
- [2] R. J. C. Bose, R. S. Mans, W. M. Van Der Aalst, Wanna improve process mining results?, in: 2013 IEEE symposium on computational intelligence and data mining (CIDM), IEEE, 2013, pp. 127–134.
- [3] F. Mannhardt, N. Tax, Unsupervised event abstraction using pattern abstraction and local process models, *arXiv preprint arXiv:1704.03520* (2017).
- [4] S. J. van Zelst, F. Mannhardt, M. de Leoni, A. Koschmider, Event abstraction in process mining: literature review and taxonomy, *Granular Computing* 6 (2021) 719–736.
- [5] I. Beerepoot, D. Barenholz, S. Beekhuis, J. Gulden, S. Lee, X. Lu, S. Overbeek, I. Van De Weerd, J. M. Van Der Werf, H. A. Reijers, A window of opportunity: Active window tracking for mining work practices, in: 2023 5th International Conference on Process Mining (ICPM), IEEE, 2023, pp. 57–64.
- [6] I. Beerepoot, A month’s worth of labelled active window tracking data, in: Proceedings of the Best Dissertation Award, Doctoral Consortium, and Demonstration and Resources Forum at BPM 2024, 2024.