

ENHANCING LEARNING THROUGH PEDAGOGICAL DRIVEN RESPONSE TO INFERRED AFFECT

Field of the Invention

This invention relates to processes for early detection of frustration and providing certain actions to alleviate frustration.

Prior Art

A major challenge with the design of affect sensitive tutoring systems involves the development of computational systems to reliably detect the learner's emotions. Prior studies on the modeling of frustration and other affective states have focused on physical manifestations of the subject e.g. facial expressions, eye gaze, posture and physiological signals. Most of these sensors though yield encouraging recognition results, suffer from various issues and constraints when adapted and deployed in a naturalistic learning environment. For example, eye gaze and facial cameras suffer from occlusion and lighting issues while physiological sensors may be overly obtrusive, cumbersome to setup and rarely available for most educational contexts.

As such, those skilled in the art are constantly striving to provide analytics tools that provide a more accurate result of sensing and detecting of the frustration and engagement of students.

Summary of the Invention

The above and other problems are solved and an advance in the art is made by a system and method in accordance with this disclosure. A first advantage of a system and method is that the system is able to determine frustration of a student based on the affect state of a person. This is different from conventional method that is based on historical data points of each person. In other words, the system and method according to this disclosure can be used to monitor any user immediately without having to obtain substantial data points of a person. A second advantage of a system and method is that the system can be

executed in the background of any current educational applications. Hence, this system and method can be easily implemented to existing software applications.

A first aspect of the disclosure describes an affect-sensitive monitoring system for determining frustration of a student. The system comprises a processing unit and a memory; an image capturing device communicatively connectable to the processing unit; a keyboard communicatively connectable to the processing unit; a mouse communicatively connectable to the processing unit; instructions stored on the memory; in response to receiving a selected software application from a user, the instructions are executable by the processing unit to:

- receiving sequence of images from the image capturing device;
- receiving a mouse timing log file;
- receiving a keyboard timing log file;
- determining aggregated keyboard actions based on the keyboard timing log file;
- determining aggregated mouse actions based on the mouse timing log file;
- determining aggregated image actions based on the sequence of images;
- arranging each of the aggregated keyboard actions, mouse actions and image actions according to at least one selected machine learning based classifier;
- applying a decision template fusion algorithm on the arranged aggregated keyboard actions, mouse actions and image actions to obtain a final classification that indicates affect state of the user.

According to an embodiment of the first aspect of the disclosure, the instruction to determining aggregated keyboard actions based on the keyboard timing log file comprises: retrieving a baseline latency associated to the user; determining a latency for each key with the following expression: $\text{latency of each key} = \text{flight time} + \text{dwell time}$, where flight time is time duration from releasing a previous key to pressing the current key and dwell time is time duration that the current key is pressed; normalising the latency of each key based on the

retrieved baseline latency; and aggregating the normalised keys into a sliding window of a predetermined size to obtain the aggregated keyboard actions.

According to an embodiment of the first aspect of the disclosure, the instruction to aggregating the normalised keys into a sliding window of a predetermined size to obtain the aggregated keyboard actions comprises: aggregating into sliding window sizes of 30 seconds, with an overlap of 66.7% of the window size. According to another embodiment of this embodiment, the instructions further comprises: forwarding the aggregated keyboard actions for the past 30 seconds window to the classifier at 10 seconds interval.

According to an embodiment of the first aspect of the disclosure, the instruction to retrieving a baseline associated to the user comprises: obtaining a new baseline in response to absence of a baseline associated to the user.

According to an embodiment of the first aspect of the disclosure, the instruction to obtaining a new baseline comprises: requesting the user to perform typing of a random paragraph; and determining the baseline latency based on a keystroke data obtained for the random paragraph performed by the user, the keystroke data comprises latency for each key which is the sum of flight time and dwell time and the key type.

According to an embodiment of the first aspect of the disclosure, the instruction to determining the baseline latency based on a keystroke data obtained for the random paragraph performed by the user comprises: eliminating all keys with duration greater than or equals to 1 second from the keystroke data; eliminating all keys with key type associated to navigation, backspace and delete from the keystroke data; selecting 30 consecutive keys from the remaining keystroke data; forming a window for the first 10 keys of the selected 30 consecutive keys, calculating the mean of the keys duration and store the calculated mean in an array; forming a next window by moving the window forward by 2 keys, calculate the

mean of the keys duration and append the array to include the calculated mean for the next window; repeating the step of forming the next window until eleven windows are formed; and determining baseline latency with a median of mean duration values in the array.

According to an embodiment of the first aspect of the disclosure, the instruction to determining aggregated mouse actions based on the mouse timing log file comprises: recording the exact timestamp of each mouse click; determining the type of mouse actions wherein the type of mouse actions includes single click and double click; aggregating the determine mouse actions into a sliding window of a predetermined size to obtain the aggregated mouse actions.

According to an embodiment of the first aspect of the disclosure, the instruction to aggregating the determine mouse actions into a sliding window of a predetermined size to obtain the aggregated mouse actions comprises: aggregating into sliding window sizes of 30 seconds, with an overlap of 66.7% of the window size wherein each sliding window contains the determined single click and double clicks. According to another embodiment of this embodiment, the instructions further comprises: forwarding the aggregated mouse actions for the past 30 seconds window to the classifier at 10 seconds interval.

According to an embodiment of the first aspect of the disclosure, the machine learning based classifier comprises random forest, naïve Bayes, or multinomial logistic regression.

According to an embodiment of the first aspect of the disclosure, the instruction to applying a decision template fusion algorithm on the arranged aggregated keyboard actions, mouse actions and image actions to obtain a final classification that indicates affect state of a subject comprises:

a decision profile, DP, that receives the output of the classifier and re-arranges the output accordingly in the following expression,

$$DP(X) = \begin{bmatrix} d_{1,1}(X) & d_{1,j}(X) & d_{1,M}(X) \\ d_{i,1}(X) & d_{i,j}(X) & d_{i,M}(X) \\ d_{L,1}(X) & d_{L,j}(X) & d_{L,M}(X) \end{bmatrix}$$

$d_{i,j}$ denotes the probability or measure of confidence that classifier i considers the input or feature vector X to be of class j ;

a decision template, DT, that is an $L \times M$ matrix where L is the number of classifiers, M is the number of classes and $(k, s)^{th}$ element, $DT_i(k, s)(Z)$, is computed with the following expression,

$$dt_i(k, s)(Z) = \frac{\sum_{j=1}^N Ind(z_j, i) d_{k,s}(z_j)}{\sum_{j=1}^N Ind(z_j, i)}$$

where $k = 1 \dots L$, $s = 1 \dots M$ and $Z = \{z_1, \dots, z_n\}$ is the training data set, and DT_i for class i is the average of the DP of the elements of the training set Z labelled in class i ;

determining the final classification, μ_d^i , with the following expression,

$$\mu_d^i = 1 - \frac{1}{LM} \sum_{k=1}^L \sum_{s=1}^M (DT_i(k, s) - DP_i(X))^2$$

wherein the i value that gives the maximum similarity measure μ_d^i would be the final classification.

A second aspect of the disclosure describes an affect-sensitive monitoring method for determining frustration of a student. The method comprises in response to receiving a selected software application from a user: receiving sequence of images from the image capturing device; receiving a mouse timing log file; receiving a keyboard timing log file; determining aggregated keyboard actions based on the keyboard timing log file; determining aggregated mouse actions based on the mouse timing log file; determining aggregated image actions based on the sequence of images; arranging each of the aggregated keyboard actions, mouse actions and image actions according to at least one selected

machine learning based classifier; applying a decision template fusion algorithm on the arranged aggregated keyboard actions, mouse actions and image actions to obtain a final classification that indicates affect state of a subject.

According to an embodiment of the second aspect of the disclosure, the step of determining aggregated keyboard actions based on the keyboard timing log file comprises: retrieving a baseline latency associated to the user; determining a latency for each key with the following expression: latency of each key = flight time + dwell time, where flight time is time duration from releasing a previous key to pressing the current key and dwell time is time duration that the current key is pressed; normalising the latency of each key based on the retrieved baseline latency; and aggregating the normalised keys into a sliding window of a predetermined size to obtain the aggregated keyboard actions.

According to an embodiment of the second aspect of the disclosure the step of aggregating the normalised keys into a sliding window of a predetermined size to obtain the aggregated keyboard actions comprises: aggregating into sliding window sizes of 30 seconds, with an overlap of 66.7% of the window size. According to another embodiment of this embodiment, the method further comprises: forwarding the aggregated keyboard actions for the past 30 seconds window to the classifier at 10 seconds interval.

According to an embodiment of the second aspect of the disclosure, the step of retrieving a baseline associated to the user comprises: obtaining a new baseline in response to absence of a baseline associated to the user.

According to an embodiment of the second aspect of the disclosure, the method of obtaining a new baseline comprises: requesting the user to perform typing of a random paragraph; determining the baseline latency based on a keystroke data obtained for the

random paragraph performed by the user, the keystroke data comprises latency for each key which is the sum of flight time and dwell time and the key type.

According to an embodiment of the second aspect of the disclosure, the method of determining the baseline latency based on a keystroke data obtained for the random paragraph performed by the user comprises: eliminating all keys with duration greater than or equals to 1 second from the keystroke data; eliminating all keys with key type associated to navigation, backspace and delete from the keystroke data; selecting 30 consecutive keys from the remaining keystroke data; forming a window for the first 10 keys of the selected 30 consecutive keys, calculating the mean of the keys duration and store the calculated mean in an array; forming a next window by moving the window forward by 2 keys, calculate the mean of the keys duration and append the array to include the calculated mean for the next window; repeating the step of forming the next window until eleven windows are formed; and determining baseline latency with a median of mean duration values in the array.

According to an embodiment of the second aspect of the disclosure, the step of determining aggregated mouse actions based on the mouse timing log file comprises: recording the exact timestamp of each mouse click; determining the type of mouse actions wherein the type of mouse actions includes single click and double click; aggregating the determine mouse actions into a sliding window of a predetermined size to obtain the aggregated mouse actions.

According to an embodiment of the second aspect of the disclosure, the step of aggregating the determine mouse actions into a sliding window of a predetermined size to obtain the aggregated mouse actions comprises: aggregating into sliding window sizes of 30 seconds, with an overlap of 66.7% of the window size wherein each sliding window contains the determined single click and double clicks. According to another embodiment of this

embodiment, the method further comprises forwarding the aggregated mouse actions for the past 30 seconds window to the classifier at 10 seconds interval.

According to an embodiment of the second aspect of the disclosure, the machine learning based classifier comprises random forest, naïve Bayes, or multinomial logistic regression.

According to an embodiment of the second aspect of the disclosure, the step of applying a decision template fusion algorithm on the arranged aggregated keyboard actions, mouse actions and image actions to obtain a final classification that indicates affect state of a subject comprises:

from a decision profile, DP, receiving the output of the classifier and re-arranging the output accordingly in the following expression,

$$DP(X) = \begin{bmatrix} d_{1,1}(X) & d_{1,j}(X) & d_{1,M}(X) \\ d_{i,1}(X) & d_{i,j}(X) & d_{i,M}(X) \\ d_{L,1}(X) & d_{L,j}(X) & d_{L,M}(X) \end{bmatrix}$$

$d_{i,j}$ denotes the probability or measure of confidence that classifier i considers the input or feature vector X to be of class j ;

from a decision template, DT, that is an $L \times M$ matrix where L is the number of classifiers, M is the number of classes and $(k, s)^{th}$ element, $DT_i(k, s)(Z)$, computing with the following expression,

$$dt_i(k, s)(Z) = \frac{\sum_{j=1}^N Ind(z_j, i) d_{k,s}(z_j)}{\sum_{j=1}^N Ind(z_j, i)}$$

where $k = 1 \dots L$, $s = 1 \dots M$ and $Z = \{z_1, \dots, z_n\}$ is the training data set, and DT_i for class i is the average of the DP of the elements of the training set Z labelled in class i .

determining the final classification, μ_d^i , with the following expression,

$$\mu_d^i = 1 - \frac{1}{LM} \sum_{k=1}^L \sum_{s=1}^M (DT_i(k, s) - DP_i(X))^2$$

wherein the i value that gives the maximum similarity measure μ_d^i would be the final classification.

A third aspect of the disclosure describes an educational application. The educational application comprises: a login interface for receiving login details from users; a plurality of lesson plans; a dashboard in response to a login from a user with a tutor privilege, the dashboard displaying a heat map to identify users who require assistance and scores for each student; and the affect-sensitive monitoring method, as described above in relation to the second aspect, executed in response to successful login from user.

Brief Description of the Drawings

The above and other features and advantages of a method and/or system in accordance with this invention are described in the following Detailed Description and are shown in the following drawings:

Figure 1 illustrates an affect-sensitive monitoring system to provide or perform a system and/or method in accordance with embodiments of this disclosure;

Figure 2 illustrates a processing system of a server and/or computing unit performing processes to provide a system in accordance with this disclosure;

Figure 3 illustrates a process flow of information in accordance with this disclosure;

Figure 4 illustrating a process performed by a keyboard processing module in accordance with this disclosure;

Figure 5 illustrating two sliding windows in accordance with this disclosure;

Figure 6 illustrating a process performed by a mouse processing module in accordance with this disclosure;

Figure 7 illustrating a dashboard in accordance with this disclosure;

Figure 8 illustrating a process of a lesson plan implementing the affect-sensitive monitoring system in accordance with this disclosure; and

Figure 9 illustrating an example of a selected lesson plan in accordance with an embodiment of this disclosure.

Detailed Description

This invention relates to processes for early detection of frustration and providing certain actions to alleviate frustration.

It is envisioned that a system and/or method in accordance with embodiments of this invention may be used for sensing of student or novice programmer's frustration and engagement through keystrokes, mouse clicks, head postures and facial dynamics captured using an array of unobtrusive and ubiquitous sensors and scaffolding of their learning through generation of contextual sensitive hints and affective prompts.

The system and method according to this disclosure use an array of unobtrusive and ubiquitous sensors to capture keystrokes, mouse clicks, head postures and facial dynamics for the inferring and sensing of novice programmer's frustration and engagement. This can potentially be adapted for use in online tutoring systems where students' frustration and engagement are monitored on a granular basis for adaptation of tutoring strategies to optimize the students' learning. This technique of extracting and analyzing keystroke dynamics using machine learning algorithms has not been previously developed for frustration and engagement detection in a tutoring context.

With the ability to automatically sense the frustration and engagement of students, the affect-sensitive tutoring system can emulate the personalized instruction that a human teacher can provide, thereby optimizing the learning motivation level of students and enhancing the learning outcomes. Hints, prompts and tutoring supports can be provided when the affect-tutoring system detects that students are frustrated or disengaged. Valuable insights on the learning areas or topics where the students are facing difficulties and frustration with can also be gleaned from the recorded data. A dashboard with the layout of the lab will show a heat map to direct the tutor to the students who require more help. The dashboard will also display the scores obtained by the students and the topics which students are weak in. The system and method according to this disclosure can potentially be deployed in educational, training institutions and Massive Open Online Course (MOOC) providers.

It should also be useful to note that the system and method in accordance with this disclosure is different from systems and methods that are implemented on the basis of "behavioural profiles" which require collection of data points of individuals over a prolonged period to build up the behavioural profile of each individual. This is quite different from the

system and method according to this disclosure directed to "affect or emotion" of individuals that captures more complex measurements such as "keystroke latency", "flight time" for individual keys, "pause duration", "backspace key latency", "delete key latency", "non-navigation key latency" etc. The measures that are collected are then passed through a machine learning algorithm to learn on a granular basis (dependent on the size of each sliding window) on the affect experienced by the student.

Figure 1 illustrates an affect-sensitive monitoring system 100 that includes a server 190 and a computing unit 180 for performing the processes of an embodiment of this disclosure.

Computing unit 180 is typical processing system such as a desktop computer, laptop computer, or other computer terminal. Computing system 180 is communicatively connected to server 190 via either a wired or wireless connection to transmit to and receive data from the server 190. One skilled in the art will recognise that more than one computing unit 180 would be required and the exact number of computing unit 180 would be dependent on the number of students in a class. Computing system 180 includes a processing unit 110, camera 120, keyboard 130, mouse 140 and display 150. Camera 120, keyboard 130, mouse 140 and display 150 are communicatively connected to the processing unit 110. Further details of the computing unit 180 will be described below with reference to figure 2.

Server 190 is a typical processing system such as a desktop computer, laptop computer, or other computer terminal. Server 190 is communicatively connected to all the computing unit 180 either a wired or wireless connection. One skilled in the art will recognise that more than one server 190 may be used without departing from the invention. Further details of the server 190 will be described below with reference to figure 2.

Processes stored as instructions in a media that are executed by a processing system in processing unit 110 and server 190 provides the method and/or system in accordance with this invention. The instructions may be stored as firmware, hardware, or software. Figure 2 illustrates processing system 200 such as the processing system in processing unit 110 and server 190 that execute the instructions to perform the processes

for providing a method and/or system in accordance with this invention. One skilled in the art will recognize that the exact configuration of each processing system may be different and the exact configuration of the processing system in each device may vary. Thus, processing system 200 shown in Figure 2 is given by way of example only.

Processing system 200 includes Central Processing Unit (CPU) 205. CPU 205 is a processor, microprocessor, or any combination of processors and microprocessors that execute instructions to perform the processes in accordance with the present invention. CPU 205 connects to memory bus 210 and Input/ Output (I/O) bus 215. Memory bus 210 connects CPU 205 to memories 220 and 225 to transmit data and instructions between the memories and CPU 205. I/O bus 215 connects CPU 205 to peripheral devices to transmit data between CPU 205 and the peripheral devices. One skilled in the art will recognize that I/O bus 215 and memory bus 210 may be combined into one bus or subdivided into many other busses and the exact configuration is left to those skilled in the art.

A non-volatile memory 220, such as a Read Only Memory (ROM), is connected to memory bus 210. Non-volatile memory 220 stores instructions and data needed to operate various sub-systems of processing system 200 and to boot the system at start-up. One skilled in the art will recognize that any number of types of memory may be used to perform this function.

A volatile memory 225, such as Random Access Memory (RAM), is also connected to memory bus 210. Volatile memory 225 stores the instructions and data needed by CPU 205 to perform software instructions for processes such as the processes required for providing a system in accordance with this invention. One skilled in the art will recognize that any number of types of memory may be used as volatile memory and the exact type used is left as a design choice to those skilled in the art.

I/O device 230, keyboard 235 (or 130), display 240 (or 150), memory 245, network device 250 and any number of other peripheral devices connect to I/O bus 215 to exchange data with CPU 205 for use in applications being executed by CPU 205. I/O device 230 is any device that transmits and/or receives data from CPU 205. Keyboard 235 (or 130) is a

specific type of I/O that receives user input and transmits the input to CPU 205. Display 240 (or 150) receives display data from CPU 205 and display images on a screen for a user to see. Memory 245 is a device that transmits and receives data to and from CPU 205 for storing data to a media. Network device 250 connects CPU 205 to a network for transmission of data to and from other processing systems.

Some I/O device 230 includes camera 120 and mouse 140. Camera 120 is an image capturing device that is capable of capturing still and/or moving images such as complementary metal-oxide semiconductor (CMOS) or charge-coupled sensor (CCD) type cameras. Mouse 140 is a pointing device that translates movement of the mouse to motion of a pointer on the display 240 (or 150). Mouse 140 consists of at least one button to receive mouse click.

Figure 3 illustrates the overview of the process flow 300 of information when the affect-sensitive monitoring application is executed by the affect-sensitive monitoring system 100 in accordance with an embodiment of this disclosure. It is noted that the affect-sensitive monitoring application would be running concurrently (as a background application) with a selected software application initiated by the user. For example, if the affect-sensitive monitoring application is pre-programmed to be executed when a user opens a software application such as "Microsoft Word", the affect-sensitive monitoring application would be triggered to running concurrently when the user opens "Microsoft Word". Specifically, the affect-sensitive monitoring application would be running in the background to monitor the affect state of the user when the user is using certain software applications such as "Microsoft Office", or educational software applications provided by educational, training institutions and Massive Open Online Course (MOOC) providers. In short, the affect-sensitive monitoring application is executed in response to receiving a selected software application from a user. The process flow 300 would now be described as follows.

First, data from the camera, mouse and keyboard are fed to respective processing modules to extract relevant data. In this instance, sequence of images captured by the camera 120 is fed to the image processing module 321; mouse timing logs are retrieved from the processing unit 110 and fed to the mouse data processing module 322; and

keyboard timing logs are retrieved from the processing unit 110 and fed to the keyboard data processing module 333. In other words, the affect-sensitive monitoring application would receive the sequence of images from the camera, mouse timing logs and keyboard timing logs from the processing unit 110. It should be noted that mouse timing logs and keyboard timing logs are stored on the memory of the processing unit 110. Hence, the affect-sensitive monitoring application does not receive the mouse timing logs and keyboard timing logs from the mouse and keyboard respectively. The mouse timing logs and keyboard timing logs are stored on certain designated parts of the memory of the processing unit 110 and the exact location of the mouse timing logs and keyboard timing logs would be dependent on the configuration of the processing unit 110.

In the image processing module 321, the facial features are computed from the Action Units (AUs) readings that are extracted from the sequence of images received from the camera. The head position and rotational angles are extracted from the sequence of images received from the camera. The method of extracting head position and rotational angles by the AUs are derivable by third parties applications and hence the exact method of extracting head position and rotational angles by the AUs is omitted.

In the mouse data processing mouse 322, the mouse clicks, double clicks and scroll latency are extracted from the mouse timing logs. Certain processes are executed by the mouse data processing module 322 to extract relevant information according to an embodiment of this disclosure. Further details performed by the mouse data processing module 322 will be described below with reference to figure 6.

In the keyboard data processing module 323, the dwell and flight time are computed from the keystroke timing logs. The keystroke features are normalized with the calculated baseline flight time for each user. Further details performed by the keyboard data processing module 323 will be described below with reference to figure 4.

Beside keystrokes and mouse clicks, the process may further include monitoring of certain actions performed by a user. In the context of a use of the system and method of this disclosure in conjunction with a specific lesson pertaining to software programming, the user

is required to perform certain actions that are relevant to the lesson plan. For example, and with reference to figure 9, the software lesson plan may include a working code with missing parts 920 for students to fill in, a compile option 940 (for student to select after filling in all the missing codes), a check answer code 950 (for student to determine whether certain parts of the codes entered are correct), and an output prompt box 930 (to display the errors in the codes filled in by the student after the student selects the compile option). The actions such as duration of entering the codes, number of times the student selects compile option, number of time the student selects the check answer option, and the number of errors reflected in the output prompt box are relevant to determine the mood of the student. Specifically, whether or not the student is frustrated can be determined with the actions of the student. Hence, such actions 314 are also monitored and injected to the classifier 330. These actions may be injected to the classifier as part of the output of the mouse data processing 322 or the keyboard data processing 333 without departing from the disclosure. Alternatively, an additional classifier 330 may be added without departing from the disclosure.

We now turn to the classifier 330. Essentially, the image processing module 321, mouse data processing module 322 and keyboard data processing module 333 processes the raw data to obtain relevant data and information required to be fed to a classifier 330. As mentioned above, the actions 314 of the student when performing task required of the lesson plans are monitored and injected to one of the classifiers 330. In figure 3, D_1 represent the first classifier, D_L represents the last classifier while D_i represents 2, 3,, L-1.

A machine learning based classifier 330 is constructed for each type of sensor features. The classifier includes random forest, naïve Bayes, multinomial logistic regression, etc. The workings of the classifier are well known and will be omitted from this disclosure. What the classifier 330 is required to achieve is to apply a predictive model on the inputs received from the modules 321-333 and arrange the inputs in a manner according to the selected classifier 330. The classifier 330 that is implemented in the system and method is not restricted to one type of classifier as mentioned above. Rather, a few classifiers may be implemented without departing from the disclosure. For example, D_1 may implement the random forest, D_i may implement the naïve Bayes while D_L may implement the multinomial logistic regression. Alternatively, D_1 and D_i may implement the random forest while D_L may

implement the naïve Bayes. In short, various permutations may be implemented without departing from the disclosure.

Each classifier output is combined into the final output 370 using the decision template fusion algorithm 360. If the decision template fusion algorithm 360 infers that the student is disengaged or frustrated, either a prompt or a hint will be generated and shown to the student. The algorithm for the generation of hints will be described below with reference to figure 8.

The decision template fusion algorithm 360 comprises a decision profile (DP) 361 and a decision template (DT) 362. The DP 361 receives all the output of the classifier 330 and re-arranges the output accordingly and can be expressed in the following manner,

$$DP(X) = F(D_1(X), D_i(X), D_L(X)) \text{ where } F \text{ is the fusion rule.}$$

The DP 361 may also be expressed in a matrix as follows,

$$DP(X) = \begin{bmatrix} d_{1,1}(X) & d_{1,j}(X) & d_{1,M}(X) \\ d_{i,1}(X) & d_{i,j}(X) & d_{i,M}(X) \\ d_{L,1}(X) & d_{L,j}(X) & d_{L,M}(X) \end{bmatrix}$$

$d_{i,j}$ denotes the probability or measure of confidence that classifier i considers the input or feature vector X to be of class j .

The DT 362 is the $L \times M$ (where L is the number of classifiers and M is the number of classes) matrix where $(k, s)^{th}$ element $DT_i(k, s)(Z)$ is computed with the following expression,

$$dt_i(k, s)(Z) = \frac{\sum_{j=1}^N Ind(z_j, i) d_{k,s}(z_j)}{\sum_{j=1}^N Ind(z_j, i)}$$

where $k=1 \dots L$, $s = 1 \dots M$ and $Z=\{z_1, \dots z_n\}$ is the training data set.

The DT_i for class i is the average of the DP of the elements of the training set Z of the decision template labelled in class i . When a test set $x \in R^n$ is submitted for classification, the similarity measure (μ_d^i) as expressed below is evaluate where a higher similarity measure equates to a higher support for that class.

$$\mu_d^i = 1 - \frac{1}{LM} \sum_{k=1}^L \sum_{s=1}^M (DT_i(k, s) - DP_i(X))^2$$

The Euclidean distance $(DT_i(k, s) - DP_i(X))^2$ is used as the similarity measure. i refers to the class number. The i value which gives the maximum similarity measure μ_d^i would be the final classification 370. For example, if class 1, μ_d^1 , denotes frustration and class 2, μ_d^2 , denotes non-frustration and $\mu_d^1 = 0.8$, $\mu_d^2 = 0.5$, the final classification 370 would be class 1 or that the student is frustrated.

As shown in figure 3, the affect-sensitive monitoring system comprises image processing module 321, mouse data processing module 322, keyboard data processing module 333, classifier 330 and a decision template fusion algorithm 360. The affect-sensitive monitoring system 100 may be provided as a software application performed by either the computing unit 110 or the server 190 without departing from the disclosure.

Figure 4 illustrates a flow diagram of process 400 performed by the processor in server 190 or processing unit 110 in accordance with an embodiment of this invention. In particular, process 400 illustrates the processes undertaken by the keyboard data processing module 333. Process 400 begins with step 405 to receive the keyboard timing logs. In response to receiving the keyboard timing logs, process 400 retrieves the baseline associated to the user. It should be noted that a student (or user) has to login with login name and password before beginning a software lesson. Hence, if the student is a first time user, the process has to perform some test to obtain a baseline for the student. This test may simply be requesting the student to perform typing of a paragraph to determine the typing speed of the student under normal condition. If the student is not a first time user, a baseline would have been created and stored. Hence, if process 400 is able to retrieve a baseline associated to the student, process 400 proceeds to step 420. Otherwise, in response to absence of a baseline associated to the student, process 400 proceeds to step 415 to determine a baseline for the student.

In step 415, process 400 performs some test to obtain and determine a baseline for the student. This test may include requesting the student to perform typing of a random

paragraph. The baseline latency for each student is then derived from the average of the latencies over a sliding window width of 10 keystrokes sample. Specifically, the algorithm for deriving the baseline latency can be described in the following manner.

1. Obtain the keystroke data from the keyboard timing log file specifically for the random paragraph performed by the student. The keystroke data comprising the latency for each key which is the sum of flight time (time duration from releasing a previous key to pressing the current key) and dwell time (time duration that the current key is pressed) and the key type (navigation, backspace, delete, alphanumeric, etc)
2. Eliminate all keys with duration greater than or equals to 1 second in the keystroke data.
3. Eliminate all navigation keys, backspace keys and delete keys from the keystroke data.
4. 30 consecutive keys are selected from the remaining keystroke data. One skilled in the art will recognise that the 30 consecutive keys may be selected from the first 30 keys, last 30 keys or anywhere between the first and last keys from the remaining keystroke data without departing from the disclosure. For the selected 30 keys,
 - a. Select the first 10 keys to form a window and calculate the mean of the keys duration and store in an array.
 - b. Move the window forward by 2 keys to form a next window, calculate the mean of the keys duration and append to the array.
 - c. Repeat step (b) for the next window. This will produce the array of $[k_0, k_1, \dots, k_{10}]$
5. Baseline latency = median of mean duration values in the array.

If process 400 is unable to determine a baseline after the student finish typing a paragraph, step 415 would be repeated and the student would be required to type a different paragraph. After a baseline is determined, the baseline for the student is stored in the memory. One skilled in the art will recognise that step 415 may be executed immediately after the student login to the software lesson without departing from the disclosure. Hence, at step 410, process 400 would likely be able to retrieve the baseline of the student and proceed to step 420.

In step 420, process 400 computes the dwell and flight time for each key. This determines the latency for each key. Dwell time refers to the time duration that a key is pressed while flight time refers to the time duration in between releasing a key and pressing the next key. For example, the latency for a key is the sum of flight time (time duration from releasing a previous key to pressing the current key) and dwell time (time duration that the current key is pressed).

The keystroke features are normalized with the calculated baseline flight time for each user in step 415. This is required since different students may type at different speeds even when they are in a neutral affective state. In order to mitigate the effect of the different typing speeds of students, it is pertinent to normalize the keystroke latencies across students. To achieve this, the keystroke latencies for individual students are divided by their individual baseline latency obtained in step 410.

In step 430, process 400 obtains an aggregated feature vector of the keystrokes. In particular, the normalised keystroke features are aggregated into sliding window of a particular size and these will be passed into the machine learning based classifier 330 and the classifier will output whether the student is frustrated or not. The classifier 330 has been pre-trained with pre-collected logs and labels of frustration. The normalised keystroke features are aggregated into sliding window sizes of 30 seconds, with an overlap of 66.7% of the window size. Figure 5 illustrates 2 sliding windows 510 and 520 with an overlap of 66.7% (i.e. 2/3 overlap). The first sliding window comprises 13 normalised keystroke features 5101-5113. As can be observed, normalised keystroke feature 5109 has a longer latency compared to the rest of the normalised keystroke features. This is because normalised keystroke feature 5109 is a “pause”. “Pause” refers to flight time extending beyond certain period of time. Essentially, a “pause” is a measurement of time taken by the student to enter a next key. The second sliding window 520 comprises 14 normalised keystroke features 5105-5120. Since the second sliding window 520 has a 20 second overlaps from the first sliding window 510, the first 20 seconds of the second sliding window comprises the same keystrokes as the last 20 seconds of the first sliding window. In this case, part of normalised keystroke feature 5105 and normalised keystroke features 5106-5113 are replicated in the second sliding window 520.

The aggregated features for the past 30 seconds window will be forwarded to the classifier 330 at 10 seconds interval, meaning every 10 seconds, the aggregated features for the past 30 seconds will be forwarded to the classifier. The features will be in the set $[x_1, x_2, x_3, x_n]$. Using the example in figure 5, features 5101-5113 will be forwarded to the classifier 330. After an interval of 10 seconds, features 5105-5120 will be forwarded to the classifier 330.

Process 400 ends after step 430.

An aggregated feature vector for actions 314 is also derived and injected to the aggregated feature vector of the keystrokes to form a combined feature array prior to forwarding to the classifier 330. Similar to step 430 in process 400, the actions 314 are aggregated into sliding windows and these will be combined with the sliding windows of the keystrokes obtained from step 430. Specifically, the aggregated features for the past 30 seconds window from both keystrokes 313 and actions 314 will be forwarded to the classifier 330 at 10 seconds interval. This means that every 10 seconds, the aggregated features for the past 30 seconds of both keystrokes 313 and actions 314 will be forwarded to the classifier 330. The features will be in the set $[x_{1,1}, x_{1,2}, x_{1,3}, x_{1,n}, x_{2,1}, x_{2,2}, x_{2,3}, x_{2,n}]$.

Figure 6 illustrates a flow diagram of process 600 performed by the processor in server 190 or processing unit 110 in accordance with an embodiment of this invention. In particular, process 600 illustrates the processes undertaken by the mouse data processing module 322. Process 600 begins with step 605 to receive the mouse timing logs 312.

In step 610, based on the mouse timing logs, process 600 records the exact timestamp of each mouse clicked and determines whether the type of mouse actions, namely, single click and double click. The process of determining single and double clicks is known in the art and hence will be omitted for brevity.

In step 6150, process 600 derives the aggregated feature vectors for the mouse action. Specifically, based on the timestamps and types of mouse actions determined, the

number of single clicks in less than 2 seconds and number of double clicks are generated within the time window of 30 seconds.

In step 630, process 600 obtains an aggregated feature vector of the mouse action. In particular, based on the timestamps and types of mouse actions determined in step 610, the number of single clicks in less than 2 seconds and number of double clicks are generated within the sliding window of a particular size and these will be passed into the machine learning based classifier 330 and the classifier will output whether the student is frustrated or not. The classifier 330 has been pre-trained with pre-collected logs and labels of frustration. Similar to that described in keystrokes, the number of single clicks in less than 2 seconds and number of double clicks are aggregated into sliding window sizes of 30 seconds, with an overlap of 66.7% of the window size. Each sliding window comprises the mouse action features such as number of single clicks in less than 2 seconds and number of double clicks over a period of 30 seconds (which is according to the size of the sliding window).

The aggregated mouse action features for the past 30 seconds window will be forwarded to the classifier 330 at 10 seconds interval. Process 600 ends after step 615.

Returning back to figure 3 that shows that aggregated mouse action features are forwarded to a classifier 330 and combined aggregated keystroke and action features are forwarded to a separate classifier 330, it is possible that the aggregated mouse action, keystroke and student action features are combined into a single feature array and forwarded to one classifier 330 without departing from the disclosure. Head pose and facial features are respectively another 2 separate feature array. In brief, any permutations may be implemented without departing from the disclosure.

If the time at which frustration is observed falls in the overlap area of 2 consecutive window slices, both window slices would be annotated as slices in which the student experiences frustration. Alternatively, if the time at which frustration is observed falls outside the overlap area, only the time window slice in which it occurs in will be annotated as the slice in which the student experienced frustration. The time window size of 30 seconds is

established as being the most optimal from previous experiments. In the actual frustration sensing, I will aggregate the keystrokes into windows of 30 seconds. However, one skilled in the art will recognise that the sliding window size may not be limited to 30 seconds and other variation such as 60, 90, 120 seconds may be implemented without departing from the disclosure.

Choice of window

The choice of a time window of affect detection is important as it informs on the affective state of students in a timely manner. A larger time window width would mean that students' frustration was left unattended for a long period of time and that increases the risk of losing the students totally. On the other hand, a shorter time window width would result in lower accuracy and performance as lesser data is accumulated within the short time window for analysis. Thus, there is a need to evaluate the optimal time window size for the model which balances timely detection of students' affective state with the accuracy of detection.

In cases where keys are sparse for a period, a larger time window width would ensure that more keys are accumulated in a time window which would in turn lead to a more accurate detection of students' affective state. The advantage of a larger time window width is that the model will need a longer period of time to establish students' affective state.

An observation noted is that when students are frustrated, the frustration may be manifested in their actions leading up to the observation and that it also persists for a period of time after the observation. To cater for this, the time window slices are overlapped such that if frustration is observed in the overlapped period, both the keystrokes in the time window slice leading up to the observation and the keystrokes in the time window slice after the observation will be included in the evaluation of the affective state (frustrated versus non-frustrated). A test was conducted and was noted that an optimal time window is 30 seconds. Using a 30 seconds time window, the sliding window with size of 30 seconds would include an overlap of 20 seconds of the window size.

Industrial application

We will now describe the processes above being implemented in a lab setting. Figure 7 illustrates a lab with 11 participants where 10 are students and 1 is a tutor. A computing unit 180 is provided to each of the 11 participants.

A dashboard with the layout of the lab as shown in figure 7 will show a heat map to direct the tutor to the students who require assistance. The dashboard will also display the scores obtained by the students and the topics which students are weak in. The affect-sensitive monitoring application according to this disclosure can be implemented by the educational, training institutions and Massive Open Online Course (MOOC) providers. Further details on the implementation of the affect-sensitive monitoring application in the lesson plans provided by educational, training institutions and MOOC providers will now be described as follows.

Students and tutor are required to login to the computing unit 180, via a login interface, in order to proceed with a lesson plan. All participants will have different login details with certain privileges. For example, a tutor would be able to access the dashboard but not the students. Once login, the students have to select the appropriate lesson plan. Figure 8 illustrates a process 800 performed by the processor in server 190 and/or processing unit 110 in accordance with an embodiment of this invention. In particular, process 800 illustrates the processes undertaken by a number of computing units 180 (each operated by a student or teacher) and a server 190. In some embodiment, the computing units 180 may be operated by the students while the server is operated by the teacher without departing from the invention.

Process 800 begins with step 805 where the participant login to the system to access a lesson plan. As mentioned above, all participants will have different login details with certain privileges. For example, a tutor would be able to access the dashboard but not the students.

In step 810, process 800 retrieves the relevant data to display to the participant accordingly. For example, if process 800 determines a student login for the first time, process 800 would proceed to determine the baseline of the student in step 815. If process

800 determines a student login (non-first time participant), process 800 would retrieve the baseline and proceed to main page that allows student to select the relevant lesson plan. If process 800 determines a teacher login, process 800 would proceed to display the dashboard with other options such as selecting lesson plans, etc.

In step 815, process 800 determines the baseline of certain actions of the participant. For example, if keyboard actions are monitored, process 800 determines the baseline as described in step 415 of process 400 above. All baselines required in the affect-sensitive monitoring system as described in process 300 would be determined at this juncture. In other words, step 415 of process 400 would be brought forward to immediately after the login step of step 810 and before executing the affect-sensitive monitoring system as described in process 300.

In step 820, process 800 receives a selection and retrieves the relevant document. Specifically, the participants have to select the correct lesson plan in order for the process to retrieve the document associated to the selected lesson plan. Figure 9 illustrates an example of a selected lesson plan 900. In this example, “Exercise 1” 910 has been selected which is a lesson on Java coding. In “Exercise 1”, a participant has to complete the code to calculate the income tax payable by a tax payer based on the number of dependents that he or she has. Specifically to “Exercise 1”, participants are required to complete the code to declare appropriate constants and variables 920. Figure 9 illustrates a completed lesson plan where the constants and variables 920 are entered.

After step 820, participants would be going through the lesson plan. Specifically to “Exercise 1”, the participant would be filling in the codes to complete the lesson plan. After the participant fills up the codes, the participant will compile the codes to determine whether he/she has entered to correct codes. If the codes are correct, the lesson plan ends and process 800 proceeds to step 830. During the lesson plan, the affect-sensitive monitoring system as described in process 300 would be running in the background to determine the affect state of the participant. This means that immediately after step 820, process 800 will proceed to step 825 to execute the affect-sensitive monitoring system. Further details of step 825 will now be described as follows.

In step 825, process 800 executes the affect-sensitive monitoring system as described in process 300. The affect-sensitive monitoring system would be running in the background. Accordingly, affect-sensitive monitoring system of process 300 would begin monitoring for the affect state of the participants such as frustration. If the final classification 370 indicates that the participant is frustrated and having difficulties, process 800 would provide active measure to assists the participant. In such scenario, a Student's Actions Module may be provided to identify the relevant assistance required. The student's action module includes a student subsystem logs that details the history of students' interactions with the tutoring system. These include the duration of time spent on an exercise, the topic and exercise the student is working on, the date and time of compilation, the compilation error details and the number of lines of errors for each compilation attempt. Based on this historical data, the Student's Actions Module is able to extract the relevant historical actions of students to provide assistance to the student.

The above is a description of embodiments of a system and method in accordance with this invention. It is envisioned that others can and will design alternative methods and systems as set forth in the following claims based upon this disclosure.

What is claimed is:

1. An affect-sensitive monitoring system for determining frustration of a student comprising:

a processing unit and a memory;

an image capturing device communicatively connectable to the processing unit;

a keyboard communicatively connectable to the processing unit;

a mouse communicatively connectable to the processing unit;

instructions stored on the memory;

in response to receiving a selected software application from a user, the instructions are executable by the processing unit to:

receiving sequence of images from the image capturing device;

receiving a mouse timing log file;

receiving a keyboard timing log file;

determining aggregated keyboard actions based on the keyboard timing log file;

determining aggregated mouse actions based on the mouse timing log file;

determining aggregated image actions based on the sequence of images;

arranging each of the aggregated keyboard actions, mouse actions and image actions according to at least one selected machine learning based classifier;

applying a decision template fusion algorithm on the arranged aggregated keyboard actions, mouse actions and image actions to obtain a final classification that indicates affect state of the user.

2. The affect-sensitive monitoring system according to claim 1 wherein the instruction to determining aggregated keyboard actions based on the keyboard timing log file comprising instructions to:

retrieving a baseline latency associated to the user;

determining a latency for each key with the following expression:

latency of each key = flight time + dwell time

where flight time is time duration from releasing a previous key to pressing the current key and dwell time is time duration that the current key is pressed;

normalising the latency of each key based on the retrieved baseline latency; and
aggregating the normalised keys into a sliding window of a predetermined size to obtain the aggregated keyboard actions.

3. The affect-sensitive monitoring system according to claim 2 wherein the instruction to aggregating the normalised keys into a sliding window of a predetermined size to obtain the aggregated keyboard actions comprises:

aggregating into sliding window sizes of 30 seconds, with an overlap of 66.7% of the window size.

4. The affect-sensitive monitoring system according to claim 3 further comprises instructions to:

forwarding the aggregated keyboard actions for the past 30 seconds window to the classifier at 10 seconds interval.

5. The affect-sensitive monitoring system according to any one of claims 2-4 wherein the instruction to retrieving a baseline associated to the user comprises:

obtaining a new baseline in response to absence of a baseline associated to the user.

6. The affect-sensitive monitoring system according to claim 5 wherein the instruction to obtaining a new baseline comprises:

requesting the user to perform typing of a random paragraph;

determining the baseline latency based on a keystroke data obtained for the random paragraph performed by the user, the keystroke data comprises latency for each key which is the sum of flight time and dwell time and the key type.

7. The affect-sensitive monitoring system according to claim 6 wherein the instruction to determining the baseline latency based on a keystroke data obtained for the random paragraph performed by the user comprises:

eliminating all keys with duration greater than or equals to 1 second from the keystroke data;

eliminating all keys with key type associated to navigation, backspace and delete from the keystroke data;

selecting 30 consecutive keys from the remaining keystroke data;

forming a window for the first 10 keys of the selected 30 consecutive keys, calculating the mean of the keys duration and store the calculated mean in an array;

forming a next window by moving the window forward by 2 keys, calculate the mean of the keys duration and append the array to include the calculated mean for the next window;

repeating the step of forming the next window until eleven windows are formed; and

determining baseline latency with a median of mean duration values in the array.

8. The affect-sensitive monitoring system according to any one of claims 1-7 wherein the instruction to determining aggregated mouse actions based on the mouse timing log file comprises:

recording the exact timestamp of each mouse click;

determining the type of mouse actions wherein the type of mouse actions includes single click and double click;

aggregating the determine mouse actions into a sliding window of a predetermined size to obtain the aggregated mouse actions.

9. The affect-sensitive monitoring system according to claim 8 wherein the instruction to aggregating the determine mouse actions into a sliding window of a predetermined size to obtain the aggregated mouse actions comprises:

aggregating into sliding window sizes of 30 seconds, with an overlap of 66.7% of the window size wherein each sliding window contains the determined single click and double clicks.

10. The affect-sensitive monitoring system according to claim 8 further comprises instructions to:

forwarding the aggregated mouse actions for the past 30 seconds window to the classifier at 10 seconds interval.

11. The affect-sensitive monitoring system according to claim 1 wherein the machine learning based classifier comprises random forest, naïve Bayes, or multinomial logistic regression.

12. The affect-sensitive monitoring system according to claim 1 wherein the instruction to applying a decision template fusion algorithm on the arranged aggregated keyboard actions, mouse actions and image actions to obtain a final classification that indicates affect state of a subject comprises:

a decision profile, DP, that receives the output of the classifier and re-arranges the output accordingly in the following expression,

$$DP(X) = \begin{bmatrix} d_{1,1}(X) & d_{1,j}(X) & d_{1,M}(X) \\ d_{i,1}(X) & d_{i,j}(X) & d_{i,M}(X) \\ d_{L,1}(X) & d_{L,j}(X) & d_{L,M}(X) \end{bmatrix}$$

$d_{i,j}$ denotes the probability or measure of confidence that classifier i considers the input or feature vector X to be of class j ;

a decision template, DT, that is an $L \times M$ matrix where L is the number of classifiers, M is the number of classes and $(k, s)^{th}$ element, $DT_i(k, s)(Z)$, is computed with the following expression,

$$dt_i(k, s)(Z) = \frac{\sum_{j=1}^N Ind(z_j, i) d_{k,s}(z_j)}{\sum_{j=1}^N Ind(z_j, i)}$$

where $k = 1 \dots L$, $s = 1 \dots M$ and $Z = \{z_1, \dots, z_n\}$ is the training data set, and DT_i for class i is the average of the DP of the elements of the training set Z labelled in class i .

determining the final classification, μ_d^i , with the following expression,

$$\mu_d^i = 1 - \frac{1}{LM} \sum_{k=1}^L \sum_{s=1}^M (DT_i(k, s) - DP_i(X))^2$$

wherein the i value that gives the maximum similarity measure μ_d^i would be the final classification.

13. An affect-sensitive monitoring method for determining frustration of a student comprising:

in response to receiving a selected software application from a user:

receiving sequence of images from the image capturing device;

receiving a mouse timing log file;

receiving a keyboard timing log file;
determining aggregated keyboard actions based on the keyboard timing log file;
determining aggregated mouse actions based on the mouse timing log file;
determining aggregated image actions based on the sequence of images;
arranging each of the aggregated keyboard actions, mouse actions and image actions according to at least one selected machine learning based classifier;
applying a decision template fusion algorithm on the arranged aggregated keyboard actions, mouse actions and image actions to obtain a final classification that indicates affect state of the user.

14. The affect-sensitive monitoring method according to claim 13 wherein the step of determining aggregated keyboard actions based on the keyboard timing log file comprises:

retrieving a baseline latency associated to the user;
determining a latency for each key with the following expression:
$$\text{latency of each key} = \text{flight time} + \text{dwell time}$$

where flight time is time duration from releasing a previous key to pressing the current key and dwell time is time duration that the current key is pressed;
normalising the latency of each key based on the retrieved baseline latency; and
aggregating the normalised keys into a sliding window of a predetermined size to obtain the aggregated keyboard actions.

15. The affect-sensitive monitoring method according to claim 14 wherein the step of aggregating the normalised keys into a sliding window of a predetermined size to obtain the aggregated keyboard actions comprises:

aggregating into sliding window sizes of 30 seconds, with an overlap of 66.7% of the window size.

16. The affect-sensitive monitoring method according to claim 15 further comprises:

forwarding the aggregated keyboard actions for the past 30 seconds window to the classifier at 10 seconds interval.

17. The affect-sensitive monitoring method according to any one of claims 15-16 wherein the step of retrieving a baseline associated to the user comprises:

obtaining a new baseline in response to absence of a baseline associated to the user.

18. The affect-sensitive monitoring method according to claim 17 wherein the method of obtaining a new baseline comprises:

requesting the user to perform typing of a random paragraph;

determining the baseline latency based on a keystroke data obtained for the random paragraph performed by the user, the keystroke data comprises latency for each key which is the sum of flight time and dwell time and the key type.

19. The affect-sensitive monitoring method according to claim 18 wherein the method of determining the baseline latency based on a keystroke data obtained for the random paragraph performed by the user comprises:

eliminating all keys with duration greater than or equals to 1 second from the keystroke data;

eliminating all keys with key type associated to navigation, backspace and delete from the keystroke data;

selecting 30 consecutive keys from the remaining keystroke data;

forming a window for the first 10 keys of the selected 30 consecutive keys, calculating the mean of the keys duration and store the calculated mean in an array;

forming a next window by moving the window forward by 2 keys, calculate the mean of the keys duration and append the array to include the calculated mean for the next window;

repeating the step of forming the next window until eleven windows are formed; and determining baseline latency with a median of mean duration values in the array.

20. The affect-sensitive monitoring method according to any one of claims 13-19 wherein the step of determining aggregated mouse actions based on the mouse timing log file comprises:

recording the exact timestamp of each mouse click;

determining the type of mouse actions wherein the type of mouse actions includes single click and double click;

aggregating the determine mouse actions into a sliding window of a predetermined size to obtain the aggregated mouse actions.

21. The affect-sensitive monitoring method according to claim 20 wherein the step of aggregating the determine mouse actions into a sliding window of a predetermined size to obtain the aggregated mouse actions comprises:

aggregating into sliding window sizes of 30 seconds, with an overlap of 66.7% of the window size wherein each sliding window contains the determined single click and double clicks.

22. The affect-sensitive monitoring method according to claim 20 further comprises:

forwarding the aggregated mouse actions for the past 30 seconds window to the classifier at 10 seconds interval.

23. The affect-sensitive monitoring method according to claim 13 wherein the machine learning based classifier comprises random forest, naïve Bayes, or multinomial logistic regression.

24. The affect-sensitive monitoring method according to claim 13 wherein the step of applying a decision template fusion algorithm on the arranged aggregated keyboard actions, mouse actions and image actions to obtain a final classification that indicates affect state of a subject comprises:

from a decision profile, DP, receiving the output of the classifier and re-arranging the output accordingly in the following expression,

$$DP(X) = \begin{bmatrix} d_{1,1}(X) & d_{1,j}(X) & d_{1,M}(X) \\ d_{i,1}(X) & d_{i,j}(X) & d_{i,M}(X) \\ d_{L,1}(X) & d_{L,j}(X) & d_{L,M}(X) \end{bmatrix}$$

$d_{i,j}$ denotes the probability or measure of confidence that classifier i considers the input or feature vector X to be of class j ;

from a decision template, DT, that is an $L \times M$ matrix where L is the number of classifiers, M is the number of classes and $(k, s)^{th}$ element, $DT_i(k, s)(Z)$, computing with the following expression,

$$dt_i(k, s)(Z) = \frac{\sum_{j=1}^N Ind(z_j, i) d_{k,s}(z_j)}{\sum_{j=1}^N Ind(z_j, i)}$$

where $k = 1 \dots L$, $s = 1 \dots M$ and $Z = \{z_1, \dots, z_n\}$ is the training data set, and DT_i for class i is the average of the DP of the elements of the training set Z labelled in class i .

determining the final classification, μ_d^i , with the following expression,

$$\mu_d^i = 1 - \frac{1}{LM} \sum_{k=1}^L \sum_{s=1}^M (DT_i(k, s) - DP_i(X))^2$$

wherein the i value that gives the maximum similarity measure μ_d^i would be the final classification.

25. An educational application comprising:

a login interface for receiving login details from users;

a plurality of lesson plans;

a dashboard in response to a login from a user with a tutor privilege, the dashboard displaying a heat map to identify users who require assistance and scores for each student; and

an affect-sensitive monitoring method according to claim 13 executed in response to successful login from user.