

Behavioral Biometrics and Ambient Intelligence: new opportunities for Context-Aware Applications

Davide CARNEIRO^{a,b,1}, Paulo NOVAIS^b

^a*CIICESI, ESTG, Polytechnic Institute of Porto, Felgueiras, Portugal*

^b*ALGORITMI Centre/Department of Informatics, University of Minho, Braga, Portugal*

Abstract. Ambient Intelligence has always been associated with the promise of exciting new applications, aware of the users' needs and state, and proactive towards their goals. However, the acquisition of the necessary information for supporting such high-level learning and decision-making processes is not always straightforward. In this chapter we describe a multi-faceted smart environment for the acquisition of relevant contextual information about its users. This information, acquired transparently through the technological devices in the environment, supports the building of high-level knowledge about the users, including a quantification of aspects such as performance, attention, mental fatigue and stress. The environment described is particularly suited for milieus such as workplaces and classrooms, in which this kind of information may be very important for the effective management of human resources, with advantages for organizations and individuals alike.

Keywords. Ambient Intelligence, Intelligent Environments, Behavioral Biometrics, Context-Aware Computing

1. Introduction

Since its first moments that Ambient Intelligence has promised, among other exciting features, the development of environments that are sensitive to its users' needs and desires [1]. This sensitiveness implies not only ways to understand the state of the users but also the intention to take actions that maximize user satisfaction and support day-to-day activities. Other chapters on this book address these issues in detail [2,3].

This, albeit simple when put in words, results extremely complex in practice. From a technological standpoint many challenges still exist such as how to integrate so many different technologies, how to do it in a transparent way (i.e. to develop intelligent environments that look like traditional ones) or how to acquire contextual information from the users without explicitly asking for it. For a more in-depth discussion on the technological challenges (among others) of implementing AmI systems, please refer to the survey by Preuveneers et al., in this book [4].

¹Corresponding Author: Davide Rua Carneiro, Universidade do Minho, Campus de Gualtar, Braga, Portugal; E-mail: dcarneiro@di.uminho.pt

On the other hand, there are also those challenges that arise from the nature of these environments and their inhabitants: humans and their interactions are intrinsically complex. Each individual user has unique desires, objectives, actions and behaviors. Moreover, these may be conflicting between different users. One significant challenge is thus how to accommodate all this in a single environment, with so many changing variables.

In this chapter we focus on a single of these problems, but one that is very significant: the non-intrusive and transparent acquisition of information that can be used to characterize the state of each individual. In this work, *non-intrusive* refers to methods that can be used, for continued periods of time, without causing any noticeable change in the users' routines. *Transparent* refers to methods that are, as the term implies, invisible, i.e., nothing in the environment or in the way it is being used hints towards their existence.

The significance of this work lies in the fact that it may support the development of truly transparent data-collection environments. Moreover, it may give access to new forms of information that were either never explored before or where analyzed in very specific domains.

Specifically, we target environments such as classrooms or workplaces, with the aim to acquire information for the modeling of contextual information that can be used to quantify stress, fatigue, performance or attention[5]. The advantages in generating and using this kind of information include a better and more contextualized management of human resources, better adaptation of teaching/working methodologies, adjustment of working/teaching rhythms, among many others [6].

Indeed, while some of the features (or other related features) have been used in the past to address some of these issues, in this chapter we show that the specific set of features presented can be used to characterize the user's stress, performance and attention to task, making this a very versatile set of features. Moreover, much of the existing work focuses on features extracted from the use of mobile devices which, given their embedded sensors and communication capabilities, are especially useful for this purpose (see for example [7,8]). In this chapter we focus on features extracted from the computer's mouse and keyboard: two peripherals that have been significantly less studied for this purpose.

The remaining of the chapter is organized as follows. Section 2 describes the relevance and richness of Human behavior and introduces the concepts of Behavioral Biometrics, mouse dynamics and keystroke dynamics as non-intrusive approaches to assess the behavior of computer users. Section 3 details an environment for the acquisition of data that can be used to characterize behavior. Specifically, this section focuses on the architecture of the environment and on the behavioral features that are extracted from the collected data, providing a first insight into the potential applications of this approach. Sections 4 to 6 describe the use of this kind of environments in three real cases of application: performance in stressful environments (with a use case in computer-based exams), performance in the workplace (with a use case in real workplaces with the aim to assess worker fatigue) and quantification of task attention (with a use case in a high school). These three sections thus show to some extent the potential real applications of the proposed environment. However, many more application scenarios exist, which are further explored in Section 7, together with a realistic analysis of the advantages and shortcomings of the environment described.

2. Behavior and Behavioral Biometrics

In this chapter we address the challenge of acquiring contextual information about the users of an intelligent environment [9], in a non-intrusive way. To achieve this, we look at Humans' interactions in real life to conclude that when interacting with one another, our behavior passes along a significant amount of information about ourselves. More than that, it provides the appropriate context for the interpretation of our interaction. For example, the same sentence may be perceived differently if the speaker is screaming it or speaking it softly.

Based on this premise, we assume that we may also adjust our behavior when we interact with a computer according to our state. Indeed, there are nowadays fields such as Human-Computer Interaction that look at these and other aspects of the interaction of the human with the machine. Specifically, the field of Behavioral Biometrics precisely studies our behavior in this interaction [10].

The name borrows from the more traditional field of biometrics, which uses human physical or physiological characteristics that are virtually unique for each individual, including fingerprints, iris or face recognition, palm print or veins, among others [11]. Moreover, in traditional biometrics, these characteristics are used mostly for the purpose of personal identification.

Behavioral Biometrics, on the other hand, rely on behavioral traits of the individual such as typing rhythm, gait, voice, among others. Behavioral Biometrics can and have also be used for identification purposes, since each one of us behaves in a very individual and unique way. However, our behavior is also likely to change under different circumstances or according to our inner state. For example, the action of identifying an individual based on the speech may be hindered by the individual being stressed or not since stress affects our pitch or our speech rhythm.

Nonetheless, while this may be a disadvantage when the aim is to identify an individual, it may not be so if the aim is to identify changes in the state or behavior of the individual: knowing how a individual usually behaves allows to detect significant behavioral changes, which may in turn indicate changes in the inner state of the individual.

In the field of Behavioral Biometrics, the mouse and the keyboard are frequently used as the source of valuable inputs for the analysis of behavioral patterns, with approaches known respectively as mouse and keystroke dynamics. These two approaches have been consistently used in the last years for a wide range of different purposes.

In [12], the authors use a total of 8 interaction features to characterize user interaction: session time, keystroke latency, dwell time, sequence, typing speed, frequency of error, pause rate and capitalization rate. While these features are collected participants also fill questionnaires to assess their emotional state, which allows to train classifiers for human emotion recognition based on typing patterns.

Typing behavior against positive/negative emotions has also been studied by [13] in a similar approach. The study focused on the valence of emotion (positive and negative) and considered keystroke duration and latency as interaction features. Fifteen participants volunteered for the study. The authors conclude that all participants show significant differences in typing patterns when feeling positive and negative emotions, elicited through facial feedback [14]. Further emotion recognition methods based on keystroke dynamics and mouse movements can be found in [15].

Other researchers have looked at typing behavior with different objectives: to assess the effect of mental load and music in word processing tasks [16]. In this study the

features considered are typing force, typing productivity, and electromyography of the left hand *extensor digitorum* muscle. The overall conclusions are that typing productivity is compromised by music and that there is a reduction of wrong finger touch during typing. Music also resulted in an increased *extensor digitorum* muscle activity for lifting and controlling fingers. However, the validity of these conclusions may be limited since only 8 individuals participated in this study.

However, one of the most traditional fields of application of Behavioral Biometrics is undoubtedly the one of user authentication. In what concerns the use of the mouse, both holistic features (single-click statistics, double-click statistics, movement offset and movement elapsed time) and procedural features (speed curve against time and acceleration curve against time) can be used to characterize mouse movement. These features are used in [17], in a study with 37 participants that obtained satisfying acceptance rates in their identification with only 11.8 seconds of interaction.

While the work presented in [17] focuses mostly on mouse movement, the work of [18] considers only mouse clicking. Specifically, the authors considered five features that model clicking rhythm, which quantify different timings between clicks and during clicks. The combination of these two types of approaches could thus further improve the accuracy of identification.

Mouse dynamics can also be used for the same purpose. In [19], several of such approaches are analyzed. The authors review existing authentication approaches based on mouse dynamics and shed light on some important limitations regarding how the effectiveness of these approaches has been evaluated in the past. As a conclusion, the authors also present the results of several experiments conducted by them to illustrate their observations and suggest guidelines for evaluating future authentication approaches based on mouse dynamics.

Behavioral Biometrics have also been used for assessing people's level of stress and mental fatigue. In the last years we have been studying the interaction patterns of people with computers and smartphones, to build models that can be used in real time for classifying stress and fatigue [20,21]. The aim is to develop software and hardware that is sensitive to the user's state, adapting accordingly.

Other authors have also looked at Behavioral Biometrics for similar purposes. Vizer et al. [22] are able to classify cognitive and physical stress with accuracy rates comparable to those currently obtained using affective computing methods, using keystroke and linguistic features of spontaneously generated text. A case-based approach relying on Behavioural Biometrics is used by [23] to determine a user's stress level. On a somehow related approach, the authors of [24] use keystroke analysis to detect boredom and engagement during writing.

The amount and variety of existing work shows the interest of the research community in this field as well as its potential. In the specific case of the work detailed in this chapter we look at Behavioral Biometrics, specifically Keyboard and Mouse Dynamics, to acquire contextual information about individuals in the environment. In that sense, the proposed system is especially suited for environments in which people frequently interact with computers so that the amount of collected data is extensive. For this reason, we target especially environments such as modern workplaces (notably white-collar jobs) and educational environments, in which the use of the computer is nowadays standard. To a large extent this also defined the behavioral features detailed in Section 3.

3. An Environment for the Acquisition of Behavioral Features

3.1. Architecture

In this section we describe the technological environment mentioned in Section 1, designed to support the acquisition of contextual information in a continuous, transparent and non-intrusive way. Specifically, this environment was developed aiming at the acquisition of behavioral features, guided by the conviction that the behavior of the individuals in the environment tells a great deal about their state.

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

A lightweight application is installed in these devices, that implements the data collection procedure. Depending on the configuration of the environment, a log in may be required for the data collection to take place. This ensures that each source of data is correctly attributed to each user. In what concerns the data collection procedure, this is the only point in which some interaction between the environment and the user may occur (i.e. login). Other than that, the client application runs in background in a completely transparent way.

While data can be collected from all these kinds of devices, in this chapter we focus on data collected from the mouse and the keyboard and on its use in environments such as offices, where the use of these peripherals is standard.

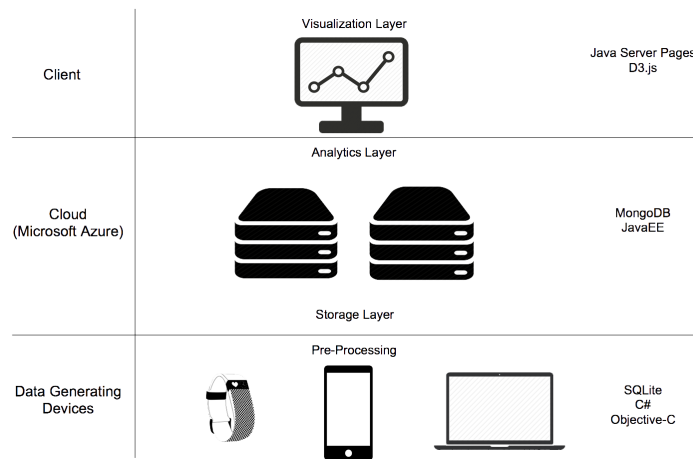


Figure 1. High-level view of the architecture of the environment.

The main element in the middle layer is a Mongo database. MongoDB is half way between relational and non-relational systems. It provides indexes on collections, it is lockless and provides a query mechanism. MongoDB provides atomic operations on fields like relational systems and supports automatic sharding by distributing the load across many nodes with automatic failover and load balancing. MongoDB supports replication with automatic failover and recovery. The data is stored in a binary JSON-like

format called BSON that supports boolean, integer, float, date, string and binary types. The communication is made over a socket connection (in CouchDB it is made over an HTTP REST interface).

MongoDB 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 shard 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. Section 3.2 details how the data is automatically aggregated, processed and the features extracted.

In what concerns fault tolerance MongoDB provides master-slave replication and replica sets. Nowadays, replica sets are recommended for most use cases. The standard (and minimum) number of replicas in a set is three: one being the primary (the only one with writes allowed), and two secondaries (can become the primary in an election), since an odd number of members ensures that the replica set is always able to elect a primary. MongoDB also provides pluggable storage engines, namely WiredTiger and MMAPv1. Multiple storage engines can co-exist within a single MongoDB replica set, making it easy to evaluate and migrate engines. Running multiple storage engines within a replica set can also simplify the process of managing the data lifecycle. WiredTiger (default storage engine starting in MongoDB 3.2) will provide significant benefits in the areas of lower storage costs, greater hardware utilization, and more predictable performance and, consequently should be used in this system.

Finally, the visualization layer (topmost 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.

3.2. *Extraction of Behavioral Features*

The data generating devices listen to system events and it is these events that allow a posterior extraction of features that characterize the behavior of the user. Although a significant amount of contextual information may be extracted from devices such as smartphones and tablets [25], in this chapter we focus on the information that can be extracted from the users' interactions with computers. Hence, each data generating device logs, locally, the following events and their details:

- MOV, timestamp, posX, posY
An event describing the movement of the mouse, in a given time, to coordinates (posX, posY) in the screen;
- MOUSE_DOWN, timestamp, [Left|Right], posX, posY
This event describes the first half of a click (when the mouse button is pressed down), in a given time. It also describes which of the buttons was pressed (left or right) and the position of the mouse in that instant;
- MOUSE_UP, timestamp, [Left|Right], posX, posY

An event similar to the previous one but describing the second part of the click, when the mouse button is released;

- **MOUSE_WHEEL**, timestamp, dif
This event describes a mouse wheel scroll of amount dif, in a given time;
- **KEY_DOWN**, timestamp, key
Identifies a given key from the keyboard being pressed down, at a given time;
- **KEY_UP**, timestamp, key
Describes the release of a given key from the keyboard, in a given time;
- **APP**, timestamp, name
Describes the moment in which a given application moved to the foreground, i.e., when the user started interacting with it.

The following example depicts a log excerpt extracted from a local database that starts with some mouse movement (first two lines), contains a click with a little drag (lines 3-5) in which the user switched to a different application (line 6), and some more movement (last two lines).

```
MOV, 635296941683402953, 451, 195
MOV, 635296941684123025, 451, 197
MOUSE.DOWN, 635296941684443057, Left, 451, 199
MOV, 635296941685273140, 452, 200
MOUSE.UP, 635296941685283141, Left, 452, 200
APP, 635296941685283283, Microsoft Access – M2_TESTE02
MOV, 635296941685723185, 452, 203
MOV, 635296941685803193, 454, 205
```

These logs are generated individually, in each device. In order for the data collection to occur, the user must log in the client application. This means that a user can use different devices in the environment, in different times, and allow the collection of data from all these devices.

The individual logs generated by the aforementioned application in the data generating devices are then processed in the database in order to compile information that can efficiently characterize the behavior of the users while in the environment. By default, this information is averaged in groups of 5 minutes of data, to facilitate posterior data analysis. Nonetheless, the database stores the raw data, the processed data and the final averaged data.

We now detail all the features that are generated by the database. It is important to note that these features were designed with the aim to assess behavioral changes in the interaction of the users with the computer. To a large extent, these features also quantify the performance of this interaction. Taking as example the movement of the mouse, one never moves it in a straight line between two points, there is always some degree of curve. The larger the curve, the less efficient the movement (much like when driving a car from point A to point B).

An interesting property of these features is that, except for mouse velocity and acceleration, an increasing value denotes a decreasing performance (e.g. longer click \Rightarrow poorer performance, larger average excess of distance \Rightarrow poorer performance). Concerning mouse velocity and acceleration, the relationship is not straightforward. While up to a certain point they might indicate better performance, after that point people have a smaller degree of control, i.e., less precision.

3.2.1. Mouse

The following features are extracted from the mouse:

ABSOLUTE SUM OF ANGLES (ASA)

UNITS - degrees

This feature seeks to find how much the mouse “turned”, independently of the direction to which it turned (Figure 2 (a)). In that sense, it is computed as the absolute of the value returned by function $degree(x_1, y_1, x_2, y_2, x_3, y_3)$, as depicted in equation 1.

$$rCls_angle = \sum_{i=0}^{n-2} | degree(posx_i, posy_i, posx_{i+1}, posy_{i+1}, posx_{i+2}, posy_{i+2}) | \quad (1)$$

AVERAGE DISTANCE OF THE MOUSE TO THE STRAIGHT LINE (ADMSL)

UNITS - pixels

This feature measures the average distance of the mouse to the straight line defined between two consecutive clicks. Let us assume two consecutive MOUSE_UP and MOUSE_DOWN events, mup and mdo , respectively in the coordinates (x_1, y_1) and (x_2, y_2) . Let us also assume two vectors $posx$ and $posy$, of size n , holding the coordinates of the consecutive MOUSE_MOV events between mup and mdo . The sum of the distances between each position and the straight line defined by the points (x_1, y_1) and (x_2, y_2) is given by 2, in which $ptLineDist$ returns the distance between the specified point and the closest point on the infinitely-extended line defined by (x_1, y_1) and (x_2, y_2) . The average distance of the mouse to the straight (Figure 2 (b)) line defined by two consecutive clicks is this given by s_dists/n .

$$s_dists = \sum_{i=0}^{n-1} ptLineDist(posx_i, posy_i) \quad (2)$$

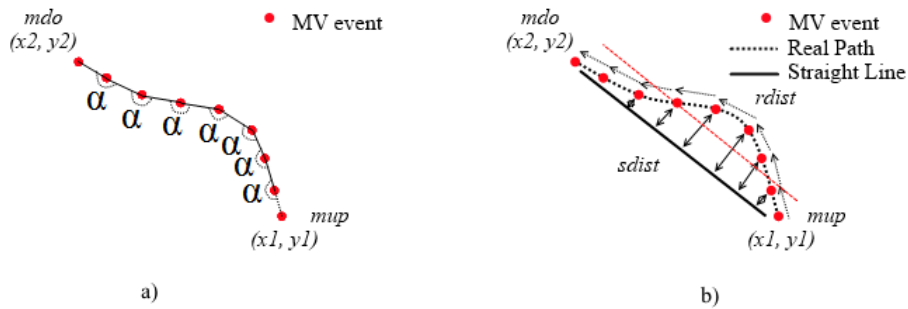


Figure 2. (a) The sum of the angles of the mouse’s movement is given by summing all the angles between each two consecutive movement vectors. (b) The average distance at which the mouse is from the shortest line between two clicks is depicted by the straight dashed line.

AVERAGE EXCESS OF DISTANCE (AED)

UNITS - pixels

This feature measures the average excess of distance that the mouse travelled between each two consecutive MOUSE_UP and MOUSE_DOWN events. Let us assume two consecutive MOUSE_UP and MOUSE_DOWN events, *mup* and *mdo*, respectively in the coordinates (x_1, y_1) and (x_2, y_2) . To compute this feature, first it is measured the distance in straight line between the coordinates of *mup* and *mdo* as $s_dist = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$. Then, it is measured the distance actually travelled by the mouse by summing the distance between each two consecutive MOUSE_MV events. Let us assume two vectors *posx* and *posy*, of size *n*, holding the coordinates of the consecutive MOUSE_MV events between *mup* and *mdo*. The distance actually travelled by the mouse, *real_dist* is given by equation 3. The average excess of distance between the two consecutive clicks (Figure 3 (a)) is thus given by r_dist/s_dist .

CLICK DURATION (CD)

UNITS - milliseconds

Measures the timespan between two consecutive MOUSE_UP and MOUSE_DOWN events.

DISTANCE BETWEEN CLICKS (DBC)

UNITS - pixels

Represents the total distance travelled by the mouse between two consecutive clicks, i.e., between each two consecutive MOUSE_UP and MOUSE_DOWN events. Let us assume two consecutive MOUSE_UP and MOUSE_DOWN events, *mup* and *mdo*, respectively in the coordinates (x_1, y_1) and (x_2, y_2) . Let us also assume two vectors *posx* and *posy*, of size *n*, holding the coordinates of the consecutive MOUSE_MOV events between *mup* and *mdo*. The total distance travelled by the mouse is given by equation 3.

$$r_dist = \sum_{i=0}^{n-1} \sqrt{(posx_{i+1} - posx_i)^2 + (posy_{i+1} - posy_i)^2} \quad (3)$$

DISTANCE OF THE MOUSE TO THE STRAIGHT LINE (DMSL)

UNITS - pixels

This feature is similar to the previous one in the sense that it will compute the *s_dists* between two consecutive MOUSE_UP and MOUSE_DOWN events, *mup* and *mdo*, according to equation 2. However, it returns this sum rather than the average value during the path.

EXCESS OF DISTANCE (ED)

UNITS - pixels

This feature measures the excess of distance that the mouse travelled between each two consecutive MOUSE_UP and MOUSE_DOWN events. *r_dist* and *s_dist* are computed as for the AED feature. However, ED is given by $r_dist - s_dist$

MOUSE ACCELERATION (MA)

UNITS - pixels/milliseconds²

The velocity of the mouse (in pixels/milliseconds) over the time (in milliseconds). A value of acceleration is computed for each interval defined by two consecutive MOUSE_UP and MOUSE_DOWN events, using the intervals and data computed for the

Velocity.

MOUSE VELOCITY (MV)

UNITS - pixels/milliseconds

The distance travelled by the mouse (in pixels) over the time (in milliseconds). The velocity is computed for each interval defined by two consecutive MOUSE_UP and MOUSE_DOWN events. Let us assume two consecutive MOUSE_UP and MOUSE_DOWN events, mup and mdo , respectively in the coordinates (x_1, y_1) and (x_2, y_2) , that took place respectively in the instants $time_1$ and $time_2$. Let us also assume two vectors $posx$ and $posy$, of size n , holding the coordinates of the consecutive MOUSE_MOVE events between mup and mdo . The velocity between the two clicks is given by $r_dist / (time_2 - time_1)$, in which r_dist represents the distance travelled by the mouse and is given by equation 3.

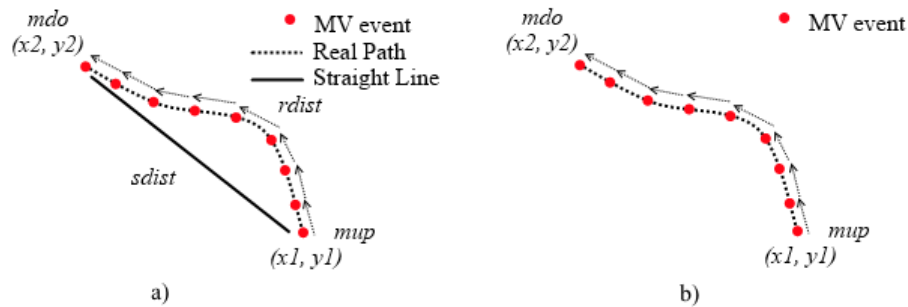


Figure 3. (a) A series of MOV events, between two consecutive clicks of the mouse. The difference between the shortest distance ($sdist$) and distance actually travelled by the mouse ($rdist$) is depicted. (b) The real distance travelled by the mouse between each two consecutive clicks is given by summing the distances between each two consecutive MOV events.

TIME BETWEEN CLICKS (TBC)

UNITS - milliseconds

The timespan between two consecutive MOUSE_UP and MOUSE_DOWN events, i.e., how long did it took the individual to perform another click.

3.2.2. Keyboard

The following features are extracted from the keyboard:

KEY DOWN TIME (KDT)

UNITS - milliseconds

The timespan between two consecutive KEY_DOWN and KEY_UP events, i.e., for how long a key is pressed while typing;

TIME BETWEEN KEYS (TBK)

UNITS - milliseconds

The timespan between two consecutive KEY_UP and KEY_DOWN events, i.e., how long does it take for the individual to press another key;

3.2.3. Attention

In what concerns attention, the key feature is a list of applications, which represent the sequence of applications that the user interacted with, and the duration of each interaction. To extract this feature, the server goes through a list of pairs (e.g. application name and timestamp) 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 application name, the time is added to the last registered application (since this means that the user is still interacting with it). An example of the output of this process is depicted in Figure 4.

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
Espaço 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

Figure 4. Sequence of applications used by a specific user, with the time in which the user 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 \leftarrow [];

$i \leftarrow 0$;

while $i < \text{Size}(p)$ **do**

 task $\leftarrow p_{i,1}$;

 time $\leftarrow p_{i,2}$;

$i++$;

while $i < \text{Length}(p)$ and $\text{StringLength}(p_{i,1}) = 0$ **do**

$i++$;

end

if $i = \text{Length}(p)$ **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 (Algorithm 2). To do this the server measures 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 administrator and is encoded in the form of regular expressions. The 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 and are not 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 5.

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 \leftarrow [];

i \leftarrow 0;

work \leftarrow 0;

others \leftarrow 0;

time \leftarrow st;

while i ; Size(t) **do**

if isWork($t_{i,1}$, set) **then**

 | work \leftarrow work + $t_{i,3}$;

else

 | others \leftarrow 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 \leftarrow 0;

 others \leftarrow 0;

 time $\leftarrow t_{i,2}$;

end

end

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








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

Figure 5. Detailed evolution of task-attention of a specific user.

4. Assessment of Performance in Stressful Environments

Modern life exerts a significant and constant pressure on individuals, driving them to a constant attempt to perform more and better. There are environments which constitute particularly "good" examples of this reality. The classroom is one of such environments in which individuals, from early in their lives, are confronted with frequent evaluations of their performance and the pressure that stems from its impact on their future and from the social judgment of their peers.

Higher education, in particular, is a period of the individual's educational path that is especially prone to result in added pressure. It is so because it constitutes a transition period before students reach the working environment, combining the fears and the pressure of both environments. This is corroborated by the high prevalence of anxiety disorders among higher education students [26].

The modern classroom constitutes thus one potentially well-suited environment for the use of this kind of environments. Specifically, we have been using it for the past years to study the behavior of students not only in regular classes but also and especially during exams. Doing so allows, on the one hand, to analyze the evolution of the students throughout their course. On the other hand, and perhaps more interestingly, it allows to determine how each individual student is affected by stress, potentially pointing out those students who are less able to effectively cope with stress. Identifying and acting on these students is a fundamental step to train future professionals who are able to perform under stress effectively.

As depicted in Figure 6, the proposed environment allows for example to analyze how the performance of the student varies during the exam. It is clearly visible that student #2 shows a marked improvement in performance during the exam, while the other two students show a smaller one. Identifying students whose performance drops in a stressful situation is made possible through this kind of analysis.

In order to assess the validity of these measures to predict stress, we have also measured stress response through salivary cortisol during the exam. Mouse Velocity and Mouse Acceleration are notably correlated to stress response, as depicted in Figure 7.

This kind of assessment allows not only to study how each individual is affected by stress but also to train models that can predict individual stress responses based on the interaction patterns with the computer. While the former is undoubtedly interesting, as it allows to identify those students who are less able to cope with stress, the latter leads to the very interesting perspective of classifying stress in real-time, from the analysis of Behavioral Biometrics. Both aspects are equally important in an academic organization that aims to create future professionals who have the adequate strategies to deal with the requirements of current work environments.

5. Assessment of Performance in Workplaces

In the last years, the proposed system has also been used to assess performance in the workplace. While in the previous case the assessment of stress is done in rather short periods of time, such as an exam, a workplace is an environment in which assessment is carried out over long periods of time, generally spanning the whole workday.

This extensive collection of behavioral data allowed some interesting insights on how performance evolves throughout the day, pointing out that performance tends to de-

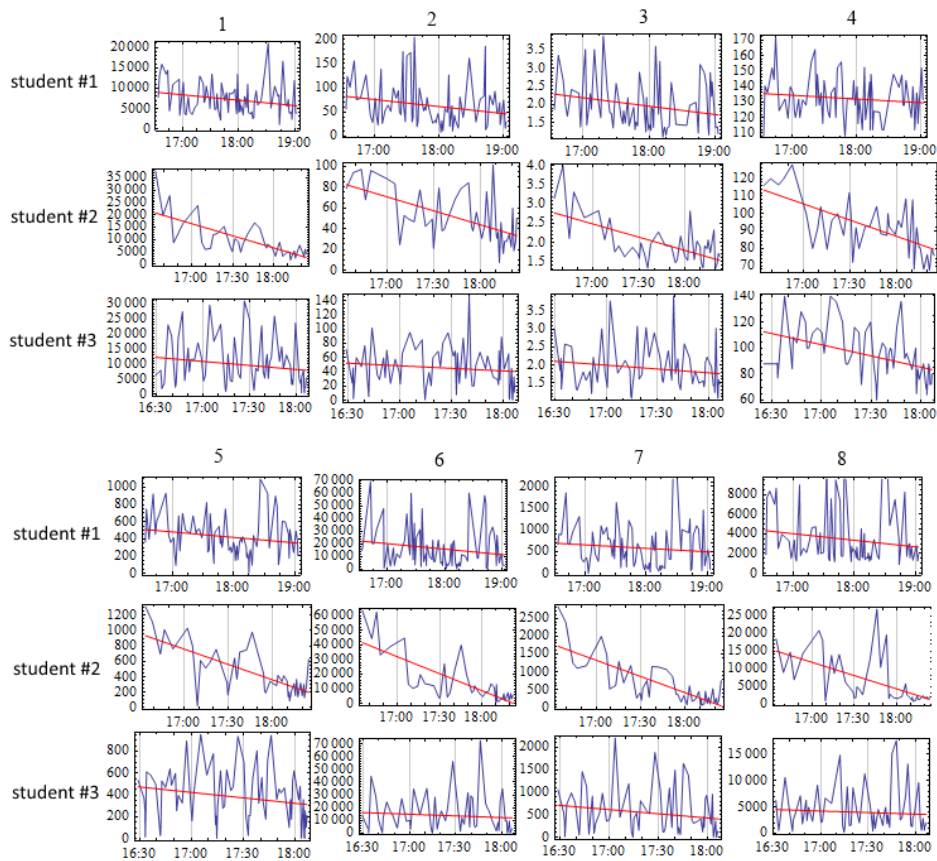


Figure 6. Time plot of the features for three arbitrary students. The negative correlation with time is visible for all features. Lines depict three different students. Columns depict the following eight features: 1 - ASA, 2 - ADMSL, 3 - AED, 4 - CD, 5 - DBC, 6 - DMSL, 7 - ED, 8 - TBC.

crease as the day progresses [27]. Figures 8 and 9 point out precisely this trend. They show the distribution of data for a group of individuals who share a common workplace. Data was selected in four moments (M1 - beginning of work day, M2 - around 11 am, M3 around 4 pm and M4 at the end of the workday). The trend of decreasing performance is evidenced by a slower mouse velocity (8 left), a higher time between keys (8 right) and an increased distance between clicks (9).

This trend is even more evident if we analyze data from the whole day. Figure 10 shows the evolution of the mouse velocity for the same team of individuals, computed through the moving average. It shows a decrease from around 0.9 pixels/milliseconds to around 0.6. This clearly shows that the use of the mouse becomes slower as the workday progresses.

Figure 11, on the other hand, shows how the key down time increases during the day, for the same group of individuals. It starts at around 87 milliseconds, at the beginning of the day, and increases to around 90 milliseconds at the end of the workday. It clearly points out how typing in a keyboard becomes slower as fatigue settles in.

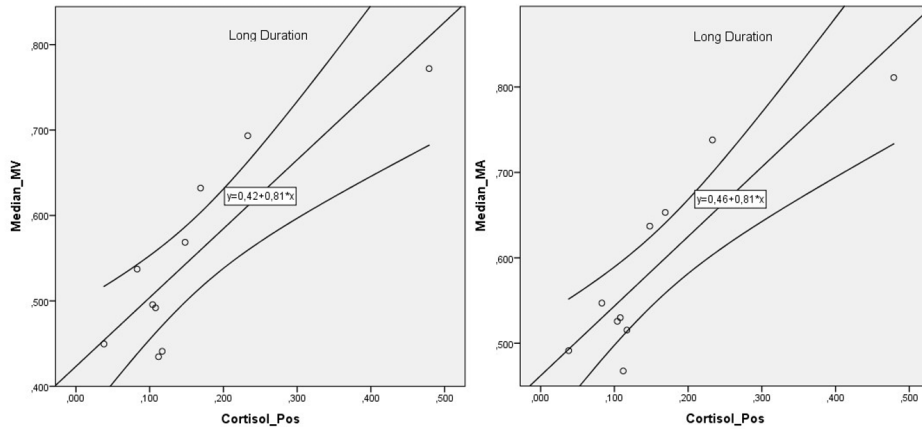


Figure 7. Correlation of salivary cortisol with mouse velocity (left, $R = 0.870$, $\rho = 0.001$) and mouse acceleration (right, $R = 0.886$, $\rho = 0.001$)

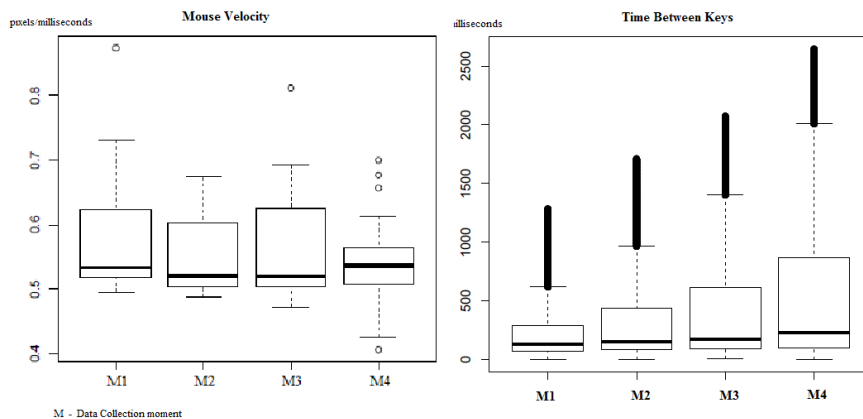


Figure 8. Distribution of the Mouse Velocity and Time Between Keys in four different moments of the day, for a group of individuals who share the same workplace.

In both cases the changes observed are rather small and would be impossible to detect through a simple observation. Nonetheless, they can be detected and quantified through these approaches, providing access to new forms of quantifying performance in the workplace.

There may be many advantages in doing this kind of analysis, for both organizations and workers. On the one hand, it allows for an organization to assess not only the performance of the team at each moment but also to identify particularly negative moments. On the other hand, it may allow to better know the workforce, namely in terms of optimum work rhythms. Altogether, this allows for a better management of the human workforce, with positive impact in indicators such as productivity, well-being, health or product quality.

In order to facilitate this kind of decision making, graphical tools such as the one

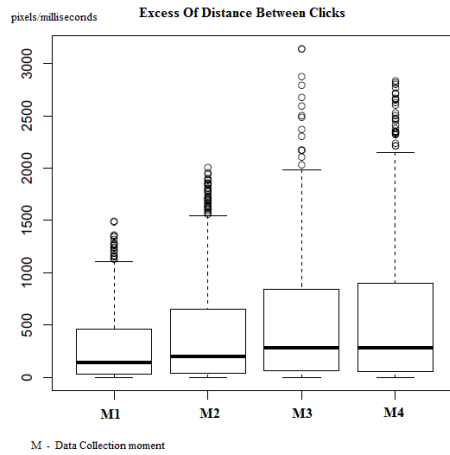


Figure 9. Distribution of the excess of distance in four different moments of the day, for a group of individuals who share the same workplace.

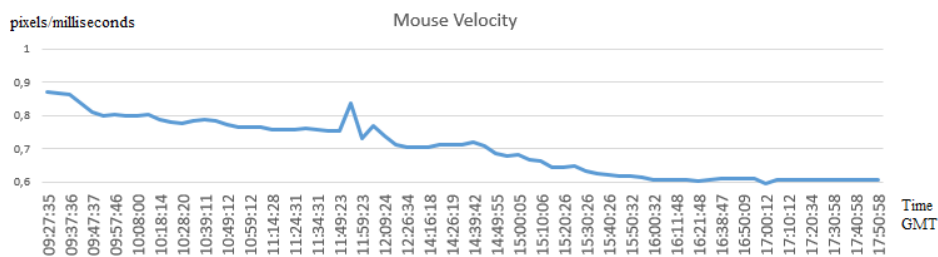


Figure 10. Evolution of mouse velocity during the workday for a team of employees.

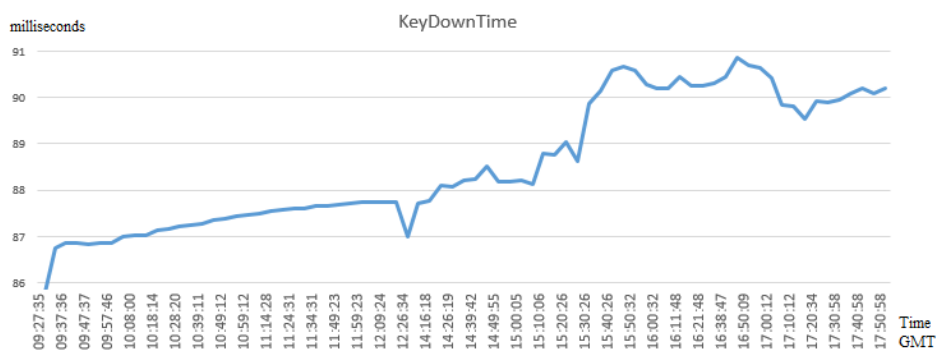


Figure 11. Evolution of the key down time during the workday for a team of employees.

depicted in Figure 12 can be developed that, with the use of real-time analytics provide feedback to team managers about the state of the workforce. Summaries of the data, temporal evolution plots and statistical measures provide the necessary abstractions for decision-makers to optimize the management o their human resources.

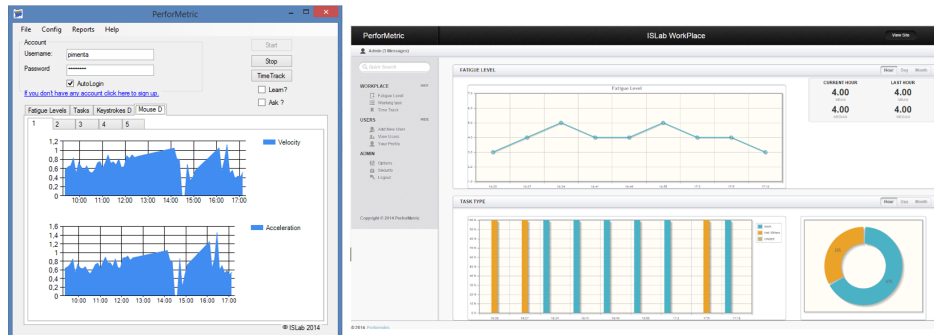


Figure 12. Graphical interfaces detailing the evolution of the performance of a specific worker (left) or of a team of workers (right).

6. Quantification of Attention to Task

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 [28]. 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 [29].

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 [30]. 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 [29,31].

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 [32].

The system presented in this chapter can be used to monitor attention in groups of people. 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 interacting with the computer and, together with knowledge about the task, is able to classify attention throughout time.

The proposed environment has been in use 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 Portuguese policies move towards the creation of 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 analyze 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 13 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

Figure 13. 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 attention for that specific student, in a given class. Figure 14 shows the evolution of attention for three specific students during the 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 classes. Figure 15 shows the global evolution of the attention of cohort 10N, in a regular class (a) and in an assessment class (b). This visual

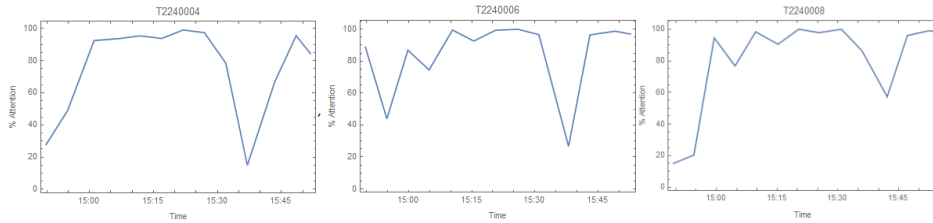


Figure 14. Temporal evolution of attention in three different students of the same class.

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 15 (c) shows the distribution of the values of attention in cohort 10N, in a regular class and in an assessment one.

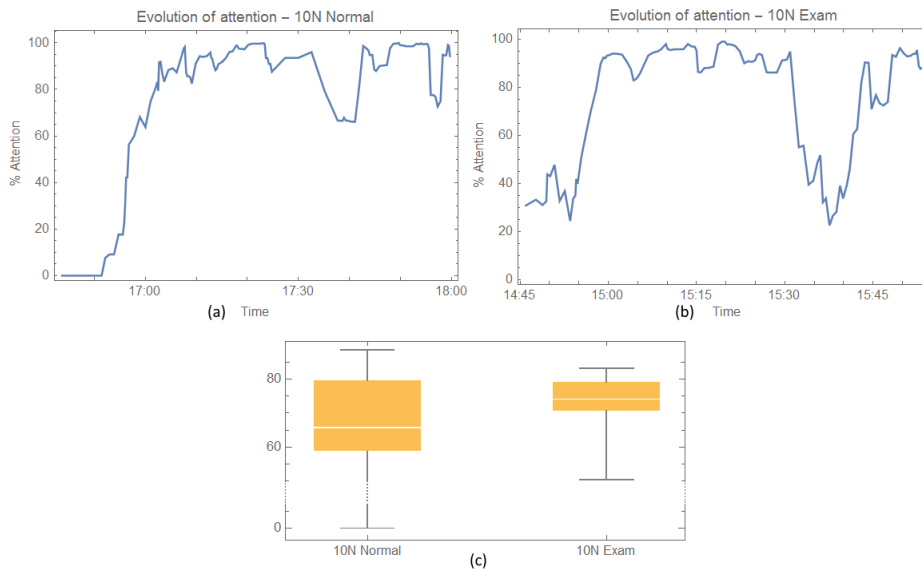


Figure 15. Temporal evolution of attention in the same cohort of students (10N) in two different situations: regular class (a) and assessment (b). Comparing the overall attention of both groups (c).

7. Discussion and Further Research Directions

In this chapter we described a technologically empowered environment for the acquisition of contextual information about the environment's users. This environment was designed with the aim to acquire information that could be used to characterize the state of the users, in a non-intrusive and transparent way.

Indeed, the understanding of the user's state is a fundamental problem in any AmI system. To address it, we focused on the acquisition of behavioral features. Specifically, we look at the users' interaction with the devices present in the environment. This leads

to the main shortcoming of the proposed solution: it is only effective in environments in which people spend a significant amount of time interacting with the computer. This includes many of nowadays workplaces and academic environments, in which the use of the computer is standard practice.

Nonetheless, for this specific type of environments, this solution has a group of very interesting characteristics:

- It does not require any additional hardware since it relies on the observation of the users' interaction with the mouse and keyboard, which are standard peripherals;
- It is completely non-intrusive and non-invasive, as opposed to traditional approaches which rely on questionnaires or physiological sensors;
- It can be used continuously, throughout the day, without limitations, providing access to a plethora of data both in real-time and in a historic perspective;
- New users can be easily added to the system by installing the client application in the data generating devices.

Most of these characteristics stem from the fact that an approach in line with the AmI view was followed. Hence, users are seen as the central component of the environment and do not need to consciously interact with the system for it to classify their state and provide valuable feedback.

There are also significant advantages from the point of view of team managers, which include teachers. Namely, the proposed system gathers valuable information from groups of people, processes and summarizes it and delivers it to team managers using graphical and intuitive representations. This information can thus be used by these professionals for significantly improving their management of human resources, namely:

- Better knowledge concerning individual and group working rhythms and dynamics, allowing an adjustment to individual differences (e.g. some individuals have better performance during the morning, others during the afternoon);
- Identification of potentially negative user states in individuals. This includes detection of stress, fatigue or distraction, which often spread to other individuals in the team, affecting the quality of the environment and of the work/teaching processes;
- Identification of events, tasks or moments associated with potentially negative user states, allowing for the improvement of processes;
- Improvement of the quality of the environment through the continued improvement of individual well-being, namely through the issuing of personalized notifications aimed at managing stress and fatigue (e.g. encouraging of stress-coping strategies when stress peaks are detected).

The potential usefulness of such environments for the acquisition of contextual information regarding its users is thus unquestionable, for organizations and individuals alike. This chapter described the process of data acquisition as well as three real scenarios of application. However, many more different applications can be supported by this kind of environments.

One interesting future direction is to acquire information regarding the emotional state of the users, which is fundamental in an AmI system. Although we have not yet addressed this subject, some authors have already determined that it is possible to identify emotional states using keystroke dynamics or mouse dynamics [33,12,15]. This means

that an environment such as the one detailed in this chapter could also be used for this purpose with minimum changes required, since the data that constitutes the input is the same and is collected in the same way. Other authors performed minor changes in the keyboard to include pressure sensors, and were also able to recognize emotions based on the pressure exerted on the keys while typing [34].

Another potential and interesting use of this environment is in the field of user identification and authentication. Indeed, another key aspect in AmI systems is the identification of each individual user. A significant amount of research supports the claim that keystroke dynamics can be used to effectively identify the user of a computer [35].

Finally, there is also still work to be carried out in what concerns privacy and data protection in these kind of environments. While user agreements may solve the issue in certain fields of application, there are others more sensitive due to the type of information collected that demand that specific measures be embedded in this kind of systems. While this important topic has been addressed in the past[36], it is outside the scope of the present chapter. Nonetheless, these are issues that must always be present in the development of this type of systems.

Many other potential applications for this kind of environments exist, such as the prediction of task load, cognition or demographic indicators such as mental age or gender [37].

Summarizing, it is our conviction that the combined use of Behavioral Biometrics and Ambient Intelligence represents a significant opportunity, especially for the latter field. Specifically, it may lead to the development of environments that are aware of their users and their users' state in several different dimensions (e.g. stress, fatigue, attention, mental indicators). More importantly, with approaches such as these, this relevant information may be acquired in a completely non-intrusive way, without any explicit interaction of the user.

One of the key aspects of AmI systems is to adapt to its users. Given that this can only be successfully achieved if relevant information regarding users' context and state is acquired, we believe that this kind of approaches will play a relevant role in the development of truly non-intrusive user-aware environments.

Acknowledgment

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

References

- [1] J. C. Augusto, V. Callaghan, D. Cook, A. Kameas, I. Satoh, Intelligent environments: a manifesto, *Human-Centric Computing and Information Sciences* 3 (1) (2013) 1–18.
- [2] C. Karyotis, F. Doctor, R. Iqbal, A. James, State of the Art on AI Applied to Ambient Intelligence, *Frontiers in Artificial Intelligence and Applications*, IOS Press, 2017, Ch. Affect Aware Ambient Intelligence: Current and Future Directions.
- [3] B. de Carolis, S. Ferilli, State of the Art on AI Applied to Ambient Intelligence, *Frontiers in Artificial Intelligence and Applications*, IOS Press, 2017, Ch. Learning Daily Routines in Smart Office Environments.

- [4] D. Preuveneers, A. Ramakrishnan, T. V. hamme, V. Rimmer, Y. Berbers, W. Joosen, State of the Art on AI Applied to Ambient Intelligence, *Frontiers in Artificial Intelligence and Applications*, IOS Press, 2017, Ch. A survey on applying machine learning techniques for behavioral awareness.
- [5] P. Novais, D. Carneiro, The role of non-intrusive approaches in the development of people-aware systems, *Progress in Artificial Intelligence* 5 (3) (2016) 215–220.
- [6] J. C. Augusto, Ambient intelligence: Opportunities and consequences of its use in smart classrooms, *Innovation in Teaching and Learning in Information and Computer Sciences* 8 (2) (2009) 53–63.
- [7] S. Elmalaki, L. Wanner, M. Srivastava, Caredroid: Adaptation framework for android context-aware applications, in: *Proceedings of the 21st Annual International Conference on Mobile Computing and Networking*, ACM, 2015, pp. 386–399.
- [8] D. Kramer, A. Kocurova, S. Oussena, T. Clark, P. Komisarczuk, An extensible, self contained, layered approach to context acquisition, in: *Proceedings of the Third International Workshop on Middleware for Pervasive Mobile and Embedded Computing*, ACM, 2011, p. 6.
- [9] D. Carneiro, R. Costa, P. Novais, J. Neves, J. Machado, J. Neves, Simulating and monitoring ambient assisted living, in: *Proceedings of the ESM*, 2008, pp. 175–182.
- [10] R. V. Yampolskiy, V. Govindaraju, Behavioural biometrics: a survey and classification, *International Journal of Biometrics* 1 (1) (2008) 81–113.
- [11] R. M. Bolle, J. Connell, S. Pankanti, N. K. Ratha, A. W. Senior, *Guide to biometrics*, Springer Science & Business Media, 2013.
- [12] P. Shukla, R. Solanki, Web based keystroke dynamics application for identifying emotional state, vol 2 (2011) 4489–4493.
- [13] W.-H. Tsui, P. Lee, T.-C. Hsiao, The effect of emotion on keystroke: an experimental study using facial feedback hypothesis, in: *Engineering in Medicine and Biology Society (EMBC), 2013 35th Annual International Conference of the IEEE*, IEEE, 2013, pp. 2870–2873.
- [14] M. B. Lewis, Exploring the positive and negative implications of facial feedback., *Emotion* 12 (4) (2012) 852.
- [15] A. Kolakowska, A review of emotion recognition methods based on keystroke dynamics and mouse movements, in: *Human System Interaction (HSI), 2013 The 6th International Conference on*, IEEE, 2013, pp. 548–555.
- [16] X. Jiang, A. K. Sengupta, Effect of music and induced mental load in word processing task, in: *Systems, Man, and Cybernetics (SMC), 2011 IEEE International Conference on*, IEEE, 2011, pp. 3261–3266.
- [17] C. Shen, Z. Cai, X. Guan, Y. Du, R. A. Maxion, User authentication through mouse dynamics, *Information Forensics and Security*, *IEEE Transactions on* 8 (1) (2013) 16–30.
- [18] C.-J. Tsai, T.-Y. Chang, Y.-J. Yang, M.-S. Wu, Y.-C. Li, An approach for user authentication on non-keyboard devices using mouse click characteristics and statistical-based classification, *International Journal of Innovative Computing, Information and Control* 8 (11) (2012) 7875–7886.
- [19] Z. Jorgensen, T. Yu, On mouse dynamics as a behavioral biometric for authentication, in: *Proceedings of the 6th ACM Symposium on Information, Computer and Communications Security*, ACM, 2011, pp. 476–482.
- [20] A. Pimenta, D. Carneiro, J. Neves, P. Novais, A neural network to classify fatigue from human–computer interaction, *Neurocomputing* 172 (2016) 413–426.
- [21] D. Carneiro, J. C. Castillo, P. Novais, A. Fernández-Caballero, J. Neves, Multimodal behavioral analysis for non-invasive stress detection, *Expert Systems with Applications* 39 (18) (2012) 13376–13389.
- [22] L. M. Vizer, L. Zhou, A. Sears, Automated stress detection using keystroke and linguistic features: An exploratory study, *International Journal of Human-Computer Studies* 67 (10) (2009) 870–886.
- [23] J. Andrén, P. Funk, A case-based approach using behavioural biometrics to determine a user’s stress level., in: *ICCBW Workshops, 2005*, pp. 9–17.
- [24] R. Bixler, S. D’Mello, Detecting boredom and engagement during writing with keystroke analysis, task appraisals, and stable traits, in: *Proceedings of the 2013 international conference on Intelligent user interfaces*, ACM, 2013, pp. 225–234.
- [25] D. Carneiro, S. Pinto, A. Pinheiro, P. Novais, An intelligent environment to assess auditory emotional recognition, in: *Innovations in Intelligent SysTems and Applications (INISTA), 2015 International Symposium on*, IEEE, 2015, pp. 1–7.
- [26] A. Lepp, J. E. Barkley, A. C. Karpinski, The relationship between cell phone use, academic performance, anxiety, and satisfaction with life in college students, *Computers in Human Behavior* 31 (2014) 343–350.
- [27] A. Pimenta, D. Carneiro, P. Novais, J. Neves, Monitoring mental fatigue through the analysis of keyboard

- and mouse interaction patterns, in: *Hybrid Artificial Intelligent Systems*, Springer, 2013, pp. 222–231.
- [28] W. K. Estes, *Handbook of Learning and Cognitive Processes (Volume 4): Attention and Memory*, Psychology Press, 2014.
 - [29] T. H. Davenport, J. C. Beck, *The attention economy: Understanding the new currency of business*, Harvard Business Press, 2013.
 - [30] D. L. McBride, Distraction of clinicians by smartphones in hospitals: a concept analysis, *Journal of advanced nursing* 71 (9) (2015) 2020–2030.
 - [31] J. Simola, J. Hyönä, J. Kuisma, Perception of visual advertising in different media: from attention to distraction, persuasion, preference and memory, *Frontiers Media SA*, 2015.
 - [32] J. Gottlieb, Attention, learning, and the value of information, *Neuron* 76 (2) (2012) 281–295.
 - [33] C. Epp, M. Lippold, R. L. Mandryk, Identifying emotional states using keystroke dynamics, in: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ACM, 2011, pp. 715–724.
 - [34] H.-R. Lv, Z.-L. Lin, W.-J. Yin, J. Dong, Emotion recognition based on pressure sensor keyboards, in: *Multimedia and Expo, 2008 IEEE International Conference on*, IEEE, 2008, pp. 1089–1092.
 - [35] S. P. Banerjee, D. L. Woodard, Biometric authentication and identification using keystroke dynamics: A survey, *Journal of Pattern Recognition Research* 7 (1) (2012) 116–139.
 - [36] Â. Costa, F. Andrade, P. Novais, Privacy and data protection towards elderly healthcare, in: *Handbook of Research on ICTs for Human-Centered Healthcare and Social Care Services*, IGI Global, 2013, pp. 330–346.
 - [37] D. G. Brizan, A. Goodkind, P. Koch, K. Balagani, V. V. Phoha, A. Rosenberg, Utilizing linguistically enhanced keystroke dynamics to predict typist cognition and demographics, *International Journal of Human-Computer Studies* 82 (2015) 57–68.