View metadata, citation and similar papers at core.ac.uk

brought to you by CORE

provided by Universidade do Minho: RepositoriUM

Supervising and Improving Attentiveness in Human Computer Interaction

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

[1]Department of Artificial Intelligence, Technical University of Madrid, Madrid, Spain
d.alves@alumnos.upm.es, jbajo@fi.upm.es
[2]Algoritmi Center, Minho University, Braga, Portugal
dcarneiro@di.uminho.pt, pjon@di.uminho.pt
[3]CIICESI, ESTGF, Polytechnic Institute of Porto, Portugal

Abstract. The collection, storage, management, and anticipation of contextual information about the user to support decision-making constitute some of the key operations in most Ambient Intelligent (AmI) systems. When the instructor has a computer-based class it is often difficult to confirm if the students are working in the proposed activities. In order to mitigate problems that might occur in an environment with learning technologies we suggest an AmI system aimed at capturing, measuring, and supervising the students' level of attentiveness in real scenarios and dynamically provide recommendations to the instructor. With this system it is possible to assess both individual and group attention, in real-time, providing a measure of the level of engagement of each student in the proposed activities and allowing the instructor to better steer teaching methodologies.

Keywords. Ambient Intelligence, Learning Styles, Innovative Scenarios, and Attentiveness.

Introduction

We live in a global and hyper connected world where technology is present in all spheres of life and is the backbone for the transformation of our society, which is in permanent change and requires continuous adaptation of the human being to the surrounding environment.

The development of new learning environments, supported by technology, may allow improving the learning process but it is also necessary to mitigate problems that may occur in an environment with learning technologies. Learning theories provide insights into the very complex processes and factors that influence learning and give precious information to be used in designing instruction that will produce optimum results. The learning models are designed in order to supply to the students with practice, evaluation and improvement procedures which will adjust the model [1].

The teaching process first requires that the instructor creates a pedagogical design of the objectives and determines the content to be taught. Second, a pre-assessment is used to determine learning abilities. Third, pedagogical procedures are used when teaching is initiated. Finally, assessment is applied to determine what learners have achieved, and, according to the assessment results, instructors should use feedback to determine the cause of ineffective instruction [1, 2].

However, for various reasons, students may not be predisposed to learning. Moreover, each student has its own particular way of assimilating knowledge, that is, his/her learning style. In this sense, and in bigger classes, it is important that the instructor has instruments to point out potential distractions (namely in what concerns the applications being used by the students) that may indicate a lack of predisposition to learning.

The concept of “big class” may however differ according to the context. In Europe, for example, a class with 100 students could be considered big class. However, in China, big classes might have 650 students. In classes with these dimensions, the instructor has difficulties in assessing the student’s commitment to the tasks during the class.

The goal of this paper is to propose an AmI system, directed at the instructor that indicates the level of attention of the students in the class when it requires the use of the computer. This AmI system captures, measures, and supervises the interaction of each student with the computer (or laptop) and indicates the level of engagement of students in the activities proposed by the instructor. When the instructor has big class, he/she can visualize in real time the level of engagement of the students in the proposed activities and act accordingly when necessary.

1. State of Art

There is no universally accepted definition of attention because there is a diversity of disciplines that study it. While in the past only psychologists studied attention, it is nowadays highly important for other fields like philosophy, chemistry, anatomy, and even computational science [3]. With the existence of these multiple sets of disciplines that study attention, its definition diverges depending on the field of study. For example, we can differentiate the concept of attention in human beings and machines. In humans, attention is processed in the brain, while in the machines there is a processor unit with a certain memory capacity that will process data. As with the brain, computers should analyze more and more data, but unlike the brain they do not or do rarely, “pay attention” [3] to the data. That is why an AmI system is proposed: not to replace the human being, but to complement it. Attention provides the brain with the capacity of selecting the main information and building priority tasks [3].

1.1. Attention Concept

As stated previously, the concept of attention diverges according to the field of study. In the psychological field, the first definition of attention was taking possession by the mind, in clear and vivid form, of one out of what seem several simultaneously possible objects or trains of thought. Its essence is the consciousness of focalization and concentration [4]. Most recently, the concept of attention is commonly used either to describe the active selection of information from the environment or the processing of information from internal sources [5, 6].

Attention also reduces information by transforming it into a meaningful structure. Attention means focusing one thought clearly among one of several subjects or objects that may occupy the mind simultaneously. In other words, attention implicitly means to cut things to deal effectively with other things [4].

1.2. Ambient Intelligence

The collection, storage, management, and anticipation of contextual information about the user to support decision-making constitute some of the key operations in most AmI systems.

An AmI system is an environment in which technology is embedded, hidden in the background, sensitive, adaptive and responsive to the presence of people and objects. This system also preserves security and privacy while using information when needed and with an appropriate context [7].

In the case of this work, adaptive systems aim at supporting and enhancing a student's learning process [8]. In their supply of adaptability, adaptive systems generally consider the student's knowledge, background, interest, goals, targets and/or choices [9].

2. Architecture

When students are affected by positive or negative states, they produce different kinds of thinking and this might hold important implications on the educational and training perspective. When this occurs it would be important to notify and advise the instructor, so he/she can be able to dynamically modify the teaching style according to students' feedback signals which include cognitive, emotional and motivational aspects.

While the student conscientiously interacts with the system and takes his/her decisions and actions, a parallel and transparent process takes place in which the AmI system uses the information. The architecture of the proposed AmI system, presented in Figure 1, depicts the process through which the system operates. It is possible to divide it into three major parts: data generating devices, cloud, and client.

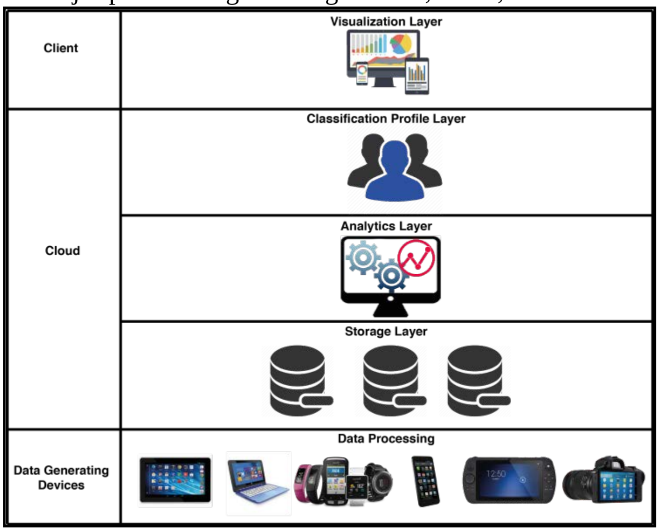

Figure 1. Architecture of the system.

The devices where students work have software that generates raw data. These devices store the raw data locally until it is synchronized with the web server in the cloud, which occurs at regular intervals (normally 5 minutes).

The cloud is composed of three layers: storage, analytics, and profile classification. In the storage layer the raw data received from the data generating devices is stored in a data store engine. The analytic layer provides powerful tools for performing analytics and statically analysis in real-time. This layer simplifies the code and limits resources requirements. It is important that in this process some values are filtered to eliminate possible negative effects on the analysis. The system calculates, at regular intervals, an estimation of the general level of performance and attention of each student.

The profile classification layer is where the indicators are interpreted. For example, interpreting data from the attentiveness indicators and build the meta-data, that will support decision-making. 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.

This layer has access to the current and historical state of the group from a global perspective, but it can also refer to each student individually, creating each student learning profile.

Profile classification is also a very important aspect to have control of since it allows carrying out analyses within longer time frames. This information will be used by another sub-module, the affective adaptive agent, to provide relevant information to the platform and to the mentioned personalization module.

Finally, the Client layer is developed as a web app with intuitive and visual representation (diagrams and other graphical tools) of the attentiveness states of the group and each student, abstracting from the complexity of the data level where they are positioned. At this point, the system can start to be used by the people involved, especially the instructor, who can better adapt and personalize teaching strategies. With a focus on individual and group performance and using real time analytics, the intuitive visual tools suggest and facilitate decision-making and student management. The actual quantification of the students' attention is displayed in the visualization layer, and can be used to personalize instruction according to the specific student, enabling the instructor to act differently with different students, and also to act differently with the same student, according to his/her past and present level of attention.

2.1. Data Acquision

The first stage in the life cycle of the proposed system takes place in the data generating devices, which was designed and implemented using a logger application developed in previous work [9]. The data collected by the logger application characterizing the students' interaction patterns is aggregated in a server to which the logger application connects after the student logs in.

The privacy of the students is ensured, since the necessary data that is collected in the registration process are an ID that does not identify the student, password, and gender. Furthermore, the privacy issues of the system are assured, since the instructor will only have access to the final results on the level of attention.

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. Table 1 describes the features extracted from the use of the mouse and the keyword.

These data are further processed, stored and then used to calculate the values of the behavioral biometrics. Mouse movements and keyboard usage patterns can also help predict the state of mind of the user [11, 12].

Table 1. Data acquisition features**.**

Symbol	Feature	Description
		Mouse Events
mv	Mouse Velocity	The distance travelled by the mouse (in pixels) over the time (in milliseconds).
ma	Mouse Aceleration	The velocity of the mouse (in pixels/milliseconds) over the time (in milliseconds).
cd	Click Duration	the timespan between MOUSE_UP events, whenever this timespan is inferior to 200 milliseconds.
tbc	Time Between Clicks	the timespan between two consecutive MOUSE_UP and MOUSE_DOWN events, i.e., how long did it took the individual to perform another click.
dbc	Distance Between Click	represents the total distance travelled by the mouse between two consecutive clicks, i.e., between each two consecutive MOUSE_UP and MOUSE_DOWN events.
ddc	Duration Distance Clicks	the time between consecutive MOUSE_UP and MOUSE_DOWN events.
edbc	Excess Distance Between Clicks	represents the excess total distance travelled by the mouse between two consecutive clicks, i.e., between each two consecutive MOUSE_UP and MOUSE_DOWN events.
aedbc	Absolute Excess Distance Between Click	this feature measures the average distance of the excess total distance travelled by the mouse between two consecutive clicks, i.e., between each two consecutive MOUSE_UP and MOUSE_DOWN events.
asdbc	Absolute Sum Distance Between Clicks	this feature measures the average sum of distance that the mouse travelled between each two consecutive MOUSE_UP and MOUSE_DOWN events.
dplbc	Distance Point to Line Between Clicks	this feature will compute the distance between two consecutive MOUSE_UP and MOUSE_DOWN events.
adpbc	Absolute Distance Point Between Clicks	this feature will compute the average distance between two consecutive MOUSE_UP and MOUSE_DOWN events.
		Keyword Events
kdt	Key Down Time	the timespan between two consecutive KEY_DOWN and KEY_UP events.
tbk	Time Between Keys	the timespan between two consecutive KEY_UP and KEY_DOWN events
kdtv	Key Down Time Velocity	The times that two consecutive keys are press

Asides from this information, which describe the interaction of the student with the computer, the system also registers the application usage by recording the timestamp in which each student switched to a specific application, by recording a tuple in the form (Id, Username, Timestamp, [AppName, Timestamp]).

Once information about the individual's attention exists in these terms, it is possible to start monitoring attentiveness in real-time and without the need for any explicit or conscious interaction. This makes this approach especially suited to be used in learning activities in which students use computers, as it requires no change in their working routines. This is the main advantage of this work, especially when compared to more traditional approaches that still rely on questionnaires (with issues concerning wording or question construction), special hardware (that has additional costs and is frequently intrusive) or the availability of human experts.

The Analytic layer detailed in Figure 1 is responsible for processing the data received from the storage layer in order to transform this data into the metrics presented.

It is important to state that in this process some values representing outliers are filtered to eliminate possible negative effects on the analysis (e.g. a key pressed for more than a certain amount of time). 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.

3. Results

The collected data were analyzed in two different ways. First, a general analysis is carried out with the aim of searching for group trends, i.e., behaviors common to a significantly large slice of students. Secondly, an individual analysis is made, in order to compare the different moments.

In the first step, the AmI system counts the number of interactions with the keyboard and the mouse in order to determine each data length student in each class. In this step it is possible to verify the mouse and keyboard activity of each student in the class which depends on the subjects that are applied in the class.

In the second step, the AmI system goes through the list of pairs and measures the time during which a window was active. Often there are cases in which the student doesn't change applications for a large extended period of time. In this case, which are represented by a pair with an empty AppName, the time is added to the last known AppName (since this means that the student is still interacting with it).

The level of attention of each student is quantified in step 3. However, in the beginning it is necessary that the instructor defines the task-related applications that the students will use during the class. The team administrator uses a graphical interface to set rules such as "starts with Google" or "Contains the word Code" which are then translated to regular expressions that are used by the algorithm to determine which applications are and are not work-related. In this sense it is necessary to measure the amount of time in each interval, that the student spent interacting with task-related applications. By default, applications that are not considered task-related are marked as "others" and count negatively towards the quantification of attention. When the student uses an application that does not match any of the known rules for a specific task, the application name is saved so that the instructor can later decide if a new rule should be created for it. The instructor may also determine the regular intervals at which attention is calculated.

Several interesting functionalities can be implemented with this approach, which provides valuable information to improve the instructor's decision-making process. Figure 2 shows the output of the evaluation of attention of a specific student, which allows the instructor to assess the temporal evolution of attention. These results consider the entire length of a class and give the percentage of time spent in task-related or other applications, for each student.

A system with these characteristics is a powerful tool in very different spheres, including administrative, academics or any environment in which people operate with computers.

In order to validate the proposed system, we have implemented it for the last past months in Caldas das Taipas High School, located in northern Portugal. In the Portuguese academic context, this system gains increased importance as current policies move towards the creations or larger classes, which make it increasingly difficult for the instructor to individually address each student. In this section we show tools supported by this system that, when at the disposal of the instructors, may allow to:

- Identify in real time oscillations in attention level, improving decision-making concerning aspects like breaks or when the student leaves the class;
- Decide in real time in which student to focus, according to their level of attention;
- Evaluate, a *posteriori* which contents are more likely for the students and which contents are predisposed to create distractions, providing a change for improvement.

Date	% Work	% Others
Mon 7 Dec 2015 08:58:06 GMT	87.4223	12.5777
Mon 7 Dec 2015 09:03:24 GMT	34.9715	65.0285
Mon 7 Dec 2015 09:16:05 GMT	73.1357	26.8643
Mon 7 Dec 2015 09:21:07 GMT	77.2261	22.7739
Mon 7 Dec 2015 09:26:38 GMT	93.2369	6.7631
Mon 7 Dec 2015 09:33:26 GMT	100.	0.
Mon 7 Dec 2015 09:43:10 GMT	100.	0.
Mon 7 Dec 2015 09:48:10 GMT	99.3287	0.671286
Mon 7 Dec 2015 09:53:23 GMT	100.	0.
Mon 7 Dec 2015 09:56:10 GMT	100.	0.

Figure 2. Detail of evaluation of attention for a specific student.

All these data collection processes will allow us to assess the influence on attention of aspects such as the time of the day, breaks, class contents, class objectives, among others.

As an example, we briefly analyzed the data collected for two different classes. In the first class a bells-letters class (12F) and in the second class a vocational class (12I). The subject of both classes was the same: a game programming application to teach algorithms. The conditions for both classes were the same and the task required the use of an application named “code.org” and Microsoft Word.

Figure 3 allows the instructor 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 instructor to perform a self-evaluation of how the class took place. The students of the belles-letters class have IDs between T2230001 and T2230014 while the students of the vocational class have IDs between T2210001 and T2210012.

Student	Task Duration	Work	Work %	Others	Others %
T2210001	50.0026 min	2347.41 s	78.2417	652.793 s	21.7583
T2210002	55.0297 min	2071.97 s	62.6977	1232.73 s	37.3023
T2210003	50.011 min	1564.75 s	52.1404	1436.28 s	47.8596
T2210004	55.0029 min	2709.18 s	82.0908	591.043 s	17.9092
T2210006	50.0025 min	1487.39 s	49.5761	1512.82 s	50.4239
T2210007	50.0023 min	22.1258 min	44.2481	1672.69 s	55.7519
T2210008	45.0024 min	2340.92 s	86.6918	359.359 s	13.3082
T2210009	55.0017 min	3260.52 s	98.7981	39.666 s	1.20193
T2210010	50.0011 min	47.176 min	94.3469	169.602 s	5.65309
T2210012	55.0025 min	1826.02 s	55.3309	1474.16 s	44.6691

Student	Task Duration	Work	Work %	Others	Others %
T2230001	69.978 min	2518.52 s	60.0207	1677.56 s	39.9793
T2230002	64.9661 min	2010.09 s	51.6181	1884.07 s	48.3819
T2230003	64.983 min	3340.72 s	85.7284	556.147 s	14.2716
T2230004	64.9835 min	3104.17 s	79.6538	792.909 s	20.3462
T2230011	59.9824 min	1170.88 s	32.5502	2426.28 s	67.4498
T2230013	59.9943 min	1325.93 s	36.8439	2272.85 s	63.1561
T2230014	59.9881 min	997.912 s	27.7368	2599.88 s	72.2632

Figure 3. The task duration and the amount of time that each student in the class spent interacting with the computer and the amount of actual work versus the amount spent interacting with other applications.

If necessary, the instructor may also click on a student in order to analyze the temporal evolution of his/hers attention. Figure 4 shows the evolution of attention for two specific students of each class described previously.

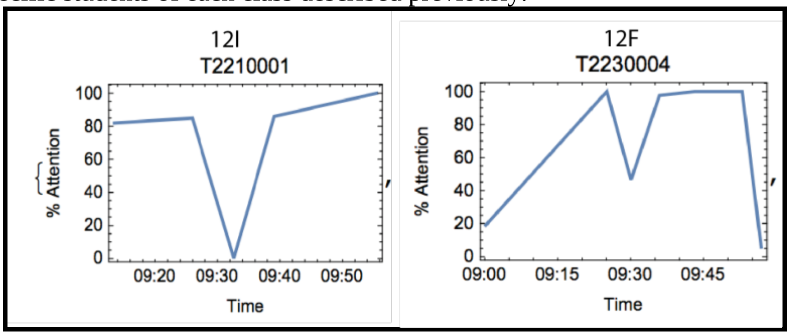

Figure 4. Temporal evolution of attention in two different students in different class.

The instructor may also find it very important to assess in real time or a posteriori, the evolution of attention of the whole class. In order to obtain this visual representation, the instructor may select which group of students to compare. Figure 5 shows the global evolution of attention in the vocational class (a) and bells-letters class (b). This graphical representation is built combining data from the students in each class and computing a running average.

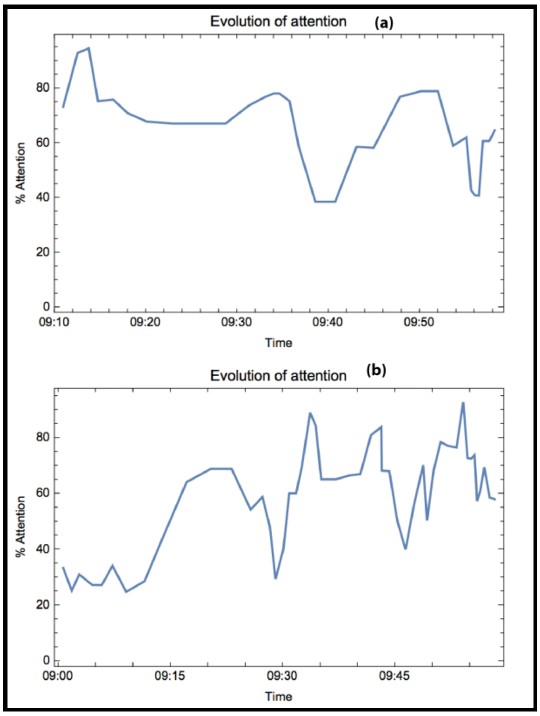

Figure 5. Temporal evolution in two class in the same situations. (a) vocational class 12I and (b) belles-letters class 12F.

4. Discussion and Future Work

The work developed so far resulted in a very useful system for the instructors to monitor, in real-time, the level of attention of their students. However, a limitation was also identified that happens whenever a student opens an application that is not task-related and does not interact with the computer anymore until the end of the task. In such a case, and according to the collected data, the student's attention during the class is quantified at 0%. Similarly, if the student opens a task-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. In order to address this issue, the system must not only consider the amount of time that each student devoted to the task and to other activities, but also the amount of interaction with the mouse and the keyword.

To address this limitation, in future work we will implement a tighter integration between the quantification of attention and the analysis of interaction patterns. These interaction patterns allow knowing all the actions that each user performed with the mouse or the keyboard, and at what time. We will thus define 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.

The door is thus open to allow a better analysis of the students' profile, taking into account their individual characteristics, and to propose new strategies and actions. By providing instructors with access to this information, we will allow them to better manage their interactions with the students, namely by pointing out the most problematic cases of inattention in real-time.

Acknowledgment

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

References

[1] P. Eggen, & D. P. Kauchak, “Educational psychology: classroom connections,” New York: Merrill, 1992.
[2] K. D. Hopkins, “Educational and psychological measurement and evaluation,” (8th ed.). Boston: Allyn and Bacon, 1998.
[3] M. Mancas, “Computational Attention – Toward Attentive Computers”. Press Universitaire de Louvain, 2007.
[4] W. James, “Principles of Psychology, Vol.1”, Dover Publications Inc., New York, 1890, pp. 255-260.
[5] B.A. Campbell, H. H-ayne, R. Richardson, “Attention and Information Processing in Infants and Adults. Perspectives from Human and Animal Research”. Psichology Press. New York and London, 1992.
[6] Durães, D., Carneiro. D, Bajo, J., Novais, P., “Using Computing Peripheral Devices To measure Attentiveness,” In Advanced in Intelligent System and Computing, Vol. 473, 2016, pp. 147-155.
[7] W. Weber, J. M. Rabaey, E. Aarts, Ambient Intelligence, Springer, 2005, pp. 1-2.
[8] W. B. Neto, F. Gauthier, S. M. Nassar, “An Adaptive e-Learning Model for the Semantic Web,” In International Workshop on Applications of Semantic Web technologies for E-Learning. Banff, Canada, 2005, pp. 63-64.
[9] F. A. Khan, S. Grap, E. R. Weippl, A. M. Tjoa, “Identifying and Incorporating Affective Sate and Learning Styles in Web-Based Learning Management Systems,” Interaction Design and Architecture (s) Journal – IXD&A, 2010, N. 9-10, pp. 85-103.
[10] A. Pimenta, D. Carneiro, P. Novais, J. Neves, “Neural Network to Classify Fatigue from Human-Computer Interaction,” Neurocomputing, Elsevier, ISSN: 0925-2312, Volume 172, Pages 413–426, 2016, http://dx.doi.org/10.1016/j.neucom.2015.03.105.
[11] D. Carneiro, P. Novais, J. Miguel Pêgo, N. Sousa, J. Neves, “Using Mouse Dynamics to Assess Stress During Online Exams,” Hybrid Artificial Intelligent Systems, Henrique Onieva et al. (eds), Springer - Lecture Notes in Computer Science, Vol. 9121, ISBN 978-3-319-19643-5, ISSN 0302-9743, pp 345-356, 2015, http://dx.doi.org/10.1007/978-3-319-19644-2_29.
[12] M. Gomes, T. Oliveira, D. Carneiro, P. Novais, and J. Neves, “Studying the Effects of Stress on Negotiation Behaviour, Cybernetics and Systems,” Taylor & Francis Ltd, , ISSN: 0196-9722, Volume 45 issue 3, pp 279-291, 2014 , http://dx.doi.org/10.1080/01969722.2014.894858.