

Supervising Attention in an E-learning System

Dalila Durães^{1,3}[0000-0003-3336-2449], Javier Bajo¹[0000-0003-0824-4133] and Paulo Novais²[0000-0002-3549-0754]

¹Department of Artificial Intelligence, Technical University of Madrid, Madrid, Spain
d.alves@alumnos.upm.es, jbajo@fi.upm.es

²Algoritmi Center, University of Minho, Braga, Portugal
pjon@di.uminho.pt

³CIICESI, ESTG, Polytechnic Institute of Porto, Felgueiras, Portugal

Abstract. Until now, the level of attention of a worker has been evaluated through his/her productivity: the more one produces, the better his/her attention at work. First, the worst aspect about this approach is that it only points out a potential decrease of attention after a productivity loss. An approach that could point out, in advance, upcoming breaks in attention could allow active/preventive interventions rather than reactive ones. In this paper we present a distributed system for monitoring attention in teams (of people). It is especially suited for people working with computers and it can be interesting for domains such as the workplace or the classroom. It constantly analyzes the behavior of the user while interacting with the computer and together with knowledge about the task, is able to temporally classify attention.

Keywords: Distributed Intelligent System, Attention, Behavioral Biometrics.

1 Introduction

In the field of computer science, an intelligent environment is a digitally augmented physical world where sensor-enabled and networked devices work continuously and collaboratively to make the lives of the inhabitants more comfortable. With this technological evolution, job offers have changed, bringing along many significant and broad changes. Some of the most notorious ones can be pointed out by the emergence of indicators such as attentiveness which, in extreme cases, can compromise the life and well-being of the workers. In more moderate cases it will impair attention, general cognitive skills and productivity. In addition to these factors, many of these jobs are the so-called desk-jobs, in which people frequently sit for more than 8 hours [1].

Until now, the level of attention of a worker has been evaluated through his/her productivity: the more one produces, the better his/her attention at work. While the true nature of this relationship is yet to be thoroughly studied (properly contextualized in each work domain), there are other issues that need to be addressed. First, the worst aspect about this approach is that it only points out a potential decrease of attention after a productivity loss. This means that the “damage” is already done and that it is most likely too late for the worker to cope with whatever caused the attention loss. An approach that could point out, in advance, upcoming breaks in attention (e.g. through the observation of behavioral patterns) could allow active/preventive interventions rather than reactive ones [2].

In this paper we present a distributed system for monitoring attention in teams (of people), in line with the vision of intelligent environments [3]. It is especially suited for people working with computers and it can be interesting for domains such as the workplace or the classroom. It constantly analyzes the behavior of the user while interacting with the computer and, together with knowledge about the task, is able to temporally classify attention.

This work may be very interesting for team managers to assess the level of attention of their teams, identifying potentially distracting events, hours or individuals. Moreover, distraction often appears when the individual is fatigued, bored or not motivated. This tool can thus be an important indicator of the team, allowing the manager to act accordingly at an individual or group level. In the overall, this tool will support the implementation of better human resources management strategies.

1.1 Previous Work

Part of this framework was implemented in previous work. The first version focused on the analysis of the individuals’ interaction patterns with the computer, including features such as mouse velocity or acceleration, click duration, typing speed or rhythm, among others. For a complete list of features as well as the process of their acquisition and extraction, please see [4]. However, a limitation was also identified in this previous work. In fact, a user that opened a no work-related application and did not interact anymore with the computer until the end of the task had 0% of attention. On the other hand, if the user opens a work-related application and does not interact with the computer after that, the user’s attention will be classified as 100% when he is most likely not even at the computer.

The present work adds a new feature to this previously existing framework, by providing a precise measure of attention based not on the tasks work-related patterns but also on the key typing or mouse movement pattern. It thus constitutes a much more precise and reliable mechanism for attention monitoring, while maintaining all the advantages of the existing system: nonintrusive, lightweight and transparent.

2 Architecture

From the architecture of the developed environment described in Figure 1 it is possible to collect data that describe the interaction with both the mouse and the keyboard in the devices in which students work. Three parts compose the module's architecture.

In the devices operating by the students, it's installed software that generates raw data, which store locally until it is synchronized with the web server in the cloud. In this step it's encodes each event with the corresponding necessary information (e.g. timestamp, coordinates, type of click, key pressed). These data are further processed, stored and then used to calculate the values of the behavioral biometrics. Mouse movements can also help to predict the state of mind of the user, as well as keyboard usage patterns.

The first part (HTTP Service) is responsible for all database connections, with the needed configuration to access to the MySQL database and the DB Module feature to access to the Mongo DB, and the needed transactions with it.

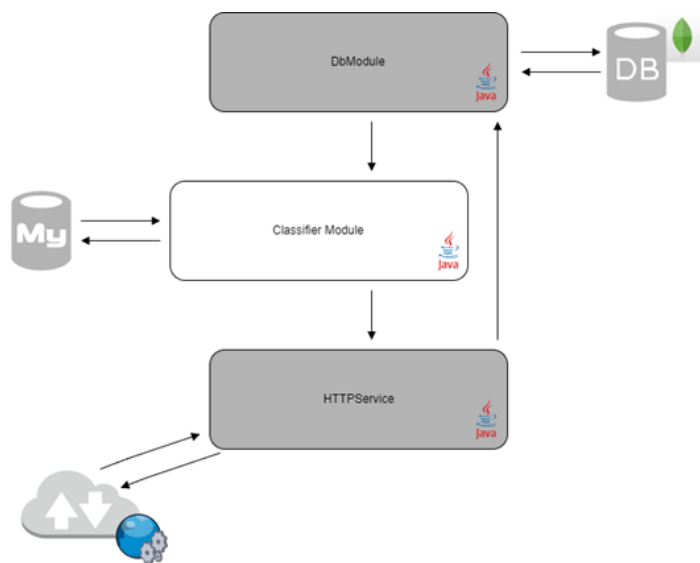


Fig. 1. System architecture.

The second part (DbModule) is the REST connection implementation. Here were built all the necessary connection the access to the system. After the raw data is stored in a data store engine, the analytic layer processes the data received (from the storage layer) in order to evaluate those data according to the metrics presented. Some data preparation tasks are also carried out in this phase, such as removing outliers (e.g. the backspace key being continuously pressed to delete a group of characters is not a regular key press). The system receives this information in real-time and calculates, at regular intervals, an estimation of the general level of performance and attention of each student.

The third part (Classifier Module) is features implementation of attention classification. This classifier will handle the treatment of the data into useful information, doing for instance the necessary calculations to reach the attention index. All the data analysis and processing were developed in this part. In the classification profile layer the indicators are interpreted. Based on data from the attentiveness indicators and building the meta-data that will support decision-making, the system will classify the user profile. When the system has a sufficiently large dataset that allows making classifications with precision, it will classify the inputs received into different attention levels in real-time, creating each student learning profile.

Finally, the user attention information send back to the user managers, where is displayed in the visualization layer in the users managers, and it can be used to personalize instructions according to the specific user, enabling the administrator to act differently with different users as well as to act differently with the same user, according to his/her past and present level of attention. In the visualization layer it is possible to obtain some graphical modules that allow the display of information in an intuitive way to the user.

2.1 Data Acquisition

As previously mentioned, the early version of this system acquired data describing the interaction of the user with the computer in terms of the mouse and the keyboard [5]. Data acquisition is supported by an application installed in the data-generating devices

(the computers of the users). It is thus a distributed data collection system, which has now been extended to acquire a new type of data about each user: the application that the user is interacting with, at each time.

Thus, at regular intervals (around 5 minutes), the server receives a new set of data about each user. This raw data includes all the important interaction events: when keys were pressed down or released, when the mouse moved (and where to), when clicks started or ended and when the user switched to a given application. This data is then transformed by the server as described in [5], to generate 15 features that describe the performance of the interaction with the computer.

In previous work [6] to identify the work-related applications, we analyzed all different applications used by all users and labeled each one as belonging to the task or not. We then quantified the amount of time that each student spent interacting with applications related to the task versus other applications. To do this we measured the amount of time, in each interval, that the user spent interacting with work-related applications. The algorithm thus needs knowledge about the domain in order to classify each application as belonging or not to the set of work-related applications. This knowledge is provided by the teacher and is encoded in the form of regular expressions. The teacher uses a graphical interface to set up rules such as "starts_with Access" or "contains Access", which are then translated to regular expressions that are used by the algorithm to determine which applications are work-related and which are not. Whenever an application that does not match any of the known rules for the specific domain is found, the application name is saved so that the teacher can later decide if a new rule should or should not be created for it. By default, applications that are not considered work-related are marked as "others" and count negatively towards the quantification of attention.

2.2 Features Extraction

Firstly, it is necessary to know the interaction with the mouse and with the keyboard that each task will have. In some tasks the interaction with the mouse will be higher while in others that will happen with the keyboard, and finally in others, it will practically be the same level of interaction. However, it is difficult to know, a priori, what the exact percentage of interaction of the mouse or the keyboard will be. To know these values, we first count the number of times that each key is pressed and the number of times that the mouse is clicked down for each user.

3 Validation

To validate the proposed system, we have been using it for the past months in the Caldas das Taipas High School, located in northern Portugal. In the Portuguese academic context, this system gains increased relevance as current policies move towards the creation of larger classes, which make it increasingly difficult for the teacher to individually address to each student. To validate this system we are following several cohorts of students during their academic activities. This data collection process will allow assessing the influence on attention of aspects such as: breaks, time of day, class contents, and class objectives, among others. For this purpose, a group of 14 (all girls) students were selected to participate in this experience. Their average age is 15.9 years old ($SD = 1.5$ years). The experiment was applied in a lesson, where they had access to an individual computer and 100 minutes to complete the task. Students received, at the beginning of the lesson, all necessary data with the goals of the task. For this class, the lesson started in the afternoon and students received, at the beginning of the lesson, a document with the goals of the task, which in this case required the use of Microsoft Access and Adobe Acrobat Reader.

This application runs in the background, which makes the data acquisition process, a completely transparent one from the point of view of the student. It collects data from the students' interaction with the mouse and the keyboard, which act as sensors. The Mouse and Keyboard Sensing layers are responsible for capturing information describing the behavioral patterns of the students while interacting with the peripherals.

Secondly, we calculated the percentage of the interaction of the mouse, which is the average number of times that the mouse was pressed in the class taking into account the total of interactions with the mouse and the keyboard of the class. Similarly, we calculated the percentage of the interaction of the keyboard, which is the average number of times that the keys were pressed in the class taking into account the total of interactions with the mouse and the keyboard of the class.

On the other hand, we obtained the amount of time that each student spent at the computer (Task Duration) as well as the amount (and percentage) of time that each student devoted to work and to other activities in the lesson as explained in [6]. However, the time that the student spent in the task-related doesn't indicate the level of attention in some cases, because in some situations the user might have opened the task-related and during that time didn't interact with the computer. In this case, if the level of attention was measured only by the task-related, the student has 100% of attention and in reality his attention level should be 0%. For these situations, it is necessary to analyze the amount of interaction with the mouse and the keyboard and cross these data.

The next step is to calculate the percentage of usage of the mouse for each user. The formula to calculate this value is:

$$IM = \frac{\sum Md}{Max Md} \quad (1)$$

Where:

- IM: is interaction of the mouse;
- Md: is the number of the times that the mouse was pressed by a user;
- MAX Md: is the higher score of usage of the mouse down in the class.

Similarly, for calculating the percentage of usage of the keyboard for each user, the formula for this value is:

$$IK = \frac{\sum Kd}{Max Kd} \quad (2)$$

Where:

- IK: is Interaction of the keyboard;
- Kd: is the number of the time that one key was pressed by a user;
- Max Kd: is the higher score of usage of the Keyboard in the class.

Finally, in order to obtain the level of attention it is necessary to combine the results of the interaction of the behavior biometrics with the results showed in the time that each student spent in the task-related. This level of attention is the relative attention of each student, as this level of attention is compared with the other user of the class. Formula (3) provides the results for each user, presented in Table 3. The formula (3) that calculated these values is:

$$Rel.At = ((PercM * Work_rel) + (PercK * Work_rel)) \quad (3)$$

Where:

- Rel.At: is Relative attention;
- PercM: is the percentage of utilization of the Mouse for a user;
- PercK: is the percentage of usage of the Keyboard for a user;

Table 1. Relative attention of each user in the lesson

User	Relative MouseDown	Relative KeyDown	Work-related	Relative Attention
T22400	31,45%	30,22%	0,00%	0,00%
T22400	31,70%	15,59%	78,70%	37,22%
T22400	16,95%	11,92%	69,73%	20,13%
T22400	48,43%	26,55%	56,77%	42,56%
T22400	21,92%	10,92%	56,95%	18,70%
T22400	43,71%	21,75%	60,60%	39,67%
T22400	17,06%	7,11%	61,81%	14,94%
T22400	44,89%	25,05%	83,73%	58,56%
T22400	25,64%	24,13%	88,58%	44,09%
T22400	24,51%	15,61%	61,77%	24,78%
T22400	38,29%	25,50%	83,72%	53,41%
T22400	32,93%	23,36%	85,60%	48,19%
T22400	45,25%	51,57%	75,52%	73,11%
T22400	33,20%	24,65%	70,21%	40,62%

Figure 2 presents the graphically result between percentage of time interacting with work-related applications and relative attention. It can be observed that the relative attention decreases in general, because the system takes in account the interaction of the user with mouse and keyboard. That is, someone who spent 100% of the time using the application that was supposed to but was interacting only 50% of the time, would have a score of attention of 50%.

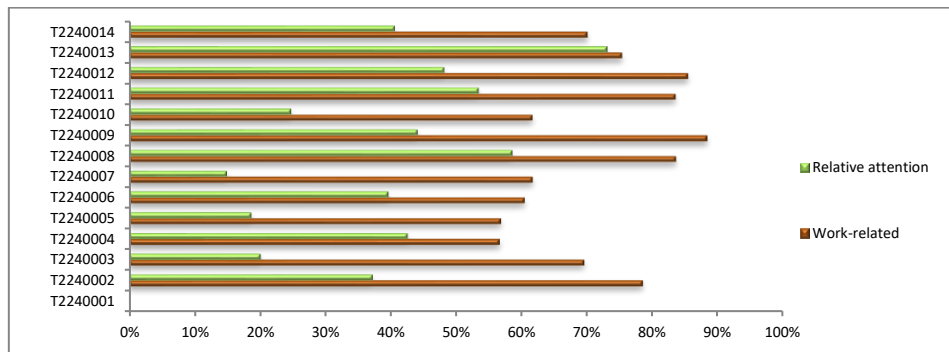


Fig. 2. Comparing work-related interaction with relative attention for each student.

4 Discussion and Conclusions

The task data from the user's interaction with the computer is the most crucial information because it derives most part of the attention level. To obtain the task results, the task rules received in the request are used to get a perception of how much time the user has spent on the applications related with the task rules.

As far as the mouse and keyboard results are concerned, their role is present on how the user interaction is occurring and it helps the manager/teacher to understand if any user, who has the application active, is really working on it or not. To do that, we have used the two features one from the keyboard (key down time) and other from the mouse (time between clicks).

Concerning attention, it is an important theme because it is one of the factors that most influences a person's performance while performing a task. Understanding how attention varies and in which situations the attention varies for each person, it is possible to act in the right moment and right time, to bring the user's attention level to the ideal value.

A framework was proposed to address these issues, especially to monitoring user. Narrowing the scope of the study, a process to detect attentiveness was proposed, through the use of a developed log tool. With this smart environment it is possible to detect potentially negative factors dynamically and non-intrusively, making it possible to foresee negative situations, allowing taking actions to mitigate them. The work developed so far resulted in a useful system for the team manager, who can monitor, in real-time, the level of attention of users.

The door is thus open to intelligent platforms that allow to analyze user's profiles, taking into account their individual characteristics, and to propose new strategies and actions. By providing managers with access to this information, we allow them to better manage their interactions with the users, namely by pointing out the most problematic cases of inattention in real-time.

Acknowledgements

This work has been supported by COMPETE: POCI-01-0145-FEDER-007043 and FCT – Fundação para a Ciência e Tecnologia within the Project Scope: UID/CEC/00319/2013.

References

- Liao, M., & Drury, C. (2000). Posture, Discomfort and Performance in a vdt task. *Ergonomics* 43 (3), 45–35.
- Carneiro, D., Novais, P., Andrade, F., Zeleznikow, J., & Neves, J. (2013). Using Case-Based Reasoning and Principled Negotiation to Provide Decision Support for Dispute Resolution. *Knowl. Inf. Syst.* 36 (3), 789–826.
- Augusto, J., Callaghan, V., Cook, D., Kameas, A., & Satoh, I. (2013). Intelligent environments: a Manifesto. *Human-Centric Computing and Information Sciences* 3(1), 1 – 18.
- Durães, D., Jiménez, A., Bajo, J., & Novais, P. (2016). Monitoring Level Attention Approach in Learning Activities. In M. Caporuscio, F. De la Prieta, T. Di Mascio, & P. Vittorini, *Advances in Intelligent System and Computing*, 478 (pp. 33-40).
- Durães, D., Gonçalves, S., Carneiro, D., Bajo, J., & Novais, P. (2016). Detection of Behavioral Patterns for Increasing Attentiveness Level. In International Conference on Intelligent Systems Design and Applications ISDA 2016: Intelligent Systems Design and Applications, pp 592-601.
- Carneiro, D., Durães, D., Bajo, J., & Novais, P. (2016). Non-intrusive Monitoring of Attentional Behavior in Teams. In International Symposium on Intelligent and Distributed Computing IDC 2016: Intelligent Distributed Computing X pp 153-162.