

Experiment data, time computation adjustment

Procedure

We collect 3 information to recover the time `totalTime` spent on a task:

- The `duration` time: the total time spent from clicking on `start` to clicking on `finish`. This value is calculated from the logs. It excludes the time taken to answer post-task surveys. **Should it include the time taken to answer to the task?**
- The `interruption` time: the sum of all periods longer than **30 seconds** during which no logs were generated. This value is calculated from the logs.
- The `declaredInterruption` time: an approximate interruption duration given by participants in each post-task survey. The possible values are:
 - Not interrupted (*we consider there was no interruption*) `declaredInterruption = 0`
 - 1-2 minutes (*we consider 1min30 of interruption*) `declaredInterruption = 1m30`
 - up to 5 minutes (*we consider 5 minutes of interruption*) `declaredInterruption = 5min`
 - up to 10 minutes (*we consider 10 minutes of interruption*) `declaredInterruption = 10min`
 - more than 10 minutes (*we consider 10 minutes of interruption*) `declaredInterruption = 10min`

To compute the time spent for a participant, we apply the following manual procedure.

Case 1 (145 detected) `interruption` and `declaredInterruption` are consistent, we consider the computed interruption time as reference:

```
totalTime := duration - interruption
```

Examples:

- `interruption` = 4 minutes and `declaredInterruption` = "up to 5 minutes". We count 4 minutes of interruption.
- `interruption` = 3 minutes and `declaredInterruption` = "1-2 minutes". We count 3 minutes of interruption.

Case 2 (7 detected) `interruption` is 0 (*i.e.*, no period longer than 30 seconds found in logs), and the participant declared they were interrupted.

We take the lower bound of the declared interruption time in their answer:

```
totalTime := duration - interruption
```

Examples:

- `interruption` = 0 and `declaredInterruption` = "up to 5 minutes". We count 5 minutes of interruption.

- **interruption** = 0 and **declaredInterruption** = "1-2 minutes". We count 1 minute of interruption.
- **interruption** = 0 and **declaredInterruption** = "up to 10 minutes" or "more than 10 minutes". We count 10 minutes of interruption.

This can happen if participants were interrupted or their concentration diverted for multiple periods shorter than 30 seconds.

We therefore consider the answer of the participant as reference.

For 10 minutes and more, we can only reasonably consider a total of 10 minutes of interruption.

We have no means of knowing exactly the interruption time, and such long interruption is unlikely to happen in batches of less than 30 seconds.

Case 3 (633 detected) **interruption** is 0 and **declaredInterruption** was **none** in the survey or was not answered (e.g., it was just not answered by the participant).

We count no interruption as none was detected.

Case 4 (36 detected)

interruption and **declaredInterruption** are not consistent:

- The participant declared no interruption, but the system detects interruption times in the logs.
- The declared time is not consistent with the effective duration of the task (e.g., the declared interruption time is higher than the task execution time and it would put the task execution time below 0)
- The participant declared an interruption time, but the system detected a different interruption time and both times are considerably different.

Examples:

- **interruption** = 2 minutes and **declaredInterruption** = "Not interrupted".
- **interruption** = 24 hours and **declaredInterruption** = "up to 10 minutes" or "more than 10 minutes".
- The total execution time of the task is measured to 7 minutes, but the participant declared more than 10 minutes of interruption, which, if deduced, would produce a negative task execution time.

For each case, we manually look at the logs and determine if the detected interruption can safely be assumed to be an interruption.

For each case, we justify our decision in a comment that is saved as metadata associated with the participant's data.

We decide that a detected interruption is really an interruption only if we have strong evidence that this is really an interruption time.

For example, if the participant mentioned in their answer that they had a precise interruption, or if the interruption clearly appear in the logs (e.g., a 24 hours gap is clearly an interruption).

Case 5 (145 detected)

The detected **interruption** time is lower than the **declaredInterruption**, for example

Examples:

- **interruption** = 2 minutes and **declaredInterruption** = "Up to 5 minutes".
- **interruption** = 6 minutes and **declaredInterruption** = "up to 10 minutes" or "more than 10 minutes".

It is possible participant declared a safe interruption time because they lost track of it or because they estimate that time wrongly.

In any cases, we cannot decide anything on a declared time that is not detected by the system.

Therefore, we interpret these cases as **case 1** occurrences, *i.e.*, we deduce the detected interruption time.

Case 6 (7 detected)

Implementation numbering error. All cases 6 are cases 2, but may be labelled as **case 6** in the data.

Procedure setup

To perform the procedure failing in the aforementioned cases, all tasks of all participation corresponding to one of these cases are presented in a GUI list:

| | | |
|------------|----------------------|------------|
| 0:00:02:51 | No choice | 0:00:00:00 |
| 0:00:00:45 | No choice | 0:00:00:00 |
| 0:00:02:49 | 1-2 minutes | |
| 0:01:11:24 | more than 10 minutes | |
| 0:14:54:49 | more than 10 minutes | |
| 0:00:13:35 | more than 10 minutes | 0:00:00:00 |
| 0:00:01:27 | up to 5 minutes | 0:00:00:00 |

The data is presented for each of these tasks with only the detected interruption duration, the declared interruption duration and the total amount of retained duration.

We do not know if the task under observation is a control or a treatment task, nor for which participant.

For each line in the list, we can either automatically fix the time if it fails into one of the obvious cases.

When it does not (*e.g.*, case 4), the **Fix idle times...** action opens a new GUI that presents the logs where the gap was detected.

[32 seconds] Case 4: computed a
[1 minute 38 seconds] Case 4: cor
[7 minutes 32 seconds] Case 4: cc
[1 minute 42 seconds] Case 4: cor
[6 minutes 14 seconds] Case 4: cc
[12 minutes 39 seconds] Case 4: c
[1 minute 55 seconds] Case 4: cor
[2 minutes 3 seconds] Case 4: cor
[4 minutes 4 seconds] Case 4: cor
[3 minutes 26 seconds] Case 4: cc
[10 minutes 42 seconds] Case 4: c
[1 minute 11 seconds] Case 4: cor
[1 minute 17 seconds] Case 4: cor
[34 seconds] Case 4: computed a
[49 seconds] Case 4: computed a
[17 minutes 19 seconds] Case 4: c
[36 seconds] Case 4: computed a
[32 seconds] Case 4: computed a
[1 minute 13 seconds] Case 4: cor
[31 seconds] Case 4: computed a
[2 minutes 30 seconds] Case 4: cc

Idle time analyzer

Time gap betweenAll events:
Mouse on text editLeave window
Mouse on text editMouse on text editor
Enter window
Leave window
Enter window
Mouse on text editor
Mouse on text editor
Leave window
Mouse on text editor
Enter window

: Variable

self
Σ dateTime
Σ windowId
{ } uuid
itemElement

Comment:
Case 4: computed and declared interruption are not consistent.
The participant is reading.
Decision: not an interruption.

Remove time: false

Reset decision

Deduce time

Keep time

For tasks failing in **case 4**, we look at these logs to try to understand if the detected time is really an interruption and write down in a comment field the evidence supporting this claim.

In that case, we might observe logs indicating that we are observing the data of a treatment task (e.g., object-centric breakpoint logs).

In the following screenshot, in the first columns we see the two events between which the time gap was detected.

When we select one of these events, it is automatically selected in the second column in the flow of events of the task, and the third column shows the details of that event.

| Time gap betweenAll events: | | : Variable | : Value |
|--------------------------------|---------------------------------------|--------------|---------------------------------------|
| Mouse on text editEnter window | | self | Window opened |
| Window opened | Window closed | Σ dateTime | 2024-07-10T16:03:14.394597+02:00 |
| | Leave window | Σ windowId | 1063643648 |
| | Enter window | { } uuid | an UUID [16 items] (21 178 35 216 70) |
| | Mouse on text editor | ¶ windowName | a ClyFullBrowserMorph(46113536) |
| | Window opened | | |
| | Browse | | |
| | Leave window | | |
| | Enter window | | |
| | Mouse on table item: starting | | |
| | Mouse on table item: instance created | | |

For transparency, we report that information in the comment.

An example of case 4 time deduction:

| Idle time analyzer | | | |
|---|----------------------|---------------|----------------------------------|
| Time gap betweenAll events: | | : Variable | : Value |
| Leave window | Mouse on table item | self | Leave window |
| Leave window | Mouse on table item | Σ dateTime | 2023-12-26T10:30:33.741-05:00 |
| | Enter window | Σ windowId | 1012957440 |
| | Leave window | { } uuid | an UUID [16 items] (200 230 176) |
| | Enter window | ¶ itemElement | |
| | Leave window | | |
| | Leave window | | |
| | Mouse on table item | | |
| | Enter window | | |
| | Enter window | | |
| | Leave window | | |
| | Leave window | | |
| | Enter window | | |
| | Mouse on text editor | | |
| | Enter window | | |
| | Leave window | | |
| | Mouse on table item | | |
| | Enter window | | |
| | Leave window | | |
| Comment: | | | |
| Case 4: computed and declared interruption are not consistent. | | | |
| The participant spends 6 minutes outside the system windows without doing anything. | | | |
| Decision: interruption. | | | |

Threats to validity

It is possible that participants have a biased appreciation of their interruption time (e.g., they have a wrong impression that they were interrupted a lot), or that they declare shorter or larger interruption times for many reasons (e.g., they are not totally sure, or they do not remember because the task took all their concentration).

It is also possible that a detected a time gap was in fact the participant taking notes however.

If we have a doubt and cannot strongly backup the claim that the detected time gap was an interruption, we

do not count that time gap as an interruption.

Examples

Case 4: computed and declared interruption are not consistent.

The participant spends about 38 seconds outside the system windows without doing anything.

However, just after the participant moves their mouse over a text editor and starts navigating.

It is possible that the participant was reading that text editor.

Decision: not an interruption.

Case 4: computed and declared interruption are not consistent.

The participant moves their mouse over items in a list, with an interruption of 35 seconds.

This case cannot be decided with any strong argument.

Decision: not an interruption.

Case 4: computed and declared interruption are not consistent.

The participant is reading.

Decision: not an interruption.

Case 4: computed and declared interruption are not consistent.

The participant spends about 3 minutes outside the system windows without doing anything.

Decision: interruption.

Each time there is an immediate (by checking time) text editor, a read event, or that such event follows the event gap, we consider that the participant is reading.

In some cases we can't detect the declared time by the participant, and we can't make a decision to deduce time while analyzing the detection.

In that case we trust the participant, and we force the retained idle time as a case 2 ==> redo a complete pass over these cases