# Non-intrusive Monitoring of Attentional Behavior in Teams

Davide Carneiro[1,3], Dalila Durães[2], Javier Bajo[2], and Paulo Novais[3]

[1] CIICESI, ESTGF, Polytechnic Institute of Porto, Portugal
[2] Department of Artificial Intelligence, Technical University of Madrid, Madrid, Spain
[3] Algoritmi Center/Department of Informatics, University of Minho, Braga, Portugal

**Abstract.** Attention is a very important cognitive and behavioral process, by means of which an individual is able to focus on a single aspect of information, while ignoring others. In a time in which we drawn in notifications, beeps, vibrations and blinking messages, the ability to focus becomes increasingly important. This is true in many different domains, from the workplace to the classroom. In this paper we present a non-intrusive distributed system for monitoring attention in teams of people. It is especially suited for teams working at the computer. The presented system is able to provide real-time information about each individual as well as information about the team. It can be very useful for team managers to identify potentially distracting events or individuals, as well as to detect the onset of mental fatigue or boredom, which significantly influence attention. In the overall, this tool may prove very useful for team managers to implement better human resources management strategies.

**Keywords:** Attentional Behavior, Non-intrusive, Monitoring, Distributed Computing

## 1 Introduction

Attention is a very complex process through which one individual is able to continuously analyze a spectrum of stimuli and, in a sufficiently short amount of time, chose one to focus on [1]. In most of us, which can only focus on a very reduced group of stimuli at a time, this implies ignoring other perceivable stimuli and information.

Research on attention involves nowadays many fields, including education, psychology, neuroscience, cognitive neuroscience and neuropsychology. For this reason, many different views and theories on attention can be found. One of the most frequent ones is the so-called *Attention Economics*, which treats human attention as a scarce commodity or resource, which we must use wisely in order to attain our goals [2].

Individuals who have difficulties in focusing their attention can see the performance of other high-level cognitive processes negatively affected, such as learning or decision making. In extreme cases, such as in Attention-Deficit/Hyperactivity Disorder (ADHD), this may have a significant negative impact in the development and function of the individual [3].

Although these aspects have always existed, in the last years we have witnessed the increasing of distracting stimuli, which make this topic a still more important one. Nowadays we have to deal with constant notifications from our e-mail, our social networks, our messaging applications, advertisements and so on. We live immerse in beeps, vibrations, notifications and blinking icons, which constantly call for our attention and distract us [4]. Even if we return immediately to our task, the fact that we had to consciously evaluate the stimuli to decide that it is not important at the moment already had a toll on our brain, making it spend resources [2, 5].

This is especially worrying in young children, who nowadays have a facilitated access to computers, mobile phones and tablets, with their games and engaging applications. For them it is so easy to get distracted by these stimuli, making learning less efficient and more frustrating, negatively affecting their development [6].

In this paper we present a distributed system for monitoring attention in teams of people, in line with the vision of Intelligent Environments[7]. It is especially suited to people working with computers and can be interesting for domains such as the workplace or the classroom. It constantly analyzes the behavior of the user while interaction 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 the framework presented in this paper was implemented in previous work. This first version focused on the analysis of people's 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 [8].

While this early work focused on the detection of stress [8] and mental fatigue [9] from the analysis of Human-Computer Interaction, we also found out that people tend to interact differently with different applications and in different contexts. For example, and although both tasks involve typing, people tend to type differently if they are in a messaging application and in an word processing application [10].

For relatively simple domains in which people interact with a reduced and known set of applications, we achieved correct classification rates of around 90% [11]. However, in more complex scenarios and, especially, if there are no limits on the possible applications, the accuracy of the classification tends to decrease.

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

## 2 Architecture

The architecture of the developed system (Figure 1) is divided in three major parts. The lower-level is composed by the devices that generate the raw data (e.g. computers). These devices store the raw data locally in SQLite databases, until it is sychnronized with the web servers in the cloud, which happens at regular intervals.

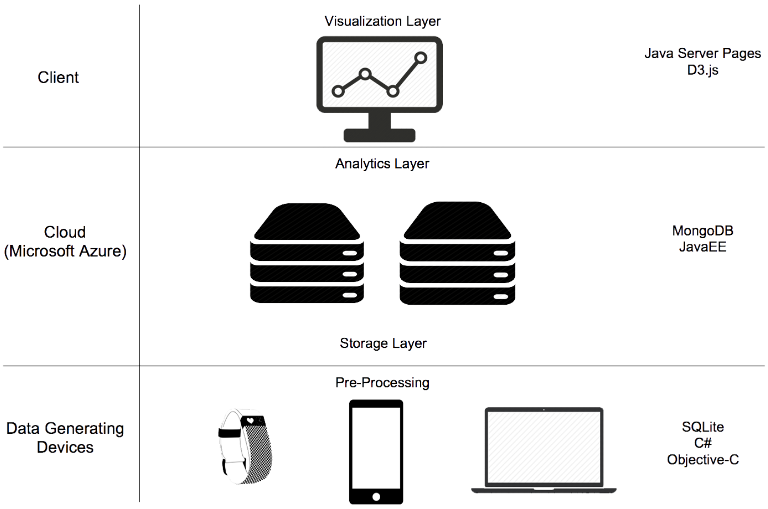


**Fig. 1.** Architecture of the developed system.

The main element in the middle layer is a MongoDB database, which is half way between relational and non-relational systems. It is actually more than a data storage engine, as it also provides native data processing tools: MapReduce and the Aggregation pipeline. Both the aggregation pipeline and map-reduce can operate on a shared collection (partitioned over many machines, horizontal scaling). These are powerful tools for performing analytics and statistical analysis in real-time, which is useful for ad-hoc querying, pre-aggregated reports, and more. MongoDB provides a rich set of aggregation operations that process data records and return computed results, using this operations in the data layer simplifies application code and limits resource requirements.

Finally, the visualization layer (top-most layer) is developed as a web app on Java technology and uses the D3 library for graphics and diagrams. It includes a set of intuitive data visualization tools to facilitate decision making and human resources management, with a focus on individual and group performance real time analytics.

### 2.1 Data Acquisition

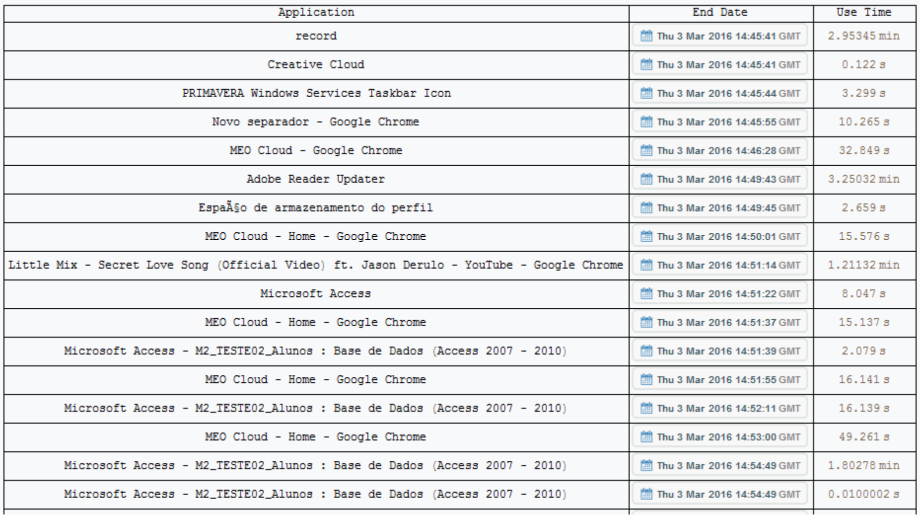
As previously mentioned, the early version of this system acquired data describing the interaction of the user with the computer in terms of the mouse and the keyboard [9]. Data acquisition is supported by an application installed in the data-generating devices (the computers of the users). It is thus 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 [8], to generate 15 features that describe the performance of the interaction with the computer.

In the early version of the system, the server classified the level of attention of the user based on the interaction patterns, as described previously in this paper. With this work, the server now has access to a new type of raw data denoting that a given user switched to a specific application at a given timestmap. Since the server receives this data at regular intervals, it receives a list of triplets for each user. Thus, the new raw data received at regular intervals is as follows: $(Id, Username, Timestamp, [(AppName, Timestamp)])$. Id denotes the unique identifier of this group of data, Username uniquely identifies the user, Timestamp marks the end of the time interval for this group of data, and the last element is a list of pairs containin, in each element, the name of the app that the user switched to and the timestamp in which this happened.

### 2.2 Feature Extraction

When the server receives the data described in the previous subsection, it transforms it so its features can be extracted. Specifically, it goes through the list of pairs and computes the time during which each window was active (Algorithm 1). There are often cases in which the user does not change applications for a large amount of time. In these cases, which are represented by a pair with an empty AppName, the time is added to the last known AppName (since this means that the user is still interacting with it). An example of the output of this process is depicted in Figure 2.



| Application | End Date | Use Time |
|---|---|---|
| record | Thu 3 Mar 2016 14:45:41 GMT | 2.95345 min |
| Creative Cloud | Thu 3 Mar 2016 14:45:41 GMT | 0.122 s |
| PRIMAVERA Windows Services Taskbar Icon | Thu 3 Mar 2016 14:45:44 GMT | 3.299 s |
| Novo separador - Google Chrome | Thu 3 Mar 2016 14:45:55 GMT | 10.265 s |
| MEO Cloud - Google Chrome | Thu 3 Mar 2016 14:46:28 GMT | 32.849 s |
| Adobe Reader Updater | Thu 3 Mar 2016 14:49:43 GMT | 3.25032 min |
| EspaÃ§o de armazenamento do perfil | Thu 3 Mar 2016 14:49:45 GMT | 2.659 s |
| MEO Cloud - Home - Google Chrome | Thu 3 Mar 2016 14:50:01 GMT | 15.576 s |
| Little Mix - Secret Love Song (Official Video) ft. Jason Derulo - YouTube - Google Chrome | Thu 3 Mar 2016 14:51:14 GMT | 1.21132 min |
| Microsoft Access | Thu 3 Mar 2016 14:51:22 GMT | 8.047 s |
| MEO Cloud - Home - Google Chrome | Thu 3 Mar 2016 14:51:37 GMT | 15.137 s |
| Microsoft Access - M2_TESTE02_Alunos : Base de Dados (Access 2007 - 2010) | Thu 3 Mar 2016 14:51:39 GMT | 2.079 s |
| MEO Cloud - Home - Google Chrome | Thu 3 Mar 2016 14:51:55 GMT | 16.141 s |
| Microsoft Access - M2_TESTE02_Alunos : Base de Dados (Access 2007 - 2010) | Thu 3 Mar 2016 14:52:11 GMT | 16.139 s |
| MEO Cloud - Home - Google Chrome | Thu 3 Mar 2016 14:53:00 GMT | 49.261 s |
| Microsoft Access - M2_TESTE02_Alunos : Base de Dados (Access 2007 - 2010) | Thu 3 Mar 2016 14:54:49 GMT | 1.80278 min |
| Microsoft Access - M2_TESTE02_Alunos : Base de Dados (Access 2007 - 2010) | Thu 3 Mar 2016 14:54:49 GMT | 0.0100002 s |

**Fig. 2.** Sequence of applications used by a specific student, with the date in which the student switched to other application and the time spent interacting with it.

```
Data:
p - A list of pairs of the type (AppName, Timestamp)
ft - the finishing time of the task
Result: durations - A list of triplets of the type (AppName, Timestamp,
        Duration)
durations ← [];
i ← 0;
while $i < Size(p)$ do
    task ← $p_{i,1}$;
    time ← $p_{i,2}$;
    i++;
    while $i < Length(p)$ and $StringLength(p_{i,1}) = 0$ do
        i++;
    end
    if $i = Length(p)$ then
        AppendTo(durations, task, ft, ft - time);
    else
        AppendTo(durations, task, $p_{i,1}$, $p_{i,1}$-time)
    end
end
```

**Algorithm 1:** Creating triplets with the durations and timestamp of each application.

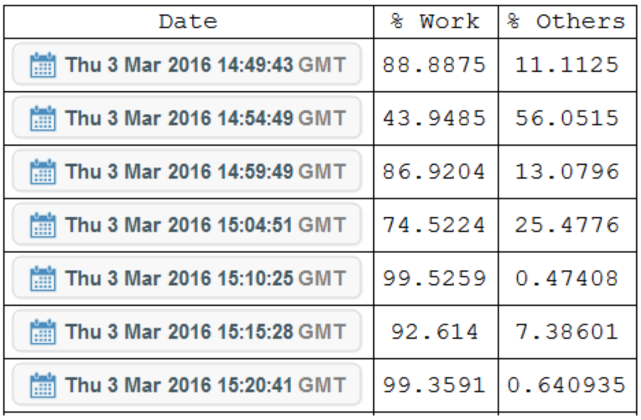
The next step is to compute the level of attention of the user (Algorith, 2. To do this we measure the amount of time, in each interval, that the user spent interacting with work-related applications. The algorithm thus needs knowledge about the domain in order to classify each application as belonging or not to the set of work-related applications. This knowledge is provided by the team administrator and is encoded in the form of regular expressions. The team administrator uses a graphical interface to set up rules such as "starts with Microsoft" or "Contains word Adobe", which are then translated to regular expressions that are used by the algorithm to determine which applications are work-related.

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 team manager 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. Attention is calculated at regular intervals, as configured by the team manager (e.g. five minutes). The output of the algorithm can be visualized in Figure 3.

```
Data:
t - A list of triplets of the type (AppName, Timestamp, Duration)
st - The starting time of the task
inter - The interval to update attention
set - the set of regular expressions
Result: attention - A list of triplets of the type (timestamp, attention%,
        others%)
attention ← [];
i ← 0;
work ← 0;
others ← 0;
time ← st;
while $i$ < $Size(t)$ do
    if $isWork(t_{i,1}, set)$ then
        work ← work + $t_{i,3}$;
    else
        others ← others + $t_{i,3}$;
    end
    if $t_{i,2} > time + inter$ then
        AppendTo(attention, $t_{i,2}$, $work * 100/(work + others)$,
          $others * 100/(work + others)$);
        work ← 0;
        others ← 0;
        time ← $t_{i,2}$;
    end
end
```

**Algorithm 2:** Creating triplets at regular intervals with the timestamp and the quantification of attention.

With this approach, several interesting functionalities can be implemented that provide valuable information to improve the team manager's decision making processes. Examples of such functionalities are described in Section 3.



| Date | % Work | % Others |
|---|---|---|
| Thu 3 Mar 2016 14:49:43 GMT | 88.8875 | 11.1125 |
| Thu 3 Mar 2016 14:54:49 GMT | 43.9485 | 56.0515 |
| Thu 3 Mar 2016 14:59:49 GMT | 86.9204 | 13.0796 |
| Thu 3 Mar 2016 15:04:51 GMT | 74.5224 | 25.4776 |
| Thu 3 Mar 2016 15:10:25 GMT | 99.5259 | 0.47408 |
| Thu 3 Mar 2016 15:15:28 GMT | 92.614 | 7.38601 |
| Thu 3 Mar 2016 15:20:41 GMT | 99.3591 | 0.640935 |

**Fig. 3.** Detailed evolution of attention of a specific student.

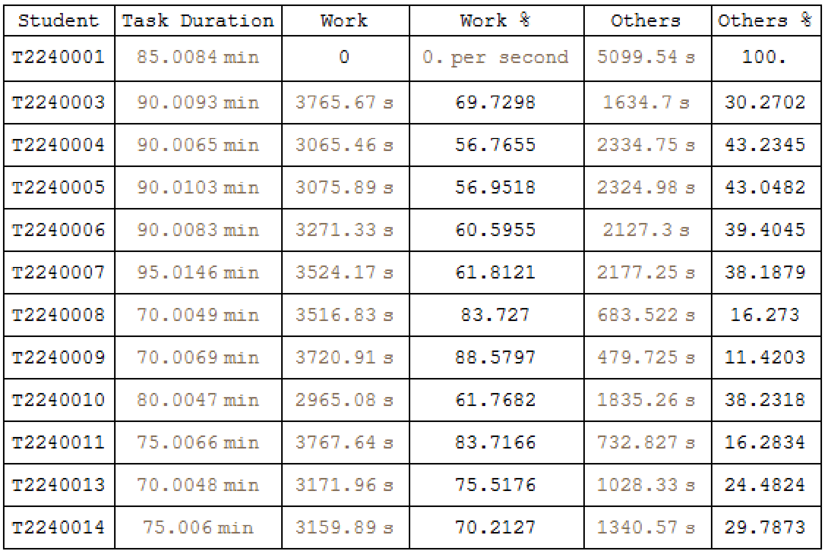
## 3 Validation

As previously mentioned, a system with these characteristics may prove useful in very different domains, including organizational, academic or any environment in which people operate computers. In order 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 or larger classes, which make it increasingly difficult for the professor to individually address each student. In this section we show several tools supported by this system that, when at the disposal of the professors, may allow to:

- Decide, in real-time, in which students to focus, according to their level of attention;
- Evaluate, *a posteriori*, which contents are more prone to generate distraction, providing a chance for improvement;
- Identify, in real-time, fluctuations in attention, improving decision-making concerning aspects such as when to make breaks or when to dismiss the class.

To validate this system we are following several cohorts of students during their academic activities. This extensive data collection process will allow to assess the influence on attention of aspects such as: breaks, time of day, class contents, class objectives, among others. In this section, as an example, we briefly analyzed the data collected for the same cohort of students (10N) in two different classes: a regular one and an assessment one. Apart from the aims, the conditions were the same: the same cohort of students working on similar tasks, which required the use of Microsoft Access and Adobe Acrobat Reader.

Figure 4 allows the professor to analyze, at the end of the class, 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. This is important for the professor to perform a self-evaluation of how the class took place.



| Student | Task Duration | Work | Work % | Others | Others % |
|---|---|---|---|---|---|
| T2240001 | 85.0084 min | 0 | 0. per second | 5099.54 s | 100. |
| T2240003 | 90.0093 min | 3765.67 s | 69.7298 | 1634.7 s | 30.2702 |
| T2240004 | 90.0065 min | 3065.46 s | 56.7655 | 2334.75 s | 43.2345 |
| T2240005 | 90.0103 min | 3075.89 s | 56.9518 | 2324.98 s | 43.0482 |
| T2240006 | 90.0083 min | 3271.33 s | 60.5955 | 2127.3 s | 39.4045 |
| T2240007 | 95.0146 min | 3524.17 s | 61.8121 | 2177.25 s | 38.1879 |
| T2240008 | 70.0049 min | 3516.83 s | 83.727 | 683.522 s | 16.273 |
| T2240009 | 70.0069 min | 3720.91 s | 88.5797 | 479.725 s | 11.4203 |
| T2240010 | 80.0047 min | 2965.08 s | 61.7682 | 1835.26 s | 38.2318 |
| T2240011 | 75.0066 min | 3767.64 s | 83.7166 | 732.827 s | 16.2834 |
| T2240013 | 70.0048 min | 3171.96 s | 75.5176 | 1028.33 s | 24.4824 |
| T2240014 | 75.006 min | 3159.89 s | 70.2127 | 1340.57 s | 29.7873 |

**Fig. 4.** The amount of time that each student spent interacting with the computer and the amount of actual work versus the amount spent interacting with other applications.

If necessary, the professor may also click on a student to analyze the temporal evolution of the attention for that specific student, in a given clas. Figure 5 shows the evolution of attention for three specific students during the class.

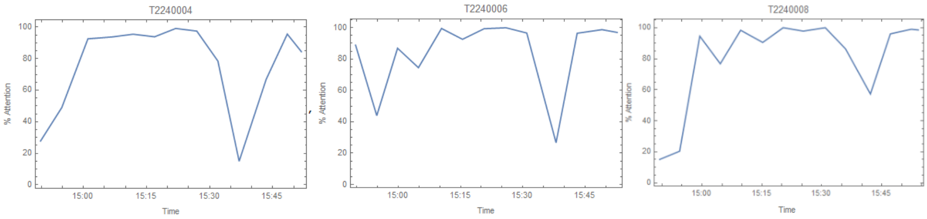


**Fig. 5.** Temporal evolution of attention in three different students of the same class.

The professor may also find it very important to assess, in real-time or *a posteriori*, the evolution of attention of the whole class. To this end, the professor may select which cohorts to compare and in which clsses. Figure 6 shows the global evolution of the attention of cohort 10N, in a regular class (a) and in an assessment class (b). This visual representation is constructed by combining data from all the students and computing a running average. Finally, several summarization techniques are also available, with the aim of providing the professor with simple and intuitive insights into the data. As an example, Figure 6 (c) shows the distribution of the values of attention in cohort 10N, in a regular class and in an assessment one.

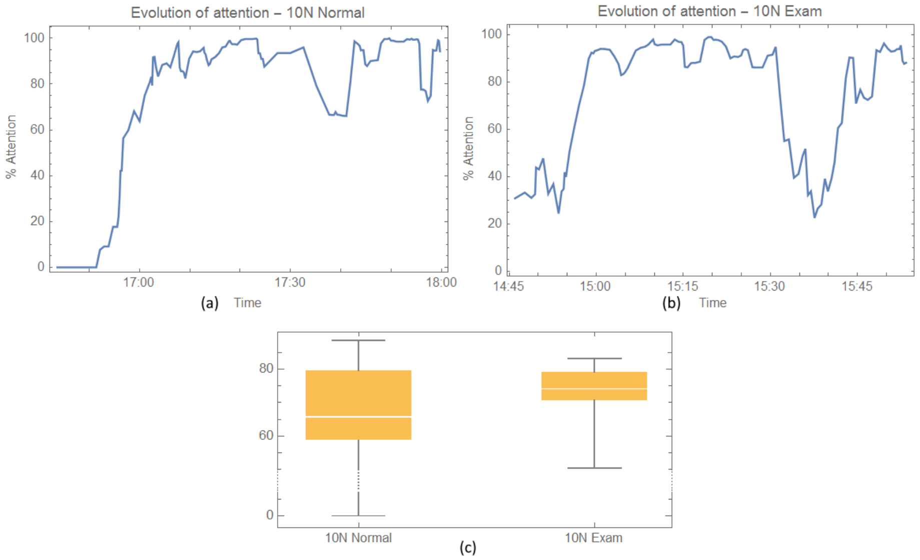


**Fig. 6.** Temporal evolution of attention in the same cohort of students (10N) in two different situations: regular class (a) and assessment (b). Comparing the overall attention of both groups (c).

## 4 Limitations and Future Work

The work developed so far resulted in a very useful system for team managers to monitor, in real-time, the level of attention of their team. However, a limitation was also identified, of which user T2240001 (depicted in Figure 4) is an example. This is a user that, according to the collected data, has 0% of attention throughout the 85 minutes of the task. In fact, what happened is that the user opened an application that is not work related and did not interact anymore with the computer until the end of the task. Similarly, 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. These cases must, evidently, be pointed out.

To address this limitation, in future work we will implement a tighter integration between the previous and the new work. Specifically, in previous work we implemented the monitoring of the interaction of the users with the keyboard and the mouse. It is thus possible to know all the actions that each user performed either with the mouse or the keyboard, and at what time. We will thus generate a new feature that will quantify the level of activity of each user throughout time. This new feature will allow a more contextualized analysis of attention, improving the performance of its classification and quantification.

## Acknowledgments

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. The work of Davide Carneiro is supported by a Post-Doctoral Grant by FCT (SFRH/BPD/109070/2015).

## References

1. Estes, W.K.: Handbook of Learning and Cognitive Processes (Volume 4): Attention and Memory. Psychology Press (2014)
2. Davenport, T.H., Beck, J.C.: The attention economy: Understanding the new currency of business. Harvard Business Press (2013)
3. ATTENTION-DEFICIT, S.O., et al.: Adhd: clinical practice guideline for the diagnosis, evaluation, and treatment of attention-deficit/hyperactivity disorder in children and adolescents. Pediatrics (2011) peds–2011
4. McBride, D.L.: Distraction of clinicians by smartphones in hospitals: a concept analysis. Journal of advanced nursing **71**(9) (2015) 2020–2030
5. Simola, J., Hyönä, J., Kuisma, J.: Perception of visual advertising in different media: from attention to distraction, persuasion, preference and memory. Frontiers Media SA (2015)
6. Gottlieb, J.: Attention, learning, and the value of information. Neuron **76**(2) (2012) 281–295
7. Augusto, J.C., Callaghan, V., Cook, D., Kameas, A., Satoh, I.: Intelligent environments: a manifesto. Human-Centric Computing and Information Sciences **3**(1) (2013) 1–18
8. Carneiro, D., Novais, P., Pêgo, J.M., Sousa, N., Neves, J.: Using mouse dynamics to assess stress during online exams. In: Hybrid Artificial Intelligent Systems. Springer (2015) 345–356
9. Pimenta, A., Carneiro, D., Novais, P., Neves, J.: Monitoring mental fatigue through the analysis of keyboard and mouse interaction patterns. In: Hybrid Artificial Intelligent Systems. Springer (2013) 222–231
10. Pimenta, A., Carneiro, D., Novais, P., Neves, J.: Detection of distraction and fatigue in groups through the analysis of interaction patterns with computers. In: Intelligent Distributed Computing VIII. Springer (2015) 29–39
11. Carneiro, D., Pimenta, A., Gonçalves, S., Neves, J., Novais, P.: Monitoring and improving performance in human–computer interaction. Concurrency and Computation: Practice and Experience (2015)