Hidden Markov Model for Inferring User Task Using Mouse Movement

Conference Paper · October 2013

DOI: 10.1109/ICTA.2013.6815305 · Source: arXiv

3 authors, including:

Anis Elbahi
University of Monastir

Mohamed Ali Mahjoub
University of Sousse

Hidden Markov Model for Inferring User Task Using Mouse Movement

Anis Elbahi [#1], Mohamed Ali Mahjoub [*2], Mohamed Nazih Omri [#3]

 # Research Unit MARS,
Department of computer sciences
Faculty of sciences of Monastir, Monastir, Tunisia.
[1]Elbahi.anis@gmail.com
[3]MohamedNazih.Omri@fsm.rnu.tn
** Research Unit SAGE,*
National Engineering School of Sousse, Sousse, Tunisia.
[2]Medali.mahjoub@ipeim.rnu.tn

Abstract— **The assistive technology and e-learning have been widely used to improve web accessibility for disabled users. One of the issues of online web-based applications is to understand how a user interacts with online application and the strategy by which he reasons to perform a given activity. Know what the web user is doing can provide useful clues to better understand his behavior in order to guide him in his interaction process.**
This study proposes a methodology to analyze user mouse movement in order to infer the task performed by the user. To do this, a Hidden Markov Model is used for modeling the interaction of the learner with an e-learning application. The obtained results show the ability of our model to analyze the interaction in order to recognize the task performed by the learner.

Keywords— **E-Learning web based application, Accessibility, Interaction analysis, Hidden Markov Models, Mouse movements, Task inference.**

I. INTRODUCTION

Some disabled users have problems in handling new technologies using traditional human machine interaction tools. Usually, in online applications, pointing device must be used; therefore assistive technologies [25] offered various mutations of classical mouse device [26], [27] for users with disabilities. In human machine interaction process, it is important to know how the user interacts with the interface to perform an activity and the strategy by which he reaches his goal.

Indeed, analysis and recognition of the user task, especially disabled learner, may be too useful to provide feedback for the learning process in order to guide learner especially the one who is unable to express himself normally.

The analysis of user behavior [20], [21] and activity recognition [1], [22] in a web user interaction process is one of the most popular subjects of the human computer interaction and various studies have been achieved in this context to analyze the navigational behavior of the user in order to infer the activity provided, using techniques such as eye tracking [1]-[3], psychological and physiological tracking [4] and mouse cursor tracking [5]-[8]. Many earlier works show that it is obvious that mouse trajectories can be recorded, processed, averaged, visualized, and explored for analyzing user behavior.

Generally, the trajectory of the mouse is considered as a powerful tool to provide indicators of the way in which a person interacts with the interface. In fact, mouse movements are guided by the goal of the task and reflect the cognitive process of the user as shown in Fig. 1.

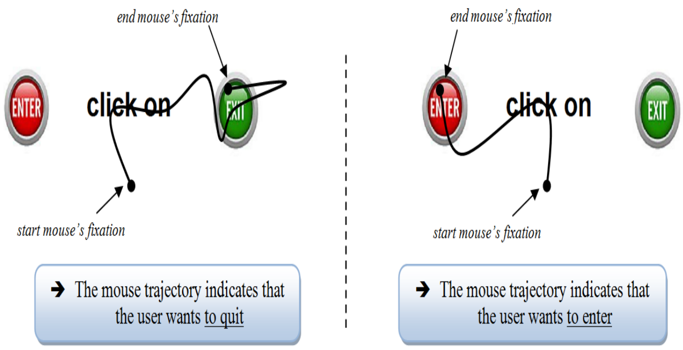

Fig. 1 Mouse trajectories for two different tasks

Have clues of cognitive process of the user using the trajectory of the mouse, helps to infer his strategy to accomplish his goal, then know his needs and provide him assistance.

In [9] "an inverse Yarbus process" whereby the authors infer the visual task by observing the measurements of a viewer's eye movements while executing the visual task.

The eye-tracking technology provide a real-time tracking of the cognitive activity of users [19], but the use of this technique as a tool to infer the user task may not be an affordable alternative in various cases because the data provided by the eye tracking are expensive and may be biased [23], indeed subjects must be present in a specific location to wear (or not) a tracking device and they are aware of the test situation, hence the conditions of the experiment of gaze tracking may affect the found results much more than usual experience handling the mouse device. On the other side, Chen and al [10] found that over 75% of cases, a mouse saccades moves to significant regions of the screen and in these cases, it is quite likely that the eye gaze is very near to the cursor. For these reasons, we conclude the importance of cursor path to have clues of the user cognitive process.

In the present article, we propose to use the Markov theory to model the user interaction using the cursor trajectory in e-learning web based application. The main goal is to infer the learner task in order to improve the e-learning process.

II. Task Inference to improve accessibility for students with disabilities

According to "disabled-world.com" statistics, around 10% of the total world's population live with a disability. For a long time, disabled people who want to learn, are faced with obstacles related to their handicap which reduces their opportunities to have an learning career similar to safe students, but thanks to new technologies and the aim of improving the accessibility for students with special needs, assistive technology [25] and e-learning applications, offer for disabled students, more chance to have a course of study similar to safe students.

One of the famous examples of the utility provided by assistive technology for students with disabilities to interact with their environment is Stephan Hawking, this person has shown that the handicap was never been an obstacle to learning.

One of the most used devices in human computer interaction is the pointing device. Thanks to the assistive technology, people with disabilities can move the cursor, make clicks, execute commands, … using various techniques such as eye tracking, head tracking and tongue movements tracking [26]-[28].

For students with disabilities who can use a pointing device but can't express themselves normally, it had to have a mechanism -which is based on data relating to the pointing device handling- to help us to answer some questions like:

- Is the student has done the requested task correctly?
- What is the strategy by which the learner resolves a given problem?
- What task he repeated several times?
- What is the task that has not been done?
- …

Answering these questions, can give us clues about the level of success of the learning process, problems faced by the learner, the necessary guidance to learner and possible improvements in the learning application.

III. Related research

The learner is the "center of gravity" of the e-learning process. In addition, adaptive and intelligent web-based educational systems attempt to be more adaptive by using the student profile associated with the goals, preferences and knowledge of the each individual student [15]. Various ways such as log files were used to better understand the progress of students and to guide them in the e-learning process [17], [20], to provide good support for web users and to better understand the learner behavior. In the same way, the analysis of the trajectory of the mouse has been widely used to infer learner's strategy. Ohmori and al [14] analyse the mouse movements during the reading task to make a classification of learners in three patterns. In [5], authors explore mouse movements to provide insights into the intent behind a web search query. In [17], an application was suggested to track mouse movements of learners during their learning process in order to enable teachers to better understand the behavior of their students. Various tools have been developed for the capture and analysis of the trajectory of the mouse like OGAMA[1] [13]. Mueller and Lockerd [24] develop the "Cheese" tool by which they analyze and investigate mouse behavior trends. Freeman and Ambady [16] present "MouseTracker" tool that evaluate real-time processing in psychological tasks.

Thanks to sophisticated tools for tracking mouse movements and clicks, it became possible to have high-quality data that reflect a spontaneous, direct, precise and measurable trace of the interaction process, therefore mouse data like trajectory of the cursor can be considered as indicator of a part of the cognitive process of user.

As already presented, various studies have been performed to study the behavior of the learner based on mouse movements, but to our knowledge, there is not yet a study which is based on mouse movements for the task automatic recognition using a web-based application.

In the present paper we will show how the mouse trajectory can be used with Hidden Markov Model (HMM) to model the interaction, in order to recognize the task performed by the learner in an e-learning web-based application.

IV. Hidden Markov Models (HMM)

A. Presentation:

Although HMMs have been introduced since the late 60s by Baum and al, they are still overused in various fields for modeling stochastic sequences [12], [9] as the analysis and speech recognition, handwriting recognition, image recognition, DNA sequence analysis , activity recognition, ...

In the present article, we do not try to detail the HMMs. However, for an excellent tutorial covering the basic HMM technologies, the reader is recommended to see the documents of R. Rabiner [11], [12].

B. Formal definition:

Briefly, a HMM is a statistical mathematical model used to describe a doubly stochastic process, which can be presented by the following figure:

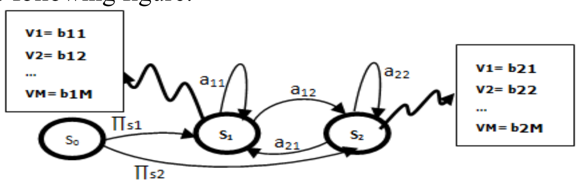

Fig. 2 HMM with two hidden states

---

[1] OGAMA available on : http://www.ogama.net/

Formally, a HMM noticed by $\lambda = (A, B, \Pi)$ is defined by:

- $S = \{S_1, S_2, ..., S_N\}$: set S of N hidden states with S_0 is a start state. (At time $_t$ the model state is q_t).
- $O = \{O_1, O_2, ..., O_M\}$: the alphabet of M symbols.
- $A=\{a_{ij}\}$ / $a_{ij} = p(q_t=S_j \mid q_{t-1}=S_i)$, $1 \leq i,j \leq N$, $\sum_{j=1}^{N} a_{ij} = 1$: the matrix of transition probability between the N states.
- $B=\{b_j(k)\}$ / $b_j(k) = p(O_k=O_t \mid q_t=S_j)$, $1 \leq j \leq N$, $1 \leq k \leq M$, $\sum_{j=1}^{N} b_j(K) = 1$: the matrix of symbol transmission probability by N states.
- $\prod = \{\prod_i\}$ / $\prod_i = p(q_1=S_i)$, $1 \leq i \leq N$, $\sum_{i=1}^{N} \prod_i = 1$: the Initial probability distribution vector.

The system starts with an initial transition from state S_0 to all other states S_i of the model with an initial transition probability Π_i ; $0 \leq \Pi_i \leq 1$.

The system can pass from any state S_i to an other state S_j with a transition probability a_{ij} ; $0 \leq a_{ij} \leq 1$.

From each state S_j the system generates a symbol O_k with an emission probability $b_j(k)$; $0 \leq b_j(k) \leq 1$.

C. Markov property:

A stochastic process has the Markov property if the prediction of the next state depends only on the relevant information contained in the present state of the process, ie:

$$P(X_{n+1}=j \mid X_0, X_1, ..., X_{n=i}) = P(X_{n+1}=j \mid X_{n=i})$$

D. HMM's problems and their solutions :

Like shown in table I, Rabiner and Juang in [11] explain the three key problems of interest that must be solved for the model to be useful in real world application, given a model $\lambda = (A, B, \Pi)$ and observation sequence $O=\{O_1, O_2, \ldots, O_T\}$.

TABLE I

PROBLEMS / SOLUTIONS OF HMMS

	Description of the problem	Solution algorithm
Prob1	How we compute $P(O \mid \lambda)$.	Forward-backward
Prob2	How we choose a state sequence $S=\{S_1, S_2, \ldots, S_T\}$ witch is optimal in some meaningful sense.	Viterbi
Prob3	How we adjust the model parameters to maximize $P(O \mid \lambda)$	Baum-Welch

E. HMM for task user modeling:

A task is the outcome of the interaction of the user with the interface areas to achieve a fixed goal.

Since the handling of the interface areas with mouse device is related to the aim of the task, the parts of the interface are not manipulated in the same manner for different tasks as shown in Fig. 1. Also, the task can be illustrated as a set of states among them the system makes a transition at each time t with a certain probability. The randomly transition to a state generates an observable symbol which is the zone crosses by cursor. So, the task can be defined as a doubly stochastic process, the first stochastic level is the transition from one state to another randomly, and the second stochastic level is the random generation of a symbol which is the area of interest manipulated by the user cursor. Then a user task can be modeled using a HMM. In the next section we will describe the proposed model.

V. PROPOSED MODEL

An e-learning activity can be defined as a set of tasks to be performed by a learner and each task can be defined as a set of actions. $TASK=\{TSK_1, TSK_2, \ldots, TSK_N\}$ is a set of tasks. To perform a task, the user focuses in the interface (with mouse cursor) areas, more than others; in fact the probability of using areas is strongly related to the task.

Let $AOIs = \{Z_1, Z_2, \ldots, Z_M\}$ the set of M Areas Of Interest (AOI) that can be manipulated or focused by the cursor during the task. Each AOI is used to perform an action of the task.

Identifying AOIs can be done by an expert who fixes parts of the interface that are needed to perform all tasks required by the user like shown in Fig. 3.

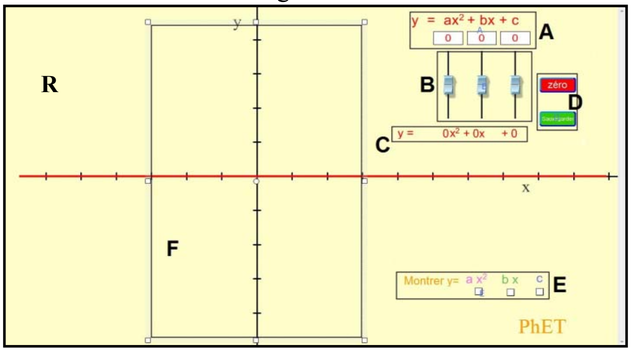

Fig 3: AOIs for the phet[2] "Equation Grapher" simulator.

A task is represented in our model by the trajectory of the mouse when a subject crosses a set of AOIs for a T period to achieve a goal. Hence, a task can be defined as set of AOI as follows: $TSK = \{Z_i(t)\}$; $1 \leq i \leq M$; $1 \leq t \leq T$.

After recording mouse path of the task, a step of vectorization will take place. The purpose of this step is to identify all areas that have been crossed during the task, using the following algorithm.

Vectorization algorithm:

<u>*Begin vectorization*</u>

<u>*Initializations:*</u>

O={} ;

Details of each AOI (Z_i);

T ← Total duration of the performed task;

ds ← Time between recordings of tow cursor coordinates;

<u>*Input:*</u>

Mouse trajectory

<u>*Treatment*</u>:

for t:=1 to T (with ds step) do

if (cursor coordinates of a mouse trajectory is in Z_i) then

O [t] ← Z_i

endif

end for

<u>*Output:*</u>

$O = \{O_1, O_2, ..., O_T\}$

<u>*End vectorization*</u>

[2] Phet available on http://phet.colorado.edu/sims/equation-grapher/equation-grapher_fr.html

Thus, the vectorization step result is $O=\{O_1,O_2, \ldots,O_T\}$; with $O_t= Z_i(t)$, $1 \le i \le M$, $1 \le t \le T$.

Fig. 4. explains the task inference process of the model.

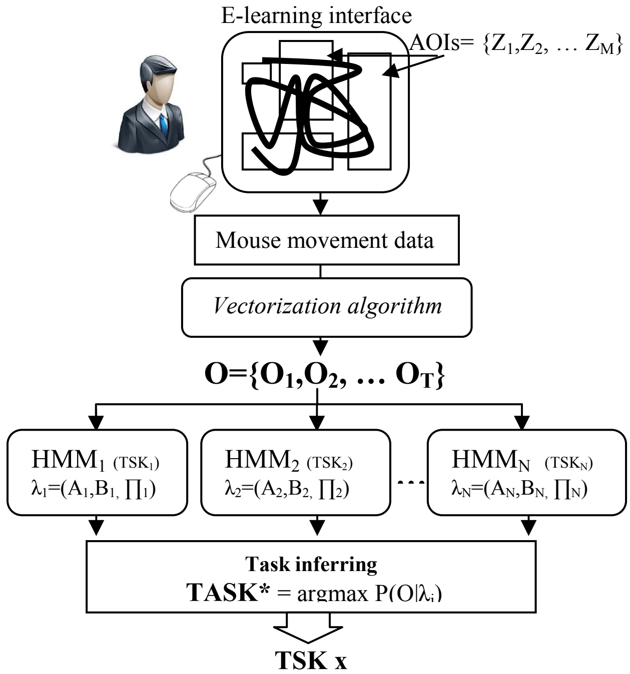

Fig 4. Task inference model using mouse movement data.

Formally, our model can be represented as follows:

- $TASK=\{TSK_1, TSK_2, \ldots ,TSK_N\}$: set of N hidden states. (with N is the number of tasks that can be performed by the user)
- $AOIs = \{Z_1,Z_2,\ldots, Z_M\}$: the alphabet of M symbols, that can be generated by hidden states. (the set of M areas of interest that can be manipulated or focused by the mouse during the task).
- A task is defined by $TSK=\{Zi(t)\}$; $1 \le i \le M$; $1 \le t \le T$.
- λ_{init} an initial HMM noticed by $\lambda_{init} = (A_{init}, B_{init}, \Pi_{init})$ and defined by :
  - $A_{init} = \{a_{ij}\}$ / $a_{ij} = p(q_t=TSK_j \mid q_{t-1}=TSK_i)$, $1 \le i,j \le N$, $\sum_{j=1}^{N} a_{ij} = 1$: the matrix of transition probability between the N states.
  - $B_{init} = \{b_j(k)\}$ / $b_j(k) = p(Z_k =Z_t \mid q_t=TSK_j)$, $1 \le j \le N$, $1 \le k \le M$, $\sum_{j=1}^{N} b_j(K) = 1$: the matrix of symbol transmission probability by N states.
- $\Pi_{init} = \{\prod_i\}$ / $\prod_i=p(q_1=TSK_i)$, $1 \le i \le N$, $\sum_{i=1}^{N} \prod_i = 1$: the Initial probability distribution vector.

In order to infer the task achieved by the learner, we proceed as follows:

1- Using the Baum-Welch algorithm for each task TSK_i, the initial model λ_{init} is instantiated and then parameterized using a training set of vectors of observations relating to task TSK_i already performed. After the step of adjusting parameters, each task TSK_i will be modeled by a specific HMM $\lambda_i = (A, B, \Pi)$; $1 \le i \le N$.

2- Using Forward-backward algorithm, for a given task TSK characterized by the sequence $O=\{O_1, O_2,\ldots, O_T\}$ and for each HMM λ_i, we calculate the probability $P_{TSK} = P(O \mid \lambda_i)$.

3 - To infer the task, we choose the task which has a maximum probability value calculated by the model:

$$\mathbf{TASK^* = \underset{1 \le i \le N}{argmax}\, [P(O \mid \lambda_i)]}$$

VI. Experiments and Technical Explanation

To validate our work, we chose the Phet simulator as a simple e-learning interface. An expert determines the details of interesting areas (AOIs) that can be manipulated by the learner using the mouse to perform an action of the task, as shown in Fig. 3. Then AOIs =\{A,B,C,D,E,F,R\} with R is the position of the mouse outside the areas A, B, C, D, E and F.

For simplicity, we asked students to achieve only one of the two tasks bellow:

TSK_1 = graphical representation of a curve (REP).

TSK_2 = graphical verification of curves intersection (INT).

For modeling tasks, firstly, for each task TSK_i, we prepared a training set (LB_i) by using OGAMA tool to record the mouse path on the "equation grapher" simulation, to do this, we asked 10 participants to perform graphical equation representations (REP) of their choice and 10 others participants to perform a task of checking the intersection of two curves (INT) of their choice. Each training set will be used for adjusting the initial model parameters λ_{init} to obtain an HMM relating to each task.

Secondly, as we have already presented, we propose an initial model $\lambda_{init} =(A_{init}, B_{init}, \Pi_{init})$ which parameters are:

$TASK = \{TSK_1, TSK_2\}$: set S of 2 hidden states.

AOIs = {A, B, C, D, E, F, R} : the alphabet of 7 symbols.

$$\prod_{init}= \begin{pmatrix} 1 \\ 0 \end{pmatrix} \qquad A_{init}= \begin{pmatrix} 0.5 & 0.5 \\ 0.5 & 0.5 \end{pmatrix}$$

$$B_{init}= \begin{pmatrix} 0.3 & 0.2 & 0.05 & 0.1 & 0.05 & 0.2 & 0.1 \\ 0.2 & 0.1 & 0.05 & 0.15 & 0.3 & 0.1 & 0.1 \end{pmatrix}$$

Thirdly, after a step of extracting data from the path of the mouse for each task of two training sets, we use the Baum-Welch algorithm to adjust the parameters, we obtain:

$\lambda1=(A1,B1,\prod1)$ learned with the Learning Base (LB1) concerning the first task.

$$\prod1 = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \qquad A1= \begin{pmatrix} 0.9535 & 0.0465 \\ 0.0604 & 0.9396 \end{pmatrix}$$

B1=

$$\begin{pmatrix} 0.3818 & 0.3596 & 0.0379 & 0.0091 & 0.0046 & 0.0553 & 0.1517 \\ 0.0192 & 0.0096 & 0.0048 & 0.8827 & 0.0645 & 0.0096 & 0.0096 \end{pmatrix}$$

$\lambda2=(A2,B2, \prod2)$ learned with the Learning Base (LB2) concerning the second task.

$$\prod2= \begin{pmatrix} 1 \\ 0 \end{pmatrix} \qquad A2= \begin{pmatrix} 0.9509 & 0.0491 \\ 0.0501 & 0.9499 \end{pmatrix}$$

B2=

$$\begin{pmatrix} 0.2356 & 0.3419 & 0.0536 & 0.0092 & 0.0046 & 0.1764 & 0.1787 \\ 0.0183 & 0.0092 & 0.0046 & 0.0963 & 0.8533 & 0.0092 & 0.0092 \end{pmatrix}$$

Once the parameters of each HMM were estimated, 10 experiments were performed in which participants were asked to perform a task of their choice (REP / INT) respectively for representation and intersection, and for each task an observation sequence $O = \{O_1, O_2, \ldots, O_T\}$ was generated using the vectorization algorithm cited above.

TABLE II
EXPERIMENTAL RESULTS

Task ID	Task time	Task type		Fixation % on each AOI							HMM1(R)	HMM2(I)	DECISION
		REP	INT	A	B	C	D	E	F	R			
T1	2800		+	**0,57**	**0,39**	**0,50**	**2,11**	**45,32**	**14,82**	**36,29**	-6908.1	**-3110.8**	INT
T2	3000		+	**0,00**	**8,00**	**0,70**	**1,70**	**70,60**	**5,43**	**13,57**	-7601.6	**-1977.4**	INT
T3	3000		+	**0,00**	**0,33**	**0,17**	**0,00**	**88,77**	**1,67**	**9,07**	-8203.4	**-1195.5**	INT
T4	1650	+		**36,42**	**43,70**	**3,52**	**7,58**	**0,00**	**0,06**	**8,73**	**-1884.5**	-2443.5	REP
T5	3700		+	**12,89**	**28,49**	**3,22**	**7,54**	**30,16**	**6,24**	**11,46**	-6733.4	**-4366.9**	INT
T6	2400	+		**15,67**	**51,13**	**7,42**	**7,88**	**3,25**	**5,54**	**9,13**	**-3379.6**	-3591.7	REP
T7	2540	+		**26,14**	**36,18**	**6,34**	**8,23**	**3,15**	**4,92**	**15,04**	**-3579.6**	-3943.3	REP
T8	2860	+		**27,27**	**34,20**	**4,79**	**15,63**	**4,72**	**3,25**	**10,14**	**-3618.4**	-4483.3	REP
T9	2050		+	**16,10**	**17,80**	**3,41**	**5,37**	**47,46**	**0,00**	**9,85**	-4115.1	**-1957.5**	INT
T10	2920	+		**44,73**	**34,04**	**0,24**	**9,69**	**0,00**	**3,56**	**7,74**	**-3223.8**	-4374.6	REP

To infer the task performed, each obtained observation sequence is estimated using both HMMs. The maximum likelihood value generated using forward-backward algorithm will be considered as an indicator of the task performed by the participant. The experimental results are shown in table II.

VII. RESULTS AND DISCUSSION

Using OGAMA, we record the path of the mouse and the heat map. The heat map can tell us about the more focused and the more ignored areas in the interface, and the mouse path presents the order by which the learner interacts with interface simulator for each task as shown in the Fig. 4.

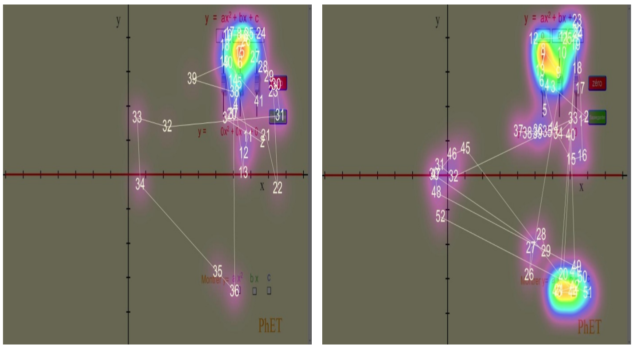

Fig 4. Two heat maps (with mouse paths) for two different tasks performed by the same learner using phet simulator.

The examination of each heat map shows that -without being aware-, the learner draws a mouse path, fixes areas more than others and spends a lot more time on elements more than others. Despite the significance and quality of data provided by the heat map and the mouse path, it is difficult to automatically infer the type of task performed.

For the 10 experiments performed, the table II shows for each task, its duration, nature, percentage of mouse fixations on each area of interest, the logarithm of the likelihood probability generated by each HMM and decision taken by the model. Remember that the HMM1 is part of the model trained to recognize TSK1 (REP) while HMM2 is part of the model trained to recognize TSK2 (INT).

For experiments T1, T2, T3, T5 and T9 in which participants made a simple graphical representation of a curve, the HMM1 generate higher values of probability then those generated by the HMM2 (remember that the HMMs values are on logarithm).

While for experiments T4, T6, T7, T8 and T10 where participants checked curves intersection, probability values generated by the HMM2 are higher than those generated by the HMM1.

So the model developed could infer the type of each task correctly. The following graph shows precisely the capacity of each HMM of the model to recognize each task.

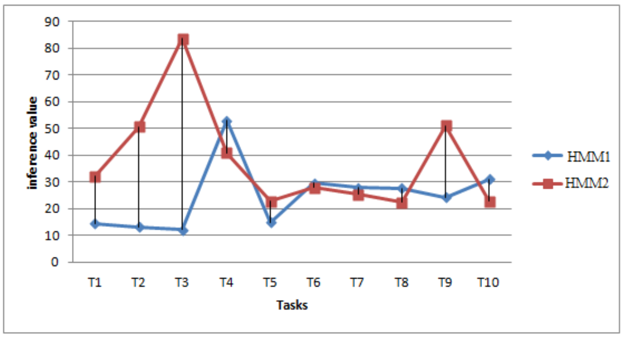

Fig 5. Task inference value for each HMM of the model

For experiments T1, T2, T3 and T9 there is a strong difference between the two HMMs whereas the other experiments the difference between the two HMMs is low.

Consider for example the experience T6 where the value generated by HMM1 is -3379,6 and by HMM2 is -3591,7. Although the model could infer the task correctly, the difference between values founded by the two HMMs is very low (182,1), whereas for the experience T3 the difference between values founded by the two HMMs is higher (7007,9)

This can be explained by the fact that learner performing the task T6 focuses on the areas A (15,67%) and B (51,13%) which are recognized by the first HMM1 as very important areas for the realization of task REP (according to the emission matrix estimated for HMM1), but the learner in this experiment manipulates other areas in relation to the task INT (C=7,42% ;D=7,88% ;E= 3,25% ;F=5,54%) , so the difference between tow HMMs is not very important.

Whereas in the case of task T3, the learner fixes more the area E (88,77%) which is considered very important for the realization of the task INT (according to the emission matrix estimated for HMM2) but he is not frequently uses areas in relation to the task REP (A=0% ;B=0,33% ;C= 0,17% ;D=0%; F=1,67%), so the difference between two HMMs is very important. The results of all the experiments can be explained in the same way.

From other side, the inference value can give an indication on how learner interacts with the e-learning application. In fact, for a learner doing a task, if the difference between two HMMs is important we conclude that the user targets well the areas needed to the achievement of the required task and therefore he has a good level of manipulation of the learning application and a good strategy to achieve his goal, but if learner performs the same task and the values returned by the two HMMs are too close we concluded that he has a low level of manipulation of the application than the first learner, so this learner presents problems of handling the application or a bad strategy to achieve his goal.

Another result that can be concluded from this work, according to the following graph which presents the rate of use of each area for the realization of all tasks.

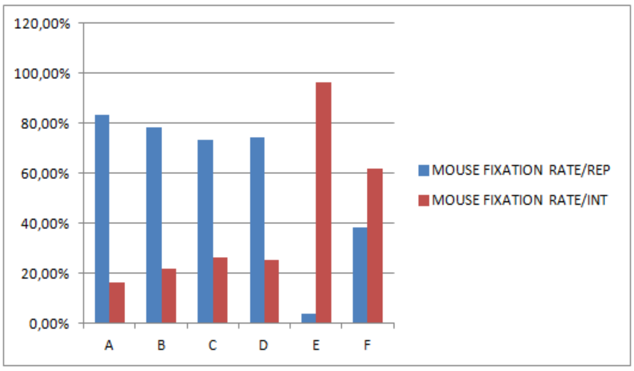

Fig 6. Mouse fixations rate in each AOI for each task.

In fact, figure Fig. 6, shows that the realization of the task INT attracts many the mouse cursor of the user on the area E but for the task REP, mouse cursor will be attracted to areas A and B. This results, brings us to conclude that the mouse movements are guided by the objective of the learner task to well-defined areas of the interface. This finding is in accordance with the work of Yarbus [19] who found that the gaze path of a subject is dependent of his task and with the work of Chen and al [10] that shows that the gaze path is strongly correlated with the mouse path.

VIII. Conclusions

In this paper we propose a model for learner interaction process with an e-learning web-based application. The proposed model has good ability to infer the task performed by the learner based on mouse path and using a Hidden Markov Model theory. The inference technique is based on the greater likelihood probability value generated by each HMMs of the model. The model can give indications about the learner level of interaction with the application; those indications can help teachers to identify problems faced by the learners and provides application developers, clues of the usability level of their product.

In this work we show that mouse movements are guided by the objective of the task and the frequency of handling areas of interest is a strong indicator of the task performed.

The inference of the task can serve as a support for assistive technology. In fact, have clues of the strategy by which a disabled user interacts with web-based application like e-learning, can improve the accessibility of this kind of users and to provide them much more help in the "disabled human machine" interaction process.

Despite their power and their wide use in various fields, HMMs have some problems like a choice of initial parameters of the model that may influence the effectiveness of the model, and also the training set which must be large and containing much more sequence learning well chosen. The enhancement of these elements in proposed model can significantly improve it.

References

[1] A. Bulling, J.A. Ward, H. Gellersen, and G. Tröster, "*Eye movement analysis for activity recognition using electrooculography*", IEEE Transactions on Pattern Analysis and Machine Intelligence archive, vol. 33, Issue 4, pp. 741-753, April 2011.

[2] A. Bulling, J.A. Ward, H. Gellersen, and G. Tröster, "*Robust Recognition of Reading Activity in Transit Using Wearable Electrooculography*," Proc. Sixth Int'l Conf. Pervasive Computing, pp. 19-37, 2008.

[3] J. H. Goldberg, M. J. Stimson, M. Lewenstein, N. Scot, and A. M. Wichansky, *"Eye tracking in web search tasks: Design implications*", Proc.02 of the Eye Tracking Research and Applications Symposium, pp. 51-58, 2002.

[4] A. Dufresne, F. Courtemanche, S. Prom Tep, S. Senecal, "*Physiological measures, eye tracking and task analysis to track user reactions in user generated content*", Proc. 7th International Conference on Methods and Techniques in Behavioral Research, p. 218, August 2010.

[5] Q. Guo, E. Agichtein, "*Exploring mouse movements for inferring query intent*". Proc. 31st international ACM SIGIR conference on Research and development in information retrieval, pp. 707-708, July 2008.

[6] J. Huang, R. W. White, and S. Dumais, "*No clicks, No problem: Using Cursor Movements to Understand and Improve Search*" Proc. SIGCHI Conference on Human Factors in Computing Systems, pp.1225-1234, 2011.

[7] J. Huang, R. W. White, G. Buscher, and K. Wang, "*Improving searcher models using mouse cursor activity*", Proc. the 35th international ACM SIGIR conference on Research and development in information retrieval, pp. 195-204, 2012.

[8] S. Cetintas, L. Si,Y. P. Xin,C. Hord, and D. Zhang, "*Learning to Identify Students'Off-task Behavior in Intelligent Tutoring Systems*", Proc. the 14th International Conference on Artificial Intelligence in Education, pp. 701-703, 2009.

[9] A. H. Abolhassani and J. J. Clark, "*Visual Task Inference Using Hidden Markov Models*", Proc. IJCAI'11 Proceedings of the Twenty-Second international joint conference on Artificial Intelligence, vol. 2, pp. 1678-1683, 2011.

[10] M.C. Chen, J.R. Anderson, and M.H. Sohn, "*What can a mouse cursor tell us more?: correlation of eye/mouse movements on web browsing*", CHI '01 Extended Abstracts on Human Factors in Computing Systems, pp. 281-282, 2001.

[11] L. R. Rabiner and B. H. Juang, "*An introduction to hidden Markov models*", IEEE ASSP Mag., vol. 3, pp.4 -16, 1986.

[12] L. R. Rabiner, "*A tutorial on hidden Markov models and selected applications in speech recognition*", Proc. IEEE, vol. 77, pp. 257 -286, 1989.

[13] A. Voßkühler, V. Nordmeier, L. Kuchinke and A.M. Jacobs, "*OGAMA (Open Gaze and Mouse Analyzer): Open source software designed to analyze eye and mouse movements in slideshow study designs*", Behavior Research Methods, pp. 1150-1162, 2008.

[14] Y. Ohmori, D. Iibuchi,I. Horie, and K. Itoh, "*Understanding learning effect using MBC*",Analysis of mouse behaviour. Conference on IT Education, pp. 51–54, 2006.

[15] P. Brusilovsky, and C. Peylo, "*Adaptive and intelligent Web-based educational systems*", Int. J. Artificial Intelligence in Education, vol.13, no.2-4, pp.159–172, 2003.

[16] J. B. Freeman, and N. Ambady, "*MouseTracker: Software for studying real-time mental processing using a computer mouse-tracking method*". Behavior Research Methods, 42 (1), pp. 226–241, 2010.

[17] M. Zushi, Y. Miyazaki, and K. Norizuki, "*Web application for recording learners' mouse trajectories and retrieving their study logs for data analysis*". Knowledge Management & E-Learning: An International Journal, 4(1), pp 37–50, 2012.

[18] G.D. Chen, C.C. Liu, K.L. Ou, and M.S. Lin, "*Web learning portfolios: A tool for supporting performance awareness*", Innovations in Education and Training International, vol.38, no.1, pp.19–30, 2000.

[19] A L. Yarbus, "*Eye Movements and Vision*". New York: Plenum Press; 1967.

[20] L. A. Granka, T. Joachims and G. Gay, "*Eye-tracking analysis of user behavior in WWW search*", Proc. the 27th annual international ACM SIGIR conference on Research and development in information retrieval, pp. 25-29,July 2004.

[21] M. Morita and Y. Shinoda, "*Information filtering based on user behavior analysis and best match text retrieval*", Proc. the 17th annual international ACM SIGIR conference on Research and development in information retrieval, pp. 272-281, July 1994.

[22] M. Perkowitz, M. Philipose, D. J. Patterson and K. Fishkin, "*Mining Models of Human Activities from the Web*", Proc. The 13th Int'l World Wide Web Conf. (WWW 2004), ACM Press, pp. 573–582, 2004.

[23] A. Poole and L. J. Ball, "*Eye tracking in human-computer interaction and usability research: current status and future prospects*", In C. Ghaoui (ed.), Encyclopedia of human computer interaction. Idea Group, Pennsylvania, pp. 211-219, 2005.

[24] F. Mueller and A. Lockerd. "*Cheese: tracking mouse movement activity on websites, a tool for user modeling*". Proc. CHI, pp. 279-280, 2001.

[25] D. Netherton, and W. Deal, "Assistive technology in the classroom", Technology Teacher, 66 (1), pp. 10 -15, 2006.

[26] R. J. K. Jacob, "*What you look at is what you get: Eye movement-based interaction techniques*", Proc. CHI '90, pp. 11–18, 1990.

[27] A. Struijk, "*Tongue-computer interface for disabled people*", Proc. of 6th Int. Conference on Disability, Virtual Reality and Associate Technologies, 2006.

[28] R. J. K. Jacob, "*Eye movement-based human-computer interaction techniques: Toward non-command interfaces*". In Advances in Human-Computer Interaction, pp. 151-190, 1993.