Facultés Universitaires Notre Dame De la Paix (FUNDP Namur)
Faculté d'Informatique

École Polytechnique de Montréal
Département de Génie Logiciel

Team Leader: Yann-Gaël Guéhéneuc

TAUPE v2.0 - User's Guide

Benoît De Smet
Lorent Lempereur

February 12, 2011

Contents

List of Figures

1 Introduction

TAUPE[1] (Thoroughly Analyzing the Understanding of Programs through Eyesight [1]) is a software designed by the *Ptidej team*[2] to import data from eye-trackers (as described in Section 6) and to allow the execution of various algorithms (descibed in Section 3) on the collected data.

TAUPE was originally designed to compare the way of looking, analysing, and reading of subjects of UML diagrams. A subject was asked different questions about her understanding of the diagrams and had to complete a set of maintenance tasks on it. Her eye's movements were recorded using an eye tracking device. These eye mouvements must be analysed by researchers to assess the comprehension process of the subject. Some softwares in the field of data analysis already existed but none was open source so it was really difficult to run customized algorithms on the collected data.

This document contains the user's documentation and all information needed to use TAUPE. All the information concerning the development and the implementation of extensions for the software can be found in the *Developer's Guide*[3].

This guide concerns the version `2.0` of TAUPE, last version currently available (February 12, 2011).

A complete description of the software including this guide and the developer's guide can be found on the ptidej website[4].

[1] "Taupe" means "mole" in French and is pronounced 'tOp.
[2] `http://www.ptidej.net/team`
[3] Available via `http://www.ptidej.net/research/taupe/`
[4] `http://www.ptidej.net/research/taupe/`

2 Vocabulary

The following are terms used regulary all trough this guide with specific meanings:

Gaze: The gaze is typicaly "where we are looking" this is the data recorded by the eye-tracker.

Fixation: A fixation is a position of the eye during a gaze.

Saccade: A saccade is a movement of the eye between two fixations.

Experiment: An experiment represents the whole system (It is a set of questions, their answers, the fixations, the saccades, the area of interest . . .) that we want to analyse with TAUPE. When the user is using the software, this one handles up to only one experiment at a time.

Question: A question is related to one image (usually an UML class diagram) associated with a task that the subject has to do on the diagram.

Subject: A subject is a person who answered a set of questions during the duration of an experiment. It is defined by a set of characteristics like its name or its level of study. Each subject is linked to a file that is generally generated by an eye-tracking device and that contains the whole set of data about the subject's answers.

Group: A group is a set of subjects. A group is defined by its type and characteristics of this one. For example groups that classify the subjects based on their UML knowladge, gender, desgin patterns knowladge. . .

Area of Interest: An area of interest is an area on a question's image. This area can be relevant or not to a task (relevance is specified in the corresponding `.aoi`). It also can be "ignorable", it means that the system will not take account of this area. Areas are defined in a text file (`AOI files`) for each question. For example, if a question is related to an image called `foo.png`, then the file that defines the areas is named `foo.aoi`. The file structure is defined in Section 5.

Answer: An answer is what a specified subject has answered to a specified question. This answer can be correct or not (determined by the researchers) and is related to a set of fixations and a set of saccades. All the answers for a subject are in the same `.subject` file (see Section 5).

Q**:** The set of questions.

A**:** A set of answers.

F**:** A set of fixations.

S**:** A set of saccades.

AOI**:** A set of area of interest.

F_i**:** Set of fixations in the answer i ($i \in A$) related to the subject's answer i without considering the fixations that are in a "ignorable" AOI.

S_i: Set of saccades in the answer i ($i \in A$).

A_i: The set of answers related to the question i ($i \in Q$).

$AORI_i$: Set of areas of relevant interest related to the question i ($i \in Q$).

$AOII_i$: Set of areas of irrelevant interest related to the question i ($i \in Q$).

AOI_i: Set of areas of interest related to the question i ($i \in Q$). This set can be calculated with

$$AOI_i = AORI_i \cup AOII_i$$

3 Algorithms

With the current version of TAUPE (2.0), different algorithms are available. The results generated by those algorihms are often sorted by group (ex: Beginners, experts, male, female...) of subjects and some statistics (average, standard deviation...) are sometimes realised.

3.1 Fixations Statistics

As explained in [2], "A fixation algorithm must produce fixations that meet certain minimum characteristics. The center of a typical fixation is within 2-3^o from the observed target object [3] and the minimum processing duration during a fixation is 100-150 ms [4]."
This algorithm provides a set of algorithms related to the fixations:

$t : F \rightarrow \mathbb{N}$: The specified fixation's duration (in milliseconds).

$fix : AOI \times A \rightarrow F^n$: The fixations in a specified area of interest for a specified answer.

$surface : AOI \rightarrow \mathbb{N}$: The specified AOI's surface (in pixels^2)

$FAORI_i$: Set of fixations in a Relevant AOI for the answer i ($i \in A_j$) as:

$$FAORI_i = \bigcup_{k \in AORI_j} fix(k, i)$$

$FAOII_i$: Set of fixations in an Irrelevant AOI for the answer i ($i \in A$) as:

$$FAOII_i = \bigcup_{k \in AOII_j} fix(k, i)$$

$TAOI_i$: Total duration (in milliseconds) of the fixations in a AOI for the answer i ($i \in A_j$).

$$TAORI_i = TAORI + TAOII$$

$TAORI_i$: Total duration (in milliseconds) of the fixations in a Relevant AOI for the answer i ($i \in A_j$).

$$TAORI_i = \sum_{f \in FAORI_i} t(f)$$

$TAOII_i$: Total duration (in milliseconds) of the fixations in an Irrelevant AOI for the answer i ($i \in A_j$).

$$TAOII_i = \sum_{f \in FAOII_i} t(f)$$

3.1.1 Total Number of Fixations

As mentionned in [2], "The number of fixations overall is thought to be negatively correlated with search efficiency". The number of fixations for a subject's answer i ($i \in A$) is here $\#F_i$.

3.1.2 Number of Fixations per Type of Areas of Interest

There exists currently three kinds of areas of interest: *Relevant*, *Irrelevant*, and *Ignorable*. This latter will not be considered, all the fixations in a "ignorable" aoi will be ignored in all algoritms concerning the aoi. "More fixations on a particular area indicate that it is more noticeable, or more important, to the viewer than other areas." [5]
Therefore, TAUPE makes the distinction between the relevant areas ($\#FAORI_i$) (according to the specified question) and the areas that the subject should not look (irrelevant) ($\#FAOII_i$).

3.1.3 Fixation Duration

According to [6], the duration of the fixations about a specified area can have two different meanings:

1. A longer fixation duration means that the subject has some difficulty to extract information [7].
2. A longer fixation duration means that the object is "more engaging in some way"[8].

3.1.4 Normalized Fixations per Area of Interest

This metric $NORM\_RATE$ represents the ratio between the normalized number of fixations in an AORI and the normalized number of fixations in an AOII[9]. It is used to assess the subject's effort. The $NORM\_RATE_i$ for a subject's answer i ($i \in A_j$) is described as:

$$NORM\_RATE_i = \frac{\frac{\#FAORI_i}{\#AORI_j}}{\frac{\#FAOII_i}{\#AOII_j}}$$

3.1.5 Overall Fixation Rate overall

Not implemented yet. "This metric is closely related to fixation duration. Since the time between fixations (typically short duration saccadic eye movements) is relatively small compared with the time spent fixating, fixation rate should be approximately the inverse of fixation duration." [10]

3.1.6 IN AORI / IN AOII

Not implemented yet. This is the ratio between the number of fixations in the areas of relevant interest and in the areas of irrelevant interest. This metric excludes the fixations that are out of an area of interest. It is used to assess the effort's relevance. This ratio IN_OUT_i for a subject's answer i is defined as:

$$IN\_OUT_i = \frac{\#FAORI_i}{\#FAOII_i}$$

3.1.7 On-target/All-target

Not implemented yet. This metric "can be defined by counting the number of fixations falling within a designated area of interest, then dividing by the all fixations. This is a content-dependent efficiency measure of search, with smaller ratio indicating lower efficiency"[2]. This metric ON_ALL_i for an area of interest i about a subject's answer j is calculable as:

$$ON\_ALL_i = \frac{\#fix(i,j)}{\#F_j}$$

3.1.8 Post-target Fixations

Not implemented yet. "The number of *post-target fixations*, or fixations on other areas, following target capture, can indicate the target's meaningfulness to a user. High values of non-target checking, following initial target capture indicate target representations with poor meaningfulness or visibility"[2]. TAUPE computes this metric $POST\_TARGET_{ij}$ for an area i about a subject's answer j with the following formula:

$$POST\_TARGET_{ij} = \#FAORI_j + \#FAOII_j - \#fix(i,j)$$

3.1.9 Spatial Density

"[c]overage of an interface due to search and processing may be captured by the spatial distribution of [fixations]. [...] The [image] can be divided into grid areas either representing specific objects of physical screen area. [...] The *spatial density* index [is] equal to the number of cells containing at least one [fixation], divided by the toal number of grid cells. [...] A smaller spatial density indicated more directed search, regardless of the temporal [fixation] sampling order."[2] The cell's size used in TAUPE is 64x64px (this value can be changed see the Developper's guide).

3.2 Saccades Statistics

This set of metrics can be used to compute statistics about saccades.

3.2.1 Transition Density

The transition density is based on the notion of *transition matrix* and *link analysis*. "[F]requent transitions from one region of a display to another indicates inefficient scanning with extensive search. [...] The transition matrix is a tabular representation of the number of transitions to and from each defined area."[2]. A cell (part of the grid areas either representing specific objects of physical screen area) is filled if a saccade starts of ends in its area. With the *transition matrix*, it is possible to compute its density.

$$TRANSITION\_DENSITY_i = \frac{\sum_{x \in C} isFilled(x)}{\#C}$$

where:

C_i is the set of cells in the transition matrix of the subject's answer i ($i \in Q$).

$isFilled : C \rightarrow \{0,1\}$**:** is a function returning 1 if the specified cell is filled and 0 otherwise.

3.3 Convex Hull Computing

This algorithm can be used to compute the convex hull[5] of all the fixations for each group defined in TAUPE. FIGURE 1[6] displays the convex hull as a imagining band stretched open to encompass whole set of fixations. TAUPE uses an algoritm described in [11] to compute the convex hull of the whole set of fixations.

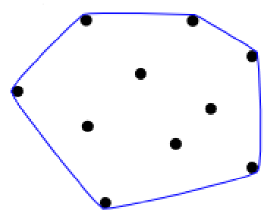


Figure 1: Convex Hull (in blue)

This algorithm generates the convex hull for each subject's answer. Moreover, the software builds the convex hulls for a series of percentages. These percentages corresponds to the percentage of keeped fixations (eg: 50% means that only half of the fixations are used, the longest ones). Currently, this series is defined on the range [5%. . . 100%] with an interval of 5%.

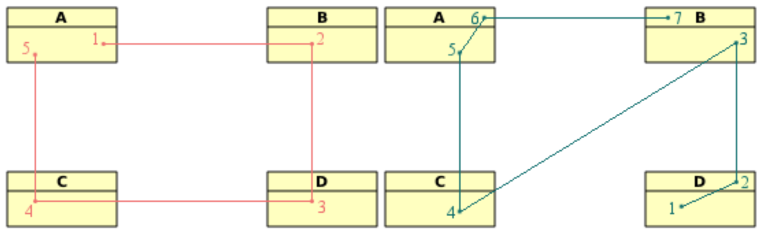
3.4 Visual Path Computing

This algorithm can be used to compute the differences between the *visual path* of two subjects' answers.
A *visual path* is a serie of visited areas of interest sorted chronologically. An area is visited if there is a fixation in it.
For example, if a question contains four areas of interest `A`, `B`, `C` and `D`, then two possible *visual paths* would be `ADCABC` or `DABC`. In the case of FIGURE 2, each area is related to a class in a simple UML class diagram), then a possible *visual path* would be `ABDC` (FIGURE 2a) or `DBCAB` (FIGURE 2b). However, a same area of interest that is visited twice consecutively (two consecutives fixations in the same area of interets) is considered only once. So, while the FIGURE 2b's visual path could be `DDBCAAB`, it is actually `DBCAB`.

The difference between two visual paths is computed using an *edit distance* algorithm: the *Levenshtein algorithm*. The Levenshtein algorithm compares two strings. TAUPE uses its own adaption of the algorithms to handle a list of area of interest.
TAUPE compares two subjects' visual paths according to the percentage of fixations taken [5%. . . 100%] with a granularity of 5%. For example, if the percentage of fixations considered is 75%, then the choice of fixations will be based on the fixations' duration (the 75% longest fixations will be taken).

[5] The convex hull or convex envelope for a set of points X in a real vector space V is the minimal convex set containing X.

[6] Source: http://commons.wikimedia.org/wiki/File:ConvexHull.png



(a) ABDC (b) DBCAB

Figure 2: The visual path

3.5 Correctness Computing

This computation method can be used to generate a summary (for each group) about the correctness of the answers provided by the subjects via the `.subject` files (5.1).

3.6 Time Computing

This algorithm can be used to compute statistics about the time spent on each question for each group. Thus, this module does not compute statistics about the fixations' duration, the time spent in a specified area of interest...

3.7 Discussion

- Two scan paths can have the same *convex hull* (section 3.3) and the same *spatial density* (section 3.1.9) but they can have a different *transition density*. A better scan path would have a lesser dense *transition matrix*.
- Typically, longer is an answer, lesser the subject's efficiency. However, [12] thinks that in the case of the reading, this metric is not exact because of the inconstant speed reading of the subjects.

4 How Tos

4.1 How to Launch Taupe

TAUPE is provided in a `.jar` file. To run TAUPE, just open a command line interface and type:

```
java -jar TAUPE.jar
```

or launch the JAR file from your file browser if your system allows it. It is possible that you need to increase the amount of memory allowed to Java if you have to load a lot of data. You do this by adding the `-Xms`A`m -Xmx`B`m` parameter, where A is the initial amount of memory (in MB) and B the maximal amount. For example:

```
java -Xms32m -Xmx1024m -jar TAUPE.jar
```

Please note that de `.jar` file should not be renamed, the file name must be *TAUPE.jar*.

4.2 How to Change the Software's Preferences

To set up the preferences, simply click on `File` → `Preferences` in the menu or press `Ctrl + P` . The configuration dialog box (FIGURE 3) will pop up.

Default directory for the AOI output generation: This directory is used by the AOI maker (FIGURE 7 and Section 4.5) to store the generated `.aoi` files (see 5.2).

Result directory: This directory is used to store the files generated by TAUPE which contains all the results. This is the output directory.

Data directory for experiments: This directory is used to load the files from the eye-tracker. The directory must contain the data from the eye-tracker (ex: files `.out` for *GazeTracker*) **AND** their corresponding `.subject` files as described in the section 5.1. The files must have the same names; for example: `Subject01.out` and `Subject01.subject`, `Subject02.out` and `Subject02.subject`, and so on. This is a input directory.

Data directory for diagrams: This is the directory that contains the images to load (ex: the diagrams images). Each image listed in the files from the eye-tracker must be in this directory. Also each image file (`.jpg` or `.png` file) must be accompagned by a `.aoi` file as described in Section 5.2. The image file and the AOI file must have the same name; for example: `JFreeChart_MD_Q1.png` and `JFreeChart_MD_Q1.aoi`.

4.3 How to Load Data

Before loading data into TAUPE the preferences must be set up as described in section 4.2.

1. To load data from an eye-tracker, in the main window, shown in FIGURE 5, click on `File` → `Load` in the menu (or press `Ctrl + L`). Select the parser according to the type of data you want to load in the dialog box FIGURE 4.

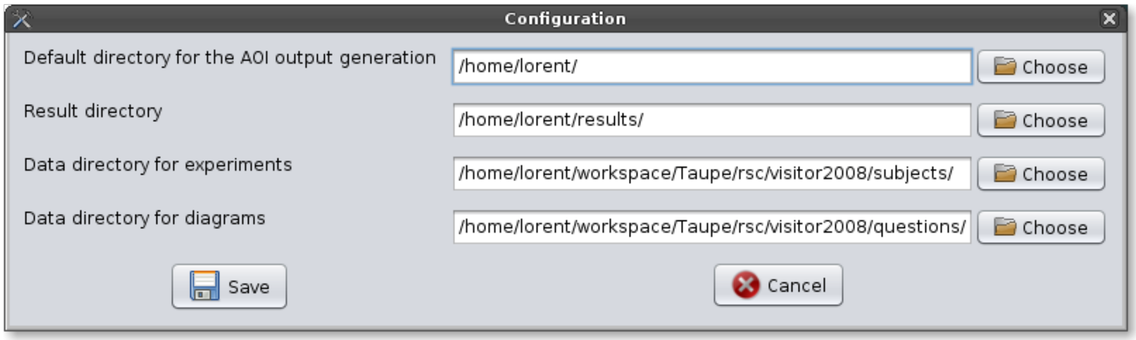


Figure 3: The preferences window

2. Choose the directory containing the data from the eye-tracker as described in Section 4.2 (*Data directory for experiments*). For the *GazeTracker* parser a file named `relevant.slides` is also needed in the directory that lists the images that should be loaded in TAUPE. All images not listed in this file will be ignored and will not be loaded. `relevant.slides` can be used if you need to ignore some data in some slides (e.g., if there is a white screen between each diagram). The structure of the `relevant.slides` file is explained in Section 5.3.

3. The *duration* (in milliseconds) parameter is used to specify the minimal amount of time that a fixation should have to be considered as such.

4. Press the `OK` button, the parsing starts. The loading could take some times depending on the amount of data contained in the eye-tracker's files and the number of subjects.

Figure 4: The parser chooser dialog box

4.4 How to Display Graphical Data

To display some data (saccades, convex hull, fixations. . .) on the panel, must load data as described in Section 4.3. Once the data is loaded in TAUPE, it is possible to view the loaded data.

1. Simply launch the visualisation tool shown on FIGURE 6 by clicking on the `Visualisation Tool` button an then on the `Execute` button on the right panel. The experimentaiton GUI is divided in two main parts. The panel on the left is used to display the image and data and the right panel is used to choose which information to be shown.

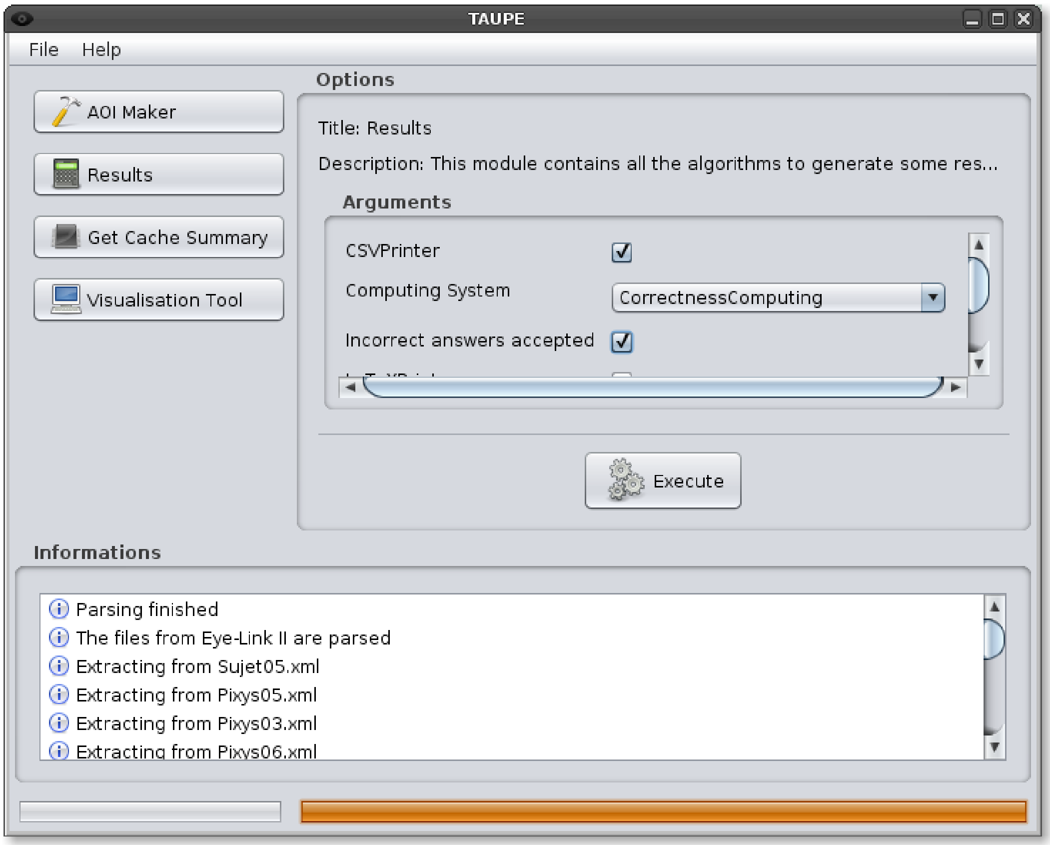


Figure 5: The TAUPE's main window

2. To display data simply tick the corresponding checkbox and choose its color. By pressing the `Draw` button the selected information will appear on the left panel. The image and the corresponding subject can be selected via the drop boxes in the top of the right panel.

3. The slider at the bottom of the right panel can be used to set the percentage of displayed fixations, for example to keep the longest fixations.

4. Clicking on a fixation will show its characteristics (duration, coordonates. . .) in the text area above the `Draw` button. Under this button, a progress bar can be found which show you the progression of the current drawing.

4.5 How to Create a Set of Areas of Interest

An area of interest is a polygon on an image taht TAUPE can use to distinguish fixations. The AOI maker shown on FIGURE 7 can be used to easily create a set of areas of interest on an image. To start the AOI maker, click on the AOI maker button and then on the `Execute` button. **This module saves you the effort to write an AOI file manually.**

1. To create a set of areas of interest it is needed to have an image loaded. To do so, click on `File` → `Load` or press `Ctrl + O`. In the bottom section the image file to be loaded

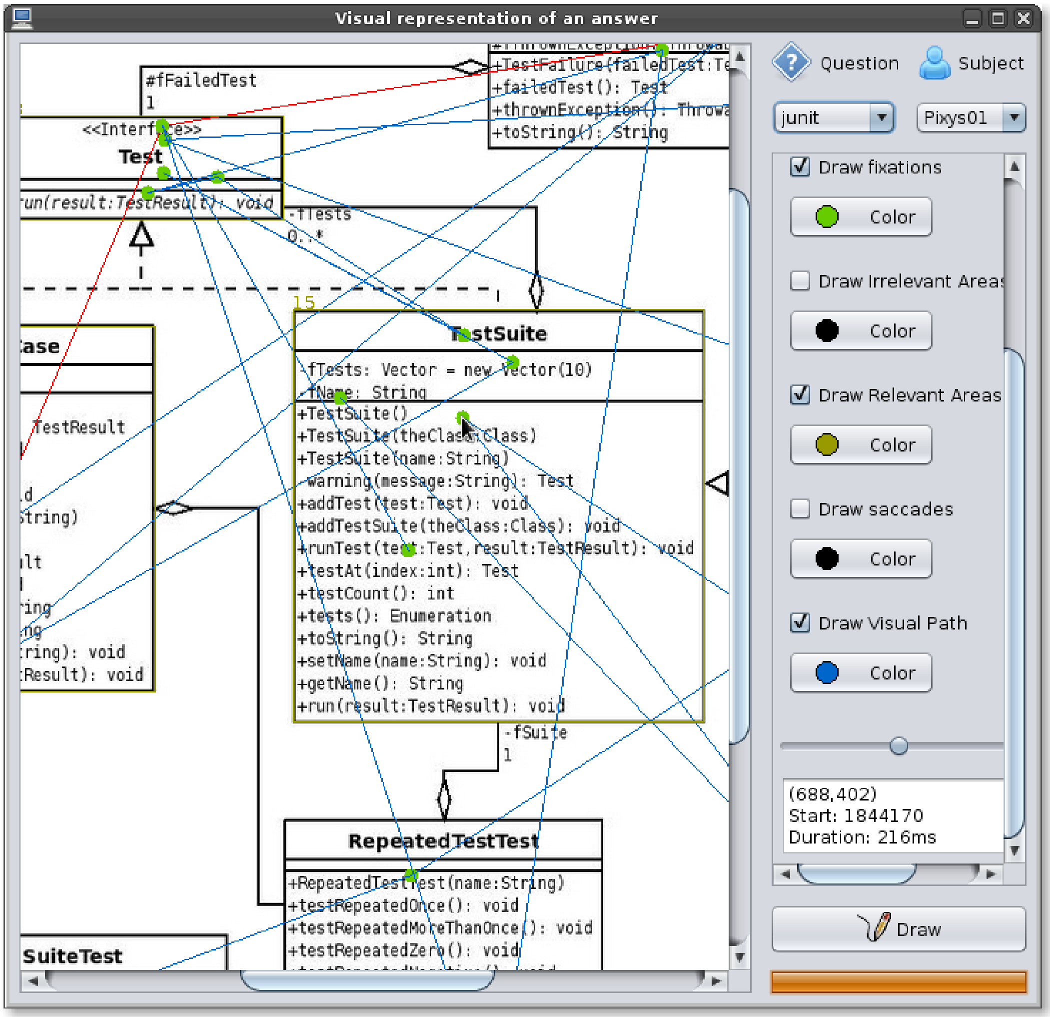


Figure 6: The visualisation tool

can be choosen. You can also load an AOI file if you want to edit it. Once the image is loaded, it is possible to draw a shape directly on the panel by left clicking on different points around the area that is of interest.

2. Once the area of interest is drawn, it is possible to add it to the set of AOI by clicking on `Area Of Interest` → `Add` or by pressing `Ctrl + A`. Some information can be entered about the area of interest such as its identifier (a unique number used to identify the AOI on this image), its name, and its relevance. There are currently three available levels of relevance: *Ignorable*, *Relevant* and *Irrelevant*. *Ignorable* means that the fixations in this AOI will not be taken in account for all the algorithms described in Section 3.

3. Press the `Add` button to add the area of interest. It is, of course, possible to add multiple area of interest on a single image. The `Clear` button can be used to remove the points which are not saved yet.

4. To save the AOI file click on `File` → `Save` or press `Ctrl + S`, then select the place and name of the file you want to save and press `Save`. To remove an AOI simply click

on `Area Of Interest` → `Remove` or press `Ctrl + R`, then select the AOI you want to remove and press the `Remove` button.

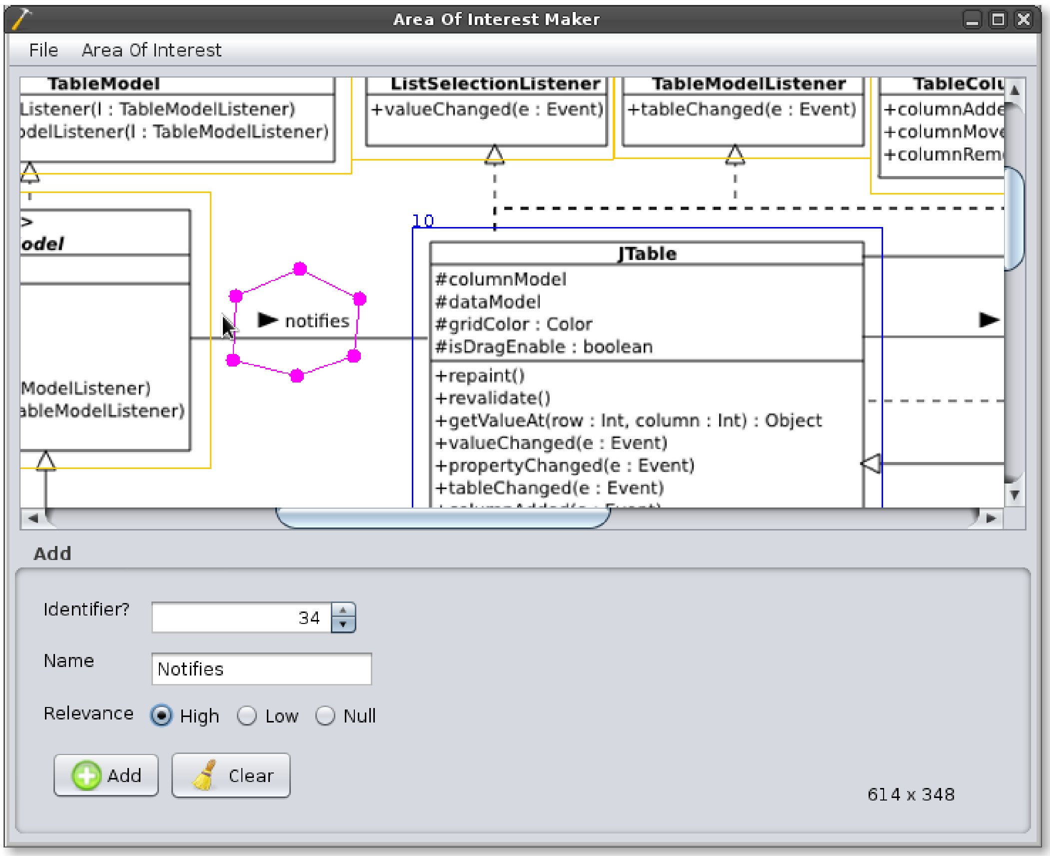


Figure 7: The AOI maker interface

4.6 How to Execute an Algorithm on some Data

To execute an algorithm on the loaded data click on the result button and then select the algorithm to apply on the data and set up its parameters. For each algorithm it is needed to select its printer and its result directory in which the resulting files will be stored. Moreover, it is needed to choose whether the wrong answers have to be taken in account or not (if the fixations corresponding to a task which was not correcly done by the subjects should be used or not).
Once the computation is done the results can be accessed in the directory previously specified.

5 Input Files Structures

5.1 .subject files

5.1.1 General description

A `.subject` file must contain the tag `Subject:` followed by the characteristic of the user followed by its value; for example, `Subject:Expert:1` means that the subject associated with this file is considered as an expert. Actually, there are six possible characteristics implemented in TAUPE:

- `UMLKnowledge:` the UML level of the subject (in [0..2]).
- `DPKnowledge:` the Design Pattern knowledge of the subject (in [0..1]).
- `Gender:` the gender of the subject (`M` — `F`) *(Male – Female)*.
- `Morethan20classes:` whether the subject has already worked on a UML diagrams that contained more than 20 classes (`1`) or not (`0`)
- `Expert:` whether the subject is considered as an expert (`1`) or not (`0`).
- `StudyLevel:` the level of study of the subject (`U` – `B` – `M` – `D`) *(Unknown – Bac – Master – PhD)*.

A `.subject` file must contain the tag `Answer:` followed by the question (name of the image file without its extension) followed by `1` if the subject answered correctly to the question, or `0` otherwise.
ex: `Answer:quickuml:1` means that the subject answered correctly the question asked on the diagram `quickuml`.

5.1.2 Example

Here follows an example of a correct `.subject` file:

```
Subject:UMLKnowledge:2
Subject:DPKnowledge:1
Subject:Gender:M
Subject:Morethan20classes:1
Subject:Expert:1
Subject:StudyLevel:D
Answer:junit:1
Answer:quickuml:1
Answer:argouml:1
```

5.1.3 EBNF

```
[<SubjectCaract> | <AnswerCaract>]*
<SubjectCaract>::= "Subject:"<caracteristic>":"<value> <EOL>
<caracteristic>::= string
<value>::= char
<EOL>::= EndOfLine
<AnswerCaract> ::= "Answer:"<question>":"<validResponse>
<question>::= string
<validResponse> ::= "1" | "0"
```

5.2 .aoi files

5.2.1 General description

An area of interest is an area on a question's image. This area can be relevant or not to a task (relevance is specified in the corresponding `.aoi`). It also can be "ignorable", it means that the system will not take account of this area. Areas are defined in a text file (`AOI files`) for each question. For example, if a question is related to an image called `foo.png`, then the file that defines the areas is named `foo.aoi`.

5.2.2 Example

```
10 NULL Question (17,19) (250,19) (250,65) (17,65)
1 AOI Event (524,83) (727,83) (727,175) (524,175)
2 AORI EventListener (835,83) (835,167) (1015,167) (1015,83)
3 AOI ActionListener (155,215) (395,215) (395,327) (155,327)
7 AORI ChartChangeListener (395,215) (395,310) (613,310) (613,215)
9 NULL ChartProgressListener (288,80) (330,58) (390,63) (390,116) (348,119) (302,107)
```

5.2.3 EBNF

```
[<id> <type> <name> <coordonate> <coordonate> <coordonate>+ <EOL>]*
<id>::= integer
<type> ::= NULL | AOI | AORI
<coordonate> ::= "("integer "," integer")"
<EOL> ::= EndOfLine
```

5.3 relevant.slides file

5.3.1 General description

This file is used to select the relevant images to load on TAUPE. It is composed by a set of image names without the extension (for lisibility reasons). All images not listed in this file will be ignored by the parser.

5.3.2 Example

```
Swing_MD_Q1
Swing_MD_Q2
...
Swing_MVP_Q6
JFreeChart_MD_Q1
JFreeChart_MD_Q2
JFreeChart_MD_Q3
...
JFreeChart_MVP_Q6
```

5.3.3 EBNF

```
[<slide> <EOL>]*
<slide> ::= String
<EOL> ::= EndOfLine
```

6 Eye-tracking systems

This section briefly describes the file structure of the output files from the eye-trackers systems. For more information, please refer to the manual of the corresponding eye-tracker. Typically, an output file is a set of fixations and saccades listed in a text file. There are different output formats but the structure is basicly the same: A time (sometimes a timestamp in milliseconds), a type (fixation or saccade), the coordonates on the screen (x and y) and the duration. Some other information such as the pupil size can also be present in the file.

6.1 Eye-link II

Eye-link's software generates `.edf` files. TAUPE cannot read directly this type of files. A tool named `edf2xml` can be used to generate `.xml` files from the `.edf` files. The `.xml` files can be loaded in TAUPE. This eye-tracker provides some information about the saccades' *peak velocity* and *amplitude* but TAUPE do not currently use this data.

6.2 GazeTracker

GazeTracker software can generate `.out` files that can be loaded in TAUPE using the corresponding parser (4.3). A screencast on how to export data from GazeTracker is available at: `http://www.ptidej.net/research/taupe/`.

7 Shortcuts

This section is a summary of all shortcuts which are available in TAUPE.

7.1 Main Window

`Ctrl + O`: Opens the parser chooser dialog box (see FIGURE 4).

`Ctrl + P`: Opens the preferences window (see FIGURE 3).

`Ctrl + A`: Opens the *About* window.

7.2 AOI Maker

`Ctrl + O`: Allows to load a image file and a AOI file.

`Ctrl + S`: Allows to save the set of AOI created.

`Ctrl + A`: Allows to add the recently drawn AOI to the set of AOI.

`Ctrl + R`: Allows to remove an AOI.

Index

References

[1] Y.-G. Guéhéneuc, "Taupe: Towards understanding program comprehension," in *Proceedings of The Conference of the Center for Advanced Studies on Collaborative Research (CASCON'06)*, October 2006, pp. 1 – 13.

[2] J. H. Goldberg and X. P. Kotval, "Computer interface evaluation using eye movements: methods and constructs," *International Journal of Industrial Ergonomics*, vol. 24, no. 6, pp. 631 – 645, 1999.

[3] G. Robinson, "Dynamics of the eye and head during movement between displays: A qualitative and quantitative guide for designers," *Human Factors*, vol. 21 (3), pp. 343 – 352, June 1979.

[4] P. Viviani, "Eye movements and their role in visual and cognitive processes," *Elsevier Science*, 1990.

[5] A. Poole and L. J. Ball, "In search of salience: A response time and eye movement analysis of bookmark recognition," *People and Computers XVIII-Design for Life: Proceedings of HCI 2004*, 2004.

[6] M. Just and P. A. Carpenter, "Eye fixations and cognitive processes," *Cognitive Psychology*, vol. 8, no. 4, pp. 441 – 480, 1976.

[7] P. M. Fitts, R. E. Jones, and J. L. Milton, "Eye movements of aircraft pilots during instrument-landing approaches," *Aeronautical Engineering Review*, vol. 9, no. 2, pp. 24–29, 1950.

[8] A. Poole and L. J. Ball, "Eye tracking in human-computer interaction and usability research: Current status and future prospects," *Ghaoui, Claude (Ed.). Encyclopedia of Human Computer Interaction*, 2006.

[9] S. Jeanmart, "Evaluation de l'impact d'un patron de conception sur la compréhension et la maintenance de programmes - une expérimentation par un système d'eye-tracking," Master's thesis, Facultés Universitaires Notre-Dame de la Paix, Namur, Belgium, 2008.

[10] R. J. K. Jacob and K. S. Karn, "Commentary on section 4: Eye tracking in human-computer interaction and usability research: Ready to deliver the promises," *Work*, no. 1905, 1958.

[11] G. T. Heineman, G. Pollice, and S. Selkow, *Algorithms in a Nutshell (In a Nutshell (O'Reilly))*. O'Reilly Media, 2008.

[12] S. Yusuf, H. Kagdi, and J. I. Maletic, "Assessing the comprehension of uml class diagrams via eye tracking," in *Proceedings of the 15th IEEE International Conference on Program Comprehension*. Washington, DC, USA: IEEE Computer Society, 2007, pp. 113–122.