Introducing an Information System for Successful Support of Selective Attention in Online Courses

Martin Ebner, Josef Wachtler, and Andreas Holzinger

Institute for Information System and Computer Media
Graz, University of Technology
martin.ebner@tugraz.at

Abstract. Human learning processes are strongly depending on attention of each single learner. Due to this fact any measurement helping to increase students' attention is from high importance. Till now there are some developments called Audience-Response-Systems only available for face-to-face education even for masses. In this publication we introduce a web-based information system which is also usable for online-systems. Students' attention will be conserved based on different interaction forms during the live stream of a lecture. The evaluation pointed out that the system helps to enlarge the attention of each single participant.

1 Introduction

Students are confronted with an increasing quantity of information: masses of shapes, colors, and text. However, they can process only a limited amount of this information at one moment [1]. Most of the perceived information is filtered out centrally [2]. It has been shown that a mechanism known as selective attention is the most crucial resource for human learning [3]. Consequently, to manage this attention enhances both behavioral and neuronal performance [4]. Further very important parts of a high-quality learning process especially of an online course are the interaction and the communication between all groups of participants [5]. This means that all sorts and directions of communication are key features for an increased attention and a valuable base to reach the course-goals.

Due to the mentioned facts attention is perhaps the scarest resource within a teaching process, it is of vital importance to think about possibilities on how to control the attention of the students. One possibility in the field of enlargement of the attention with synchronous communication includes so-called Audience-Response-Systems (ARS), which are systems enabling the lecturer to present questions to students during the lecture in a standard classroom situation [6]. These questions could be answered by the students with a special handset and finally the answers are shown to the lecturer so that he can adapt the rest of the lecture according to these results. Many studies reported that an ARS has the power to enhance the attention and the participation of the students [7]. So for instance [8] compared an ARS with other classroom communication methods

C. Stephanidis and M. Antona (Eds.): UAHCI/HCII 2013, Part III, LNCS 8011, pp. 153–162, 2013.

(e.g. answering questions by hand-raising). They noticed that questions presented by an ARS reached the highest formal participation (100%). In addition a further study regarding the advantages and disadvantages of the usage of an ARS performed by [9] claims that an ARS makes a long lecture more interesting and improves the students retention. Furthermore the provided feedback and analysis enhance the learning process for both students and lecturers. However an ARS should not be overused because it could unnecessary slow down the lecture. These facts are also confirmed by [10]. Due to the mentioned studies and facts we like to bring the idea to a complete on-line courses.

2 Theoretical Background

Let us first explain why we think our work is important. When we recognize a physical object (e.g. color, shape, etc.), or when we listen to a speech: it is a measurable information process [11]; yet, this process is heavily influenced by what is called selective attention [2]. Originally, this term goes back to [12] as he proposes a model of the human information processing system [13] and has been further developed and adopted by the three-memory system by [14]: The environmental information is processed by sensory processing registers (STSS in Figure 1) in the various physical modalities (visual, auditory, haptic, gustatoric, olfactoric) and further entered into the so called short-term store (working memory in Figure 1). The information remains temporarily in this working memory, the length of stay depending heavily on control processes (attentional resources in Figure 1). While the information remaining within the working memory, it may be transferred into the long-term memory, heavily dependent on the attentional resources. The model is one of the most used models and very important for the explanation of human learning processes [15], [16], [17] and is still in use today, consequently serves also as the theoretical basis and model for our work (see Figure 1):

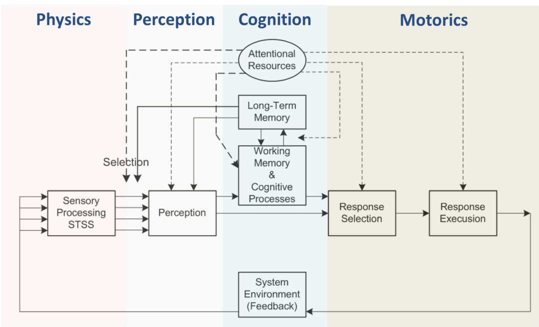

Fig. 1. This model, based on the famous three-storage model by [14] and described by [18] is still the most used and shows the powerful influence of attention to all human information processing [Image taken from [19]]

3 Study

The developed web-application (see Section 4) is used to sustain and enlarge the attention of the attendees of the live broadcasting of the lecture *Cleanroom Technology*[1] which is offered by the *Life Long Learning* department at Graz, University of Technology. This lecture presents the cleanroom technology as a core technology with its influencing factors and the basics of the most important production technologies from a scientific point of view to offer a specialist training to the attendees. This is done due to the fact that cleanroom technology is required in many fields of the economy nowadays.

The broadcasting on the web of this lecture is provided by the *E-Presence Server*[2] and the developed web-application supports this live-stream. Until now five units of the lecture took place and each unit had a length of two hours. It was observed that every session was followed by seven to twelve attendees.

The first unit of the lecture showed that the calculated degree of attention (see Section 4.2) was very low (less than 40%). A reason for that might be that the content of the lecture is very difficult and that the event was not paused. Furthermore most of the asked questions were also very difficult so that the attendees needed a long time to answer them and so the degree of attention decreased.

Due to the problems of the first unit the lecturer was advised to present the contents a little bit slower and in addition he stopped his presentation when a question was asked. With this improvements the degree of attention became better during the second and third unit of the lecture.

Finally the calculation of the degree of attention of the fourth and fifth session of the lecture delivered a high and stable value. This means that every attendee reached a degree of attention of at least 75%.

4 Implementation

This Section explains the main functionalities and shows some details of the implementation of the developed web-application. As mentioned above it provides different methods of synchronous interaction and communication to support any kind of web-content (e.g. the broadcasting of lectures on the web) during events at fixed time. This is done to sustain and enlarge the attention of the attendees [10,5].

Figure 2 presents the basic architecture of the application on a high level of abstraction. So it should be possible to obtain an overview of the different components.

It can be seen that a web-server (e.g. the *Apache HTTP-Server*) and a database-server such as *MySQL* are forming the base. These two servers are

[1] `http://portal.tugraz.at/portal/page/portal/TU_Graz/Studium_Lehre/Life_Long_Learning/ULG%20Reinraumtechnik`

[2] `http://curry.tugraz.at/`

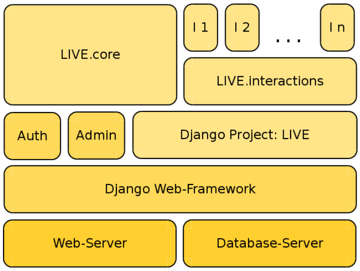

Fig. 2. Basic Architecture of the Web-Application

responsible for making the application accessible over the web and for storing all pieces of data.

To provide a consistent abstraction for the implementation the mentioned servers are encapsulated by the *Django-Web-Framework*. This popular framework for web-development uses the programming-language *Python* and follows the *model-view-controller* architectural pattern. Furthermore it offers a wide range of useful components, for example, a mighty database-API, a template-system, or some basic security mechanisms. In addition the *Auth*-package of *Django* supports the implementation of the user-management and the package named *Admin* offers an interface to build an administration-tool. [20,21]

The definition of the Django-project called *LIVE*[3] forms together with the mentioned packages the base for the actual implementation of the web-application. So the package named *LIVE.core* contains the basic components such as the management of the users or the program-logic to control an event in all phases of its life cycle (see Section 4.1).

Furthermore there exists a package called *LIVE.interactions* which offers a consistent interface for the implementation of the different methods of interaction and a mechanism to control their correct run during an event (see Section 4.2). Finally Section 4.3 describes the interaction-methods which implementations are based on the mentioned interface.

4.1 The Core-Package

As mentioned above this package contains the implementation of all basic functionalities of the web-application. So it provides the management of the user-accounts and of the events. In addition it offers an administration interface.

Due to the fact that only registered and authenticated users should be allowed to use *LIVE* a powerful user-management is implemented with the help of the *Auth*-package of *Django*. This implementation defines three types of users:

1. **Normal users** are allowed to edit their personal data and to join events. During an event they are also able to use the offered interaction-methods.

[3] Short for ***L****IVE* ***I****nteraction in* ***V****irtual learning* ***E****nvironments*.

2. **Users with lecturer-privileges** are additionally allowed to create, edit, start, and stop events.
3. **Administrators** have also the possibility to access the administration-tool.

The probably most important part of the package *LIVE.core* is the event-management. It offers the possibility to control an event in all phases of its life cycle.

At the beginning a user with lecturer-privileges has to create the event. So a suitable dialog asks for the title, for the scheduled begin and end, and for the URL pointing to the web-content which should be supported with interactions. Furthermore the offered interaction-methods could be selected. The dialog stores the collected data in the database with appropriate models.

Now the user who created the event can start it to enable the interaction and communication. For that the real begin is set in the model of the event. After starting it a user-interface for the lecturer is build dynamically (see Figure 3). It shows (1) some information about the event and offers the possibility to stop it on the top left side and on the right (2) the calculated degree of attention of all attendees is presented. Below there is a area for occurring interactions (3) and some control-elements to invoke interactions (4).

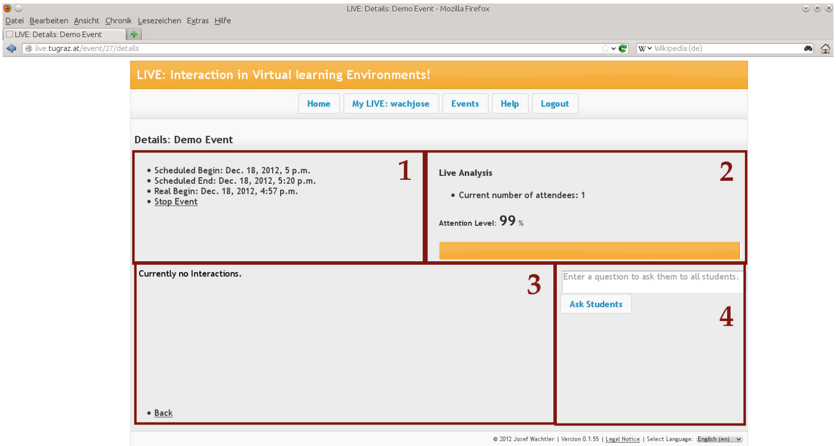

Fig. 3. The User-Interface for the Lecturer during an Event

After the start of an event users are able to join it. Internally a model is used to connect a user to an event. A joined user is immediately redirected to a special user-interface which is shown in Figure 4. It can be seen that the supported web-content fills the most space of the user-interface (1). The sidebar of *LIVE* on the left shows some general information about the event with an ample which represents the current degree of attention (2). Below there is some free space for occurring interactions (3) and there are some control-elements to leave the event or to start different interactions (4).

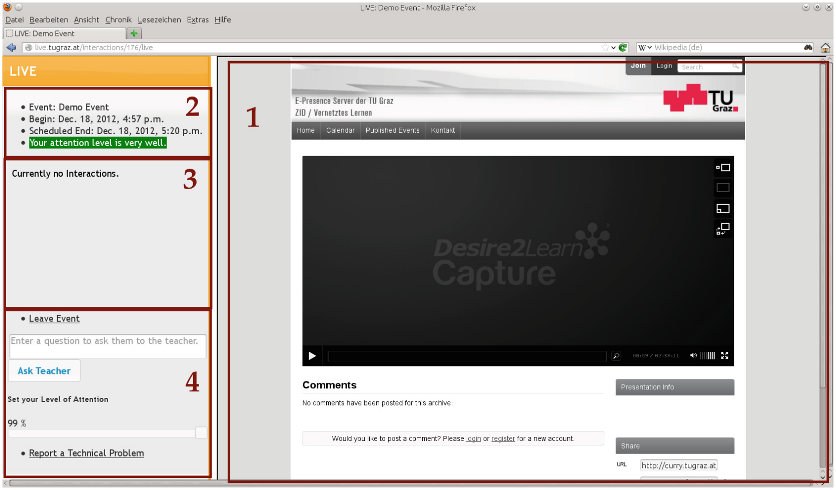

Fig. 4. The User-Interface for the Students during an Event

During an event the interaction-methods are handled and the current degree of attention is calculated. See Sections 4.2 and 4.3 for more details.

At the end of the event its lecturer has to stop it. Internally this is done by setting the real end in the model. Due to that the attendees are redirected to a dialog which asks them for their estimated average degree of attention during the event. After that an analysis of the degree of the attention is computed and presented to both the lecturer and the attendees.

4.2 The Interaction-Interface

The interaction-interface located in the package *LIVE.interactions* offers a consistent interface for both the implementation of the interaction-methods and the *LIVE.core*-package for accessing them. In addition a calculation of the degree of the attention of the attendees is provided.

To implement an interaction-method a *Django*-app has to be created as a sub-package in *LIVE.interactions*. This app additionally requires a module named *settings* which should contain at least the name and a short description of the interaction-method. Furthermore it should define its type by choosing one of the following:

1. **Automatically started**
   The interactions of an interaction-method of this type are automatically presented to the attendees in a random way. For that the *settings*-module has to provide a number which defines how often the interactions should appear per hour.

2. **Started by the attendees**
   This type allows the implementation of interaction-methods which could be started by the attendees. Due to that the *settings*-module should contain the path to a template to include the necessary control-elements in the user-interface.
3. **Invoked by the lecturer**
   With interaction-methods of this type the lecturer is able to invoke them. Again it is required that the *settings*-module offers a template containing the corresponding control-elements.

In addition to these possibilities of invocation there are three different so called interaction-models to display an interaction on the screen of an user. The first one is responsible for presenting an interaction to an attendee. In contrast to that there exists also a model to show an interaction to the lecturer. Finally a further model is present to pop up an alert on the top of the user-interface of the lecturer. All mentioned interaction-models contains the date and time of both the start and the response.

The package *LIVE.core* asks every five seconds through an interface of *LIVE.interactions* if an interaction-model has reached its start-time. If yes a special view of the interaction-method which provides the content of its interaction is called to display them. Within this view everything which is possible with current web-technologies is allowed.

The package *LIVE.interactions* offers a further major functionality of the web-application namely the calculation of the degree of the attention during an event. As mentioned above the calculation is invoked by *LIVE.core* and for that a class is provided. This class asks every interaction-method to compute the degree of attention within its context as a value between 0 and 100 percent. For that every interaction-method has to implement an interface to provide its way of computation. Finally the complete degree of attention is computed by forming the mean over the results of the interaction-methods.

Generally every interaction-method is allowed to compute its degree of attention in its own way but due to fact that most of the interaction-methods are evaluating the reaction-time of its interactions given by the interaction-models a corresponding function is provided. Figure 5 shows this way of computing. It can be seen that two parameters are required. The first one (*SUCCESS_UNTIL*) defines how many seconds after the start an attention of 100% could be reached and the other one (*FAILED_AFTER*) states after which period only 0% of attention are reachable. Between this two points the degree of attention is decreasing in a linear way. This means that the degree of attention is decreasing if the attendees are reacting slower. To find meaningful values for these parameters several test-runs under real conditions (e.g. as a support of the live-streaming of the lecture *Societal Aspects of Information Technology* at Graz, University of Technology) were performed. After each run the parameters were adopted to the observed reaction-times and with the fourth run the values became stable.

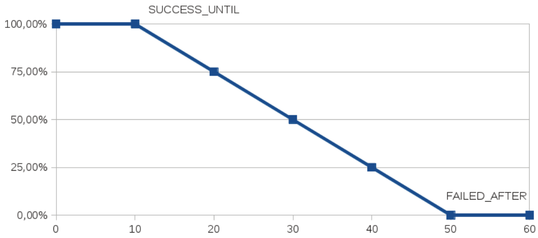

Fig. 5. The calculation of the degree of attention based on the reaction-time of its interaction

4.3 The Interaction-Methods

Based on the interface presented in the previous section six different interaction-methods were implemented. At first there are two automatically invoked methods.

So one method of interaction asks some very simple and general questions to the attendees which should be answered by clicking a button. This happens in a random and automatic way ten times per hour. Furthermore the degree of attention is computed with the described reaction-based method.

In comparison to that the second automatic interaction-method works very similar. The only difference is the usage of a captcha[4] instead of the questions.

Additionally there is an interaction-method which could be invoked by the lecturer to ask a question to the attendees. For that a template which is integrated in the user-interface of the lecturer provides a text-box and a button to submit the question. With the help of the interaction-model the question plus a dialog to answer it is presented to the attendees. If the question is answered the lecturer receives them through an interaction-model. Finally the lecturer could also answer his question as a reference. Again the calculation of the degree of attention is done by evaluating the reaction timespan of the asked attendees.

Similar to that the first attendee-started interaction-method offers the opposite asking-direction. This means that the attendees could ask a question to the lecturer.

A further attendee-invoked method offers the possibility to set the degree of attention by hand through a slider. A template includes them in the user-interface and if it is moved the new value is saved with a suitable model. In contrast to the previous presented interaction-methods the calculation of the degree of attention is based on the values set by the attendees.

Finally there exists a dialog to report a technical problem to the lecturer through an attendee-started interaction-method. So a template embeds a link to the dialog in the user-interface. After a problem-report is submitted the alert-interaction-model is used to present the report to the lecturer. Due to the fact that this interaction-method is not used regularly it is not part of the computation of the degree of the attention.

[4] Short for *Completely Automated Public Turing Test to Tell Computers and Humans Apart*.

5 Discussion

To gain more information how the observed degrees of attention were generated (see Section 3) the attendees were asked to provide a feedback. Most of this feedback addresses two main points. The first one claims that questions which covers the content of the lecture are more useful for the attention than other general questions. Furthermore the second issue which was reported by the attendees states that they feel very uncomfortable with their computed degree of attention because it was very low.

Due to the reported point that content-related questions are more favored than other sorts of questions the interaction-method which presents a captcha to the attendees was deactivated after the first unit of the lecture. In addition more questions which addresses the content of the lecture were asked and the lecturer paused the presentation to give the attendees a better possibility to answer the questions more concentrated. So the questions looks better integrated in the lecture and due to that the attendees feel much more satisfied.

The second reported point claims that the computed degree of attention was much lower than the estimation of the attendees. So this leads to a high disaffection about this value. Based on this feedback the mentioned pauses around the content-related questions were introduced. Furthermore the lecturer was advised to present his lecture a little bit slower. As a consequence the computed degree of attention grows and becomes stable on a high level during the following units of the lecture (see Section 3).

6 Conclusion

This document presents a web-application which provides different methods of interaction in real-time to support any kind of web-content and which offers a calculation of the degree of attention of the attendees. This is done to reach a sustainment and enlargement of the attention (see Section 4).

In addition some observations of the usage of the developed web-application as a support for the broadcasting of a lecture on the web are shown in Section 3. The feedback of the attendees of this lecture is discussed in Section 5.

Based on that it is planed as further work to do more research to better understand in which form the interactions disturb or support the lecture so that a higher enlargement of the attention could be reached. Furthermore the web-application should receive the functionalities to support static web-content (e.g. the recording of a lecture) too.

References

1. Shiffrin, R.M., Gardner, G.T.: Visual processing capacity and attentional control. Journal of Experimental Psychology 93(1), 72–82 (1972)
2. Moran, J., Desimone, R.: Selective attention gates visual processing in the extrastriate cortex. Science 229, 782–784 (1985)

3. Heinze, H.J., Mangun, G.R., Burchert, W., Hinrichs, H., Scholz, M., Münte, T.F., Gös, A., Scherg, M., Johannes, S., Hundeshagen, H., Gazzaniga, M.S., Hillyard, S.A.: Combined spatial and temporal imaging of brain activity during visual selective attention in humans. Nature 372, 543–546 (1994)
4. Spitzer, H., Desimone, R., Moran, J.: Increased attention enhances both behavioral and neuronal performance. Science 240, 338–340 (1988)
5. Carr-Chellman, A., Duchastel, P.: The ideal online course. British Journal of Educational Technology 31(3), 229–241 (2000)
6. Tobin, B.: Audience response systems, stanford university school of medicine (2005), http://med.stanford.edu/irt/edtech/contacts/documents/2005-11_AAMC_tobin_audience_response_systems.pdf (accessed October 9, 2012)
7. Ebner, M.: Introducing live microblogging: how single presentations can be enhanced by the mass. Journal of Research in Innovative Teaching 2(1), 91–100 (2009)
8. Stowell, J.R., Nelson, J.M.: Benefits of electronic audience response systems on student participation, learning, and emotion. Teaching of Psychology 34(4), 253–258 (2007)
9. Silliman, S.E., Mcwilliams, L.: Observations on benefits/limitations of an audience response system. In: Proceedings of the American Society for Engineering Education Annual Conference, Salt Lake City, Utah, June 20-23 (2004)
10. Latessa, R., Mouw, D.: Use of an audience response system to augment interactive learning. Family Medicine 37(1), 12–14 (2005)
11. Yu, Y., Crumiller, M., Knight, B., Kaplan, E.: Estimating the amount of information carried by a neuronal population. Frontiers in Computational Neuroscience 4(10) (2010)
12. Broadbent, D.E.: A mechanical model for human attention and immediate memory. Psychological Review 64(3), 205–215 (1957)
13. Broadbent, D.E.: Perception and communication (1958)
14. Atkinson, R.C., Shiffrin, R.M.: The control processes of short-term memory. Institute for Mathematical Studies in the Social Sciences, Stanford University (1971)
15. Moreno, R., Mayer, R.E.: Cognitive principles of multimedia learning: The role of modality and contiguity. Journal of Educational Psychology 91(2), 358 (1999)
16. Holzinger, A.: Multimedia basics. Learning, cognitive fundamentals of multimedial information systems, vol. 2. Laxmi, New Delhi (2002)
17. Holzinger, A., Kickmeier-Rust, M., Albert, D.: Dynamic media in computer science education; content complexity and learning performance: Is less more. Educational Technology & Society 11(1), 279–290 (2008)
18. Wickens, C., Lee, J., Liu, Y., Gordon-Becker, S.: Introduction to Human Factors Engineering, 2nd edn. Prentice-Hall, Upper Saddle River (2004)
19. Holzinger, A.: Biomedical Informatics: Computational Sciences meets Life Sciences. BoD, Norderstedt (2012)
20. Django Team: Django documentation, https://docs.djangoproject.com/en/1.4/
21. Holovaty, A., Kaplan-Moss, J.: The django book, http://www.djangobook.com/