Scheduling Interactions in Learning Videos

A State Machine Based Algorithm

https://doi.org/10.3991/ijai.v1i1.10995

Josef Wachtler (✉), Martin Ebner
Graz University of Technology, Graz, Austria
josef.wachtler@tugraz.at

Abstract—Based on the currently developing trend of so called Massive Open Online Courses it is obvious that learning videos are more in use nowadays. This is some kind of comeback because due to the maxim "TV is easy, book is hard" [1][2] videos were used rarely for teaching. A further reason for this rare usage is that it is widely known that a key factor for human learning is a mechanism called selective attention [3][4]. This suggests that managing this attention is from high importance. Such a management could be achieved by providing different forms of interaction and communication in all directions. It has been shown that interaction and communication is crucial for the learning process [6]. Because of these remarks this research study introduces an algorithm which schedules interactions in learning videos and live broadcastings. The algorithm is implemented by a web application and it is based on the concept of a state machine. Finally, the evaluation of the algorithm points out that it is generally working after the improvement of some drawbacks regarding the distribution of interactions in the video.

Keywords—Planning, interactivity, learning videos, algorithm, states

1 Introduction

Due to the lately evolved trend of so called MOOCs (short for Massive Open Online Courses) learning videos are currently giving some kind of comeback [1]. This means that videos were considered ineffective for learning purposes. Such a statement was motivated by the commonly accepted maxim "TV is easy and book is hard" [2]. Based on this maxim it is possible to conclude that it is crucial for the learning success that a managed attention of the students is achieved.

Furthermore, students are confronted with a growing number of texts, colors, shapes and figures. Most of them is filtered out centrally by a mechanism known as selective attention [3][4]. This system is considered as a major influencing factor of human learning because it enhances both, behavioral and neuronal performance [5]. As a consequence it seems to be obvious that managing this attention is from high importance.

In addition, this importance is pointed out by Carr-Chellman and Duchastel [6] with the observation that interaction and communication are key factors for students' attention. The interaction and the communication should happen in many different forms and in any directions. On the one hand this means that in addition to face-to-face communication there should be other means like e-mail, forums, chatrooms or something similar. On the other hand the communication should not only happen from the teacher to the students and vice versa, it should also be possible for the students to interact with the content itself.

The application of interactive components to learning videos also provides different possibilities of learning analytics to both, the teachers and researchers [7]. This means that interactions could be used to evaluate the understanding of the students. In addition the behavior of the students could be analyzed by evaluating general properties like the position of the interaction in the video or the reaction times to the interactions.

Based on the presented statements and studies regarding the benefits of interactivity for the attention of the students and therefore their increased learning success, it seems to be obvious to apply some kind of interactivity to learning videos. This is supported by the possibilities of learning analytics with benefits for teachers and researchers. For that a web application is developed which was first introduced by Ebner et al. [8]. Based on this platform some studies were performed. These studies pointed out that the timing of the interactions in the video is important (see Section 2). Because of that this research study introduces a scheduling algorithm for interactions in videos and evaluates its functionalities. Shortly the investigated research question is: "How to place different forms of interactivity in videos in automatic and planned ways?"

As mentioned above some related work is shown at first by presenting several studies about the developed web application. Furthermore, some studies evaluating other web applications for interactive videos are shown. After that the methodologies are explained by Section 3 and this is followed by a detailed description of the developed algorithm (see Section 4). The section named Evaluation shows the performance of the algorithm at several usages at lectures at Graz University of Technology and after that its capabilities are discussed (see Sections 5 and 6). Finally, an outlook is given and the outcomes of the research work are summed up.

2 Related Work

As mentioned above a web application is developed which implements the algorithm presented by this work. This web application was evaluated by several studies.

One study tries to find a basic recommendation of how to place interactions in learning videos so that the success rates of the results to the questions asked by the interactions are maximized [9]. For that the web application was used at a large lecture at Graz University of Technology to present the recordings of the classroom sessions to the students. The evaluation was done by asking subjective questions to students and by analyzing the reaction times to the interactions as well as the results of the questions. Based on that some recommendations were formed [9]:

- "The maximum number of interactions per hour is ten."
- "Content related questions are important for the satisfaction of the students."
- "General questions are useful to support the attention between content related questions."
- "The space between content related questions should not be too small (at least three minutes)."

In addition to these recommendations the mentioned study states some hypotheses about the placement of the interactions in the video. Some of these hypotheses were further evaluated by another study [10]. The first hypothesis claims that at the beginning of a video, questions are more likely to be answered wrong. This claim was confirmed by this follow up study. In contrast to this a second hypothesis was not proven to be correct. It claims that the interval lengths between the interactions have a correlation to the correctness. It was observed by the first study [9] using longer videos that the correctness rate is decreasing if the questions are placed too tight. As mentioned this was not confirmed by the later study [10] using shorter videos.

Because this study took place at a school it was possible to compare the results of the class which used the interactive videos with a class taught in conventional manner [10]. By the comparison of the results of the classes it was possible to evaluate the long-term learning success. It was measured that the first class achieved remarkably better results than the second.

The web application also implements an attention-profiling algorithm first introduced by Wachtler and Ebner [11]. This algorithm provides the possibility to fully track the watched parts of the video for each student and furthermore it computes an attention-level which indicates how attentive each watching student was by evaluating the reaction times to the interactions. The mentioned interactions are scheduled by the algorithm presented by this present work.

A further study used this attention-profiling algorithm to monitor the attendance of students at videos and compared this approach with methods used in a classroom [12]. For a better accuracy of the attendance monitoring the results of some questions presented by interactions during the video were part of the monitoring. It was pointed out that this possibility of monitoring the attendance at videos delivers the same accuracy as different methods used in standard classroom situations.

Another possible usage for interactive components in videos is assessment. This scenario was addressed by a study performed at a large lecture at Graz University of Technology [13]. For that exam questions were presented to students during the video. It was communicated to the students that these questions will be part of the final grade and because of that students worked more concentrated and received a better grade point average than before using the interactive videos. Furthermore, it was observed that the required time for the teacher to evaluate the performance of students has been optimized by using an approach with interactive videos.

In contrast to these studies using the developed web application which implements the algorithm introduced by this work, the benefits of interactive videos were also pointed out by several other research studies. One of these studies [14] compared the usage of interactive videos with non-interactive videos for the purpose of learning. For that a group of 36 students from the University of Offenburg were required to learn how to tie four nautical knots by watching videos. None of the students had any previous knowledge on the subject. The students were divided in two groups, one with interactive videos and one without interactivity. They were permitted to watch the videos as often as they needed to learn to tie the knots. To measure the benefits of the interactivity the required time to learn was recorded. It was pointed out that the group with the interactive videos required significantly less time than the group with the non-interactive ones. The differences between the required learning time of the two groups ranges from 66% to 95% of additional time for videos without interactivity. Furthermore, it was pointed out that the offered methods of interactivity were used heavily by all students.

A different study [15] evaluated the usage of in-video quizzes at a Machine-Learning course on Coursera. This course consists of 113 videos with a total length of 19.5 hours. In summary the videos were enriched with 109 in-video quizzes. Out of the 96,195 registered users 41,643 watched at least one video till its end. The study pointed out that 74% of the users who got to see a quiz tried to answer it. A common behavior of the watchers is that they seek forward to the quizzes and after seeing the question they tend to seek backward before answering it. A likely explanation is that the students try to find the content which is required to answer the questions. In the most extreme circumstances students simply jump from quiz to quiz. The correctness rate of the quizzes is quite high because 76% of the students provided a correct answer at the first attempt. Furthermore, it was observed that the drop-out rate is lower at the videos with quizzes in comparison to the ones without.

To address the mentioned forward and backward seeking in the video around the quizzes a further study [16] presents an application which implements a video viewing approach driven by questions. The application divides the video in segments and assigns a question to each segment. Users should navigate through the video by answering the questions. This means that the users are able to skip segments if they are able to answer the question before watching it. The answered questions and watched segments are forming a timeline of the history of the students' progress. This history enables the students to review the watched content. The application additionally encourages the students to rewatch segments which they have not mastered completely. To evaluate the application 18 participants were recruited. They were required to watch two videos in two different settings. The first video was presented to them using the in-video quizzes of Coursera and the second video used the developed application. It was concluded that with the developed application students are more often answering questions and are also more often reviewing the related video segments in comparison to the standard in-video quizzes. Furthermore, it was pointed out that the students remembered the correct answers to the questions significantly better at the question driven video application.

3 Methodology

To develop the algorithm two different methods of (software) development are used. The first one is named “Test Driven Development” (TDD) [17] and the second one is called “Rapid Prototyping” [19]. These two methods are interwoven because the new features created with TDD are brought to the test users very fast according to the principles of Rapid Prototyping.

In general test driven development is based on the strategy that tests for the different units of the software are written before the actual source code of it [17]. These tests are defining the requirements of the software to develop. So, one might assume that this approach is simply a testing strategy. However a more detailed examination of the different parts of TDD shows a different picture [18]:

- **Test:** This phase consists of writing tests for each unit or part of the software. To enable this early writing of tests a clear interface for testing has to be defined. It is obvious that the tests are run after the source code of the software is finished. It is not necessary that the tests as well as the source code are written by the same person. Furthermore, the execution of the tests is completely independent from writing them.
- **Driven:** With TDD the whole software development process is driven forward because the creation of tests and its implementation in the software leads to the fact that only the required parts to pass the tests are implemented. After that a refining of both, tests and source code has to happen. It can be seen that all four general phases of software development (analysis, design, implementation and testing) are covered and supported.
- **Development:** TDD alone is no full software development process. It produces tests and code. Based on this it helps to loop through the mentioned general phases of software development. However it has to be part of a larger process like Rapid Prototyping.

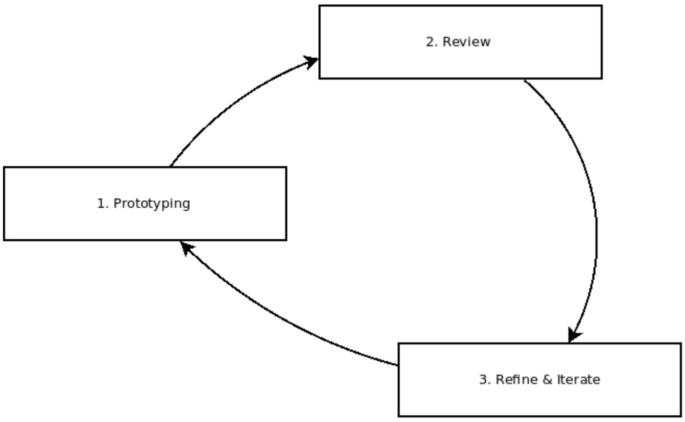


Fig. 1. Three steps of the rapid prototyping loop [19][20]

Rapid Prototyping means that newly developed features are brought to the testing users or even to productive usage very fast and the feedback is used to improve the prototype [19][20]. Now the loop can start again. This basic principle can be seen in Figure 1. In conjunction with TDD this means that the prototype is created by writing tests that cover its general functionalities and based on this the prototype is implemented. After that the users are working with the prototype and with the feedback. The tests of TDD are refined and a new iteration of the prototype will keep the loop running.

As mentioned above TDD and Rapid Prototyping are used to develop the algorithm. This means that the requirements of the algorithm are initially defined by the unit tests and a prototype is developed according to these tests. This prototype is used at lectures at Graz University of Technology and the interactions scheduled by the algorithm at these lectures are analyzed. For that the distribution of interactions for each student is searched for unwanted effects (see Section 5).

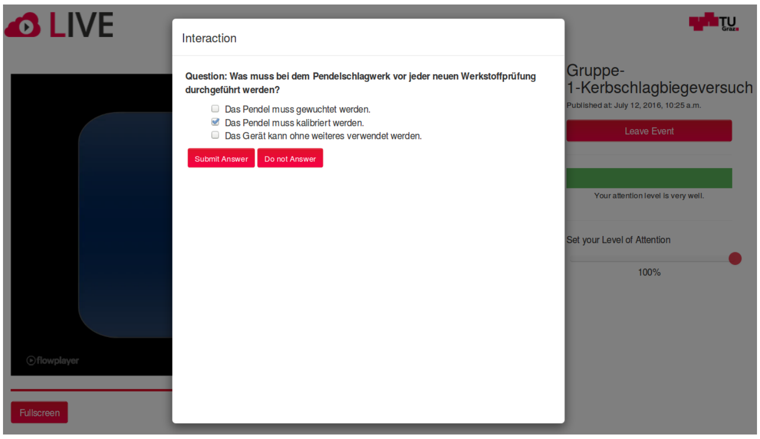
4 Implementation

This section presents the developed scheduling algorithm in a very detailed way. For a better understanding of its purpose the web application which implements the algorithm is presented at first (see Section 4.1). After that an overview of the different components (see Section 4.2) is given and this is followed by the explanation of the underlying models in Section 4.3. Now the state-machines of the interactions are shown (see Section 4.4). This is followed by a description of some helper modules by Section 4.5 and finally all parts of the algorithm are combined together (see Section 4.6).

4.1 The web application

As mentioned above the developed algorithm is implemented by a web application named LIVE (short for Live Interaction in Virtual learning Environments) [8][21]. This web application is implemented in the programming language Python using the Django-Web-Framework. It enables the enrichment of videos and live broadcastings with different types of interactivity. This means that the videos could be interrupted with questions and watching students are also able to invoke some interactions (e.g. asking a question to the teacher). In Figure 2 it can be seen how a video is interrupted by an interaction presenting a multiple choice question. The purpose of these interactions is the support of the students' attention. So the interactivity is used to help the students to stay attentive during the whole video [9].

Furthermore, there are a lot of different analysis features for the teacher. For example a detailed recording of the joined and watched timespans is available [11][21]. With this it is possible to state for each student when she/he watched which part of the video (see Figure 3). In addition, a timeline of the watching history (see Figure 4) of all students is computed.



Interaction

Question: Was muss bei dem Pendelschlagwerk vor jeder neuen Werkstoffprüfung durchgeführt werden?

- ☐ Das Pendel muss gewuchtet werden.
- ☑ Das Pendel muss kalibriert werden.
- ☐ Das Gerät kann ohne weiteres verwendet werden.

Gruppe-1-Kerbschlagbiegeversuch

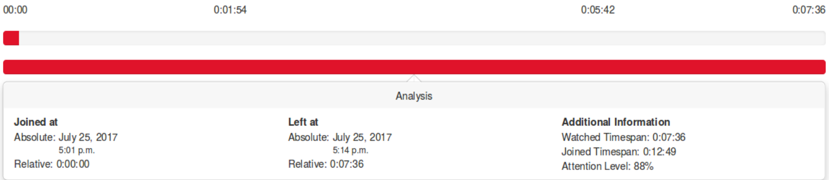
Published at: July 12, 2016, 10:25 a.m.

Your attention level is very well.

Set your Level of Attention

100%

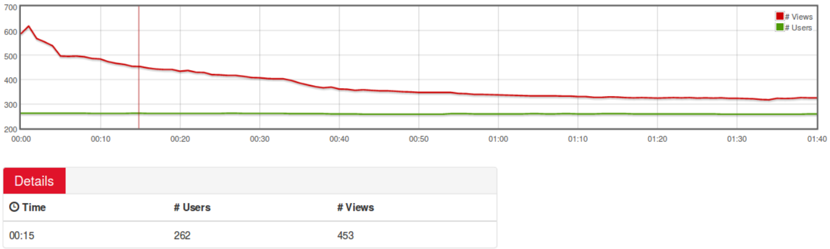
Fig. 2. A video is interrupted by an interaction.



Analysis

Joined at	**Left at**	**Additional Information**
Absolute: July 25, 2017 5:01 p.m.	Absolute: July 25, 2017 5:14 p.m.	Watched Timespan: 0:07:36
Relative: 0:00:00	Relative: 0:07:36	Joined Timespan: 0:12:49
		Attention Level: 88%

Fig. 3. The watching history of a student shows a timeline of the video and marks the watched parts with red bars. On mouse-over additional details are printed.



Time	# Users	# Views
00:15	262	453

Fig. 4. The diagram prints the timeline of the video on the x-axis and the number of watching students on the y-axis. The red line represents the number of views along the timeline and the green line states the number of different users. The vertical red crosshair could be moved along the timeline to show exact values in the "Details" box.

In summary the main functionalities of LIVE are given by the following list [8][10][11][21]:

- Only available for registered and authenticated users
- Different methods of interactivity
  - Automatically asked questions and CAPTCHAs (short for Completely Automated Public Turing test to tell Computers and Humans Apart)

  - Asking questions to the teacher
  - Asking text-based and multiple-choice questions to the students
  - Students can set their level of attention
  - Error reporting to the teacher
  - Analysis features
- Evaluations based on an attention-profiling-algorithm
- Evaluation of the answers of the students to the different questions

All of the mentioned forms of interactivity are encapsulated in so called "interaction methods" [22]. Each interaction method is completely independent from the other interaction methods and provides all functionalities of a single form of interactivity (e.g. asking text-based questions to the students). This includes the creation, the showing and the analysis of this single interaction method. From a technical point of view such an interaction method is a plugin which implements an API provided by the web application.

Each interaction method is of a given type. This interaction type defines the method used for scheduling the interactions as well as the primary targeted audience (students or teacher). Currently there are four types [22]:

- **Automatic:** Interaction methods of this type are called automatically and in a random way a given number of times per hour. This means that neither the teacher nor the students are required to trigger an interaction.
- **Student Triggered:** At this interaction type the student has to start the interaction by himself. For that there are control elements at the right hand side of the video.
- **Teacher Triggered:** While a live broadcasting is running the teacher could start interactions of this type. This is done by using similar control elements.
- **Planned:** The interactions of this type are shown at planned positions in the video. It seems to be obvious that in this case the teacher has to create and plan the interactions before the video is released.

4.2 Overview

The algorithm to schedule interactions in videos consists of several components. In Figure 5 these components are shown.

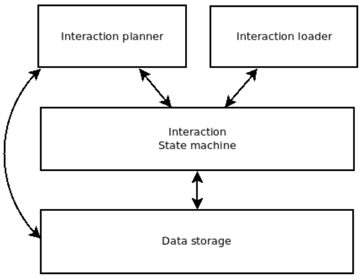


Fig. 5. The components of the algorithm

The underlaying part is the data storage which is a relational database represented by models. It is responsible for saving all the data required to handle the scheduling of the interactions. The models are accessed by the state machines of the interactions during the video. This indicates that the state machines control the main functionalities of the algorithm because depending on the state interactions are shown or discarded. To support the state machines there is the interaction planner and the interaction loader. The first one creates a plan of interactions for each student at her/his first joining to the video. For that the planner has to access the models. The second supporting part is the loader and as indicated by its name it is responsible for showing the interactions to the students when they are scheduled.

4.3 The models

All the data of the algorithm is stored by the models in a database. Figure 6 shows the used models of the algorithm in a simplified version. As mentioned above the interactions are grouped in interaction methods of a given type. This grouping is represented by the models InteractionMethod and InteractionType. The first one holds some metadata of the interaction method and defines whether the video should be paused if an interaction occurs by the attribute pause_on. The second is responsible for indicating the type of the interaction method.

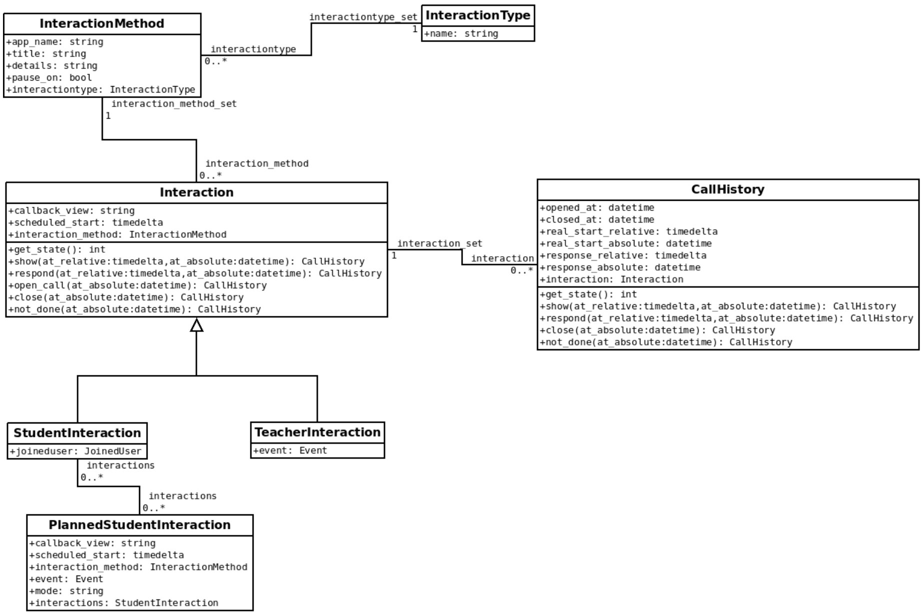

InteractionMethod
+app_name: string
+title: string
+details: string
+pause_on: bool
+interactiontype: InteractionType
interactiontype
0..*
interactiontype_set
1
InteractionType
+name: string
interaction_method_set
1
interaction_method
0..*
Interaction
+callback_view: string
+scheduled_start: timedelta
+interaction_method: InteractionMethod
+get_state(): int
+show(at_relative:timedelta,at_absolute:datetime): CallHistory
+respond(at_relative:timedelta,at_absolute:datetime): CallHistory
+open_call(at_absolute:datetime): CallHistory
+close(at_absolute:datetime): CallHistory
+not_done(at_absolute:datetime): CallHistory
interaction_set
1
interaction
0..*
CallHistory
+opened_at: datetime
+closed_at: datetime
+real_start_relative: timedelta
+real_start_absolute: datetime
+response_relative: timedelta
+response_absolute: datetime
+interaction: Interaction
+get_state(): int
+show(at_relative:timedelta,at_absolute:datetime): CallHistory
+respond(at_relative:timedelta,at_absolute:datetime): CallHistory
+close(at_absolute:datetime): CallHistory
+not_done(at_absolute:datetime): CallHistory
StudentInteraction
+joineduser: JoinedUser
TeacherInteraction
+event: Event
interactions
0..*
interactions
0..*
PlannedStudentInteraction
+callback_view: string
+scheduled_start: timedelta
+interaction_method: InteractionMethod
+event: Event
+mode: string
+interactions: StudentInteraction

Fig. 6. The models of the algorithm in a simplified version

An interaction is represented by the base model Interaction. Because an interaction could be shown to the students or the teacher there are two sub-models. The first is named Student Interaction and it is used for interactions shown to students and the second (Teacher Interaction) represents interactions displayed to the teacher at live broadcastings only. For simplifying reasons the following description of the algorithm is focused on students at videos. This is done because the basic working principle is the same at live broadcastings and for teachers.

The base model saves the name of the view which actually displays the interaction represented by the model. Furthermore, it holds the scheduled starting point of the interaction in the form of a relative position in the video. In addition to these values the sub-model for interactions shown to students holds the user to which the interaction is shown. In a similar way the sub-model for the teacher interactions saves the event of which a teacher is available to receive the interaction.

If the interaction method is of type 4 (planned interactions) there has to be a mechanism to plan the interactions. For that there is the model called Planned Student Interaction which is responsible for saving these planned interactions. It saves the name of the view which is called to display the interaction and the scheduled time of start in a very similar way like the normal interactions. Furthermore, it saves the event to which it belongs and the mode (video or live broadcasting) of the event. Because the PlannedStudentInteraction is translated to a standard Student Interaction for each watching student (see below) there is also a list of these created interactions.

Every time an interaction is shown to a student or to a teacher a new instance of a CallHistory is created. It saves different values of date and time indicating when the interaction is shown, answered or simply closed. In detail the different fields are used in the following way:

- Opened_at: Saves the date and time of the creation of the Call History. This is done by the interaction planner, by the students or the teacher (see below).
- Closed_at: Holds the date and time of closing the Call History. This happens when the interaction is never displayed.
- Real_start_relative: position in the video of the displaying of the interaction
- Real_start_absolute: date and time of the displaying of the interaction
- Response_relative: position in the video of the user's response
- Response_absolute: date and time of the user's response

To manipulate these values the offered methods of the Interaction model are used and they are mostly delegating to the corresponding methods of the Call History after some security checks. The state machine presented by the following section operates mainly on these values.

4.4 The state machine

As indicated above the state machine operates on the interactions. This means that each interaction has a state and based on that state certain actions are performed. Figure 7 visualizes the states and the transitions. When a student joins a video the first time all interactions are created for her/him in the initial state named "open" with a new CallHistory. This is done by the interaction planner (see next section). During the video runs the current position is monitored. If the scheduled starting position of the interaction is reached the state switches to “pending”. Now there are two possibilities: The first one is that the interaction loader (see section below) brings the interaction to its displaying queue but the interaction is never displayed and because of that the state switches to “closed”. This is the case if there is another interaction currently on display in its waiting state. The second possibility is that the loader displays the interaction and this leads to a state switch to “waiting”. Now the interaction is waiting for a response from the student. In the case of a response the interaction switches to the state “done” and in the other case if there is no response it switches to “not_done”. Finally, it could happen that the video is watched more often than once and this leads again to the interaction planner which is creating a new Call History. This causes a reset of the state of all interactions to their initial state “open”.

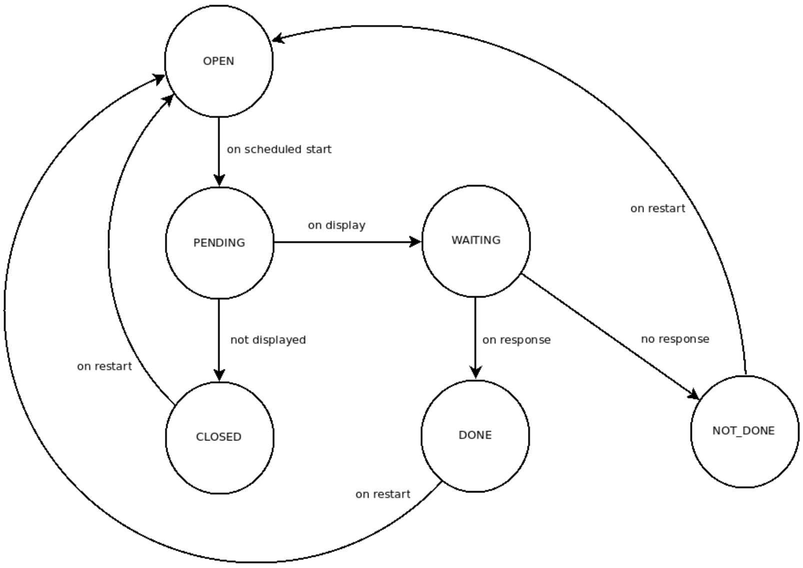


Fig. 7. The different states of an interaction and its transitions

As indicated above the state is defined by the values of the members of the current Call History. Table 1 shows this relation of each state to these values. It can be seen that each state has a different distribution of set (x) fields.

Table 1. The model fields are defining the states.

State	opened_at	closed_at	real_start_ relative	real_start_ absolute	response_ relative	response_ absolute
OPEN	x					
PENDING	x		x			
WAITING	x		x	x		
DONE	x		x	x	x	x
NOT_DONE	x	x	x	x		
CLOSED	x	x				

4.5 Planning and loading interactions

As mentioned above there are two helper modules for the state machine, namely the interaction planner and the interaction loader. The first one is responsible for planning the interactions for each student at their first joining to the video. The second one displays the interaction to the students and handles the response.

The interaction planner is responsible for creating a plan of interactions for each student on her/his first joining to the video. In general this planning happens by creating instances of the Student Interaction model. Depending on the type of the interaction method they are created in a different way.

If the type of the interaction method is "1" (automatic) the interactions of this method are created in a random way. For that each interaction method of this type has to provide a parameter stating how often its interactions should be displayed per hour. With this value a slot length in seconds is computed by dividing the seconds of an hour (3600) by the given value of the parameter. Now an interaction is created for every slot which fits into the length of the video. In each slot the interaction is placed randomly however there is the following restriction. To avoid that interactions are placed too fast after each other at the borders of the slots the first as well as the last ten percent are not used for the interaction of the slot.

The interactions of methods of the types "2" (student triggered) and "3" (teacher triggered) are not created by the interaction planner. This is the case because these interactions are created on an action triggered by the students or by the teacher on the fly during the video runs. This creation has to be done by the interaction methods themselves.

Finally, the type "4" (planned) is covered by the interaction planner. As shown above there is an instance of Planned Student Interaction for each planned interaction. Because these instances are not related to a student the interaction planner creates a Student Interaction for each of them and links them to their corresponding Planned Student Interaction. These interactions are created by the interaction planner before the automatic ones. This is done because the interaction planner tries to leave a space (+/- 10% of the slot length) of no other interactivity around each planned interaction.

A further duty of the interaction planner is to reset the state machine of each interaction on re-joining of a student to the video. This means that a new Call History is created for the interactions of the re-joining student.

Below the complete interaction planner is shown in pseudo code. It can be seen that the creation of interactions happens only at the first joining of a user. The first for-block is responsible for creating the planned interactions and the second one handles the creation of the automatic interactions.

```
if users_first_join() {
  for pi in set(PlannedInteraction) {
    interaction = create StudentInteraction from pi
    pi.interactions.add(interaction)
  }
  for each interaction_method of type automatic {
    slot_length = 3600 / interac-
tion_method.interactions_per_hour
    slots_per_video = video_length / slot_length
    for each slot in slots_per_video {
      do {
      pos = random(slot.begin + 10% of slot_length,
        slot.end - 10% of slot_length)
      }
      while pos is not around a pi
      automatic_interaction = StudentInteraction at pos
    }
  }
}
for each StudentInteraction {
  create a new CallHistory
}
```

During the video runs the interaction loader monitors the position in the video. It reports this position to the state machines of the interactions so that the state machines can switch the state. If an interaction switches to the state "pending" the interaction loader tries to show the interaction to the student by displaying its callback view. Furthermore, the video is paused if the interaction is part of an interaction method requesting this pausing. Once the interaction is shown the state switches to "waiting". In the case that another interaction is on display the pending interaction is not shown and remains pending until the other interaction is finished. If the interaction is currently displayed and therefore in its waiting state the interaction loader waits for a response from the student. If this response finally comes the state machine is informed and because of that, it switches to one of its end-states.

4.6 The algorithm

Now all parts of the algorithm are combined. In Figure 8 it is shown how the algorithm operates with the different components. At first it can be seen that the teacher has to create the interactions which are planned. Strictly this is not part of the algorithm but it is required for better understanding.

When a student joins a video the algorithm starts with the interaction planner. As explained above it creates the interactions with the state machines. For that it independently plans the automatic interactions and uses the templates (Planned Student Interaction) created by the teacher for the planned interactions. Now the interaction loader takes over and feeds the state machines with the current position so that the state switches can happen.

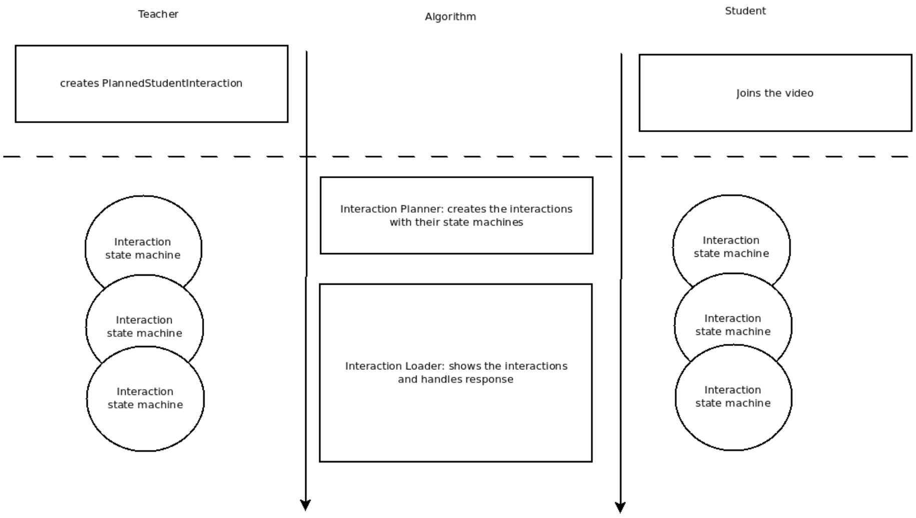

Teacher
Algorithm
Student
creates PlannedStudentInteraction
Joins the video
Interaction Planner: creates the interactions with their state machines
Interaction Loader: shows the interactions and handles response
Interaction state machine
Interaction state machine
Interaction state machine
Interaction state machine
Interaction state machine
Interaction state machine

Fig. 8. The flow of events of the algorithm

From a users' point of view the teacher has to create the planned interactions through forms offered by the interaction methods. After the teacher has released the video to students, they are able to watch it. When they are joining the video for the first time, the interaction planner builds the plan of interactions for each student. While the students are watching, interactions will be displayed to them. In the case of a live broadcasting interactions are also displayed to the teacher. This is handled by the interaction loader in both cases.

5 Evaluation

According to the described methodology tests are written for the different units or parts of the algorithm. If all of the tests are passed successfully the prototype is used in a productive environment. With the gathered results and observations from the productive usage the tests are refined to match new or adapted requirements. This indicates that the correct functionality from a technical point of view is validated by the unit tests and the practical aspects are evaluated by analyzing the results of test usages as well as by the feedback from the users.

The algorithm presented by the previous section is explained in its current version. The following steps of evaluation are performed on earlier versions and the revisions are leading to this current one.

5.1 Setting

The created prototypes were used at different lectures and projects at Graz University of Technology. This included the presentation of recordings of lectures held in classrooms to the students as well as common learning videos. In summary 950 watching students and 93 videos were part of the evaluation. In combination this led to a total number of 7531 different distributions of interactions. To evaluate the distribution of the interactions the following values of the interactions were recorded and analyzed for each student:

- Date and time
- Position in the video
- Timespan since last interaction

5.2 Automatic and random interactions

With the recorded values for each distribution of interactions it is possible to count the number of distributions with unwanted effects. These unwanted effects are mostly defined by feedback of students and could be summarized to the following constrains:

- The interval between interactions should be 10% of the video length or higher.
- However, the same interval should not be longer than 35% of the video length.

To fulfill these constrains it took three iterations of the Rapid Prototyping loop. After each productive usage of the algorithm the distributions of interactions were searched for these unwanted effects. The results could be seen in Table 2.

Table 2. The results of the analysis of the distributions of interactions.

	Number of distributions	Interval length < 10% of video length	Interval length > 35% of video length
Version 1	2167	520 (23.99%)	628 (28.98%)
Version 2	3514	457 (13.01%)	492 (14.00%)
Version 3	1850	0	0

In the first version of the interaction planner, the scheduling of the automatic and random interactions (type 1) was implemented without the explained slots. The interactions were randomly distributed through the length of the video without any restrictions. This requirement was fully covered by the unit tests and after the passed successfully the algorithm was used in a productive environment.

It was observed that with the completely random approach an uneven distribution of interactions could occur [9]. This means that in 29% of the distributions longer phases of no interactivity happened. Furthermore, it is clear that the same effect lead to an agglomeration of interactions in 24% of the cases. These results gathered by analyzing the recorded data were also reported by the watching students. Most of them claimed that an even distribution of interactions through the video is favored. These claims led to the creation of the constrains mentioned above.

Because of that the requirements were refined. With these new requirements the slot mechanism presented above was introduced. Again the prototype was used at a lecture and the distribution of the interactions was analyzed. Now a much better distribution was observed but in 13% of the cases the interactions were placed too narrow. It seems to be obvious that this could happen at the borders of the slots. If in one slot the interaction was placed very near to the end of the slot and at the proceeding slot at the beginning. The same effect led to longer phases of no interactivity in 14% of the cases. This led again to a refinement of the requirements and the tests which were used to develop the current version of the algorithm. Now the constrains are fulfilled by all distributions. Figure 9 visualizes the differences of the three versions by showing three example distributions of interactions. The points are representing the interactions. It can be seen that with each version the distribution becomes more even.

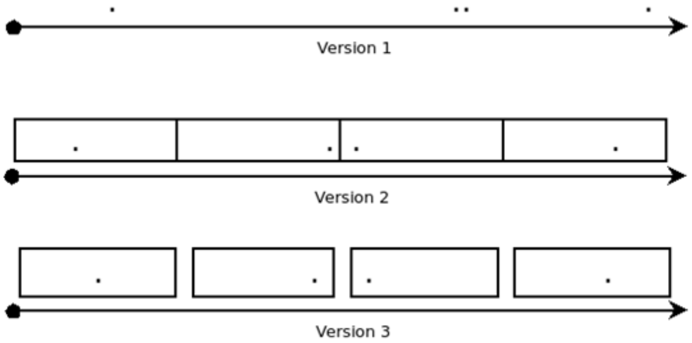


Fig. 9. The different versions of the algorithm schedule the interactions differently.

5.3 Planned nteractions

The first version of the interaction planner created the random interactions before the planned ones. However the usage at a lecture pointed out that it could happen that random interactions are occurring at the same time or very close to planned interactions. Because of that the unit tests were refined. This lead to a change of order and now the planned interactions are created at first. This enables the interaction planner not to place a random interaction too close to a planned interaction.

6 Discussion

As stated above the main drawbacks of the earlier versions of the algorithm are related to the distribution of the interactions along the timeline of the video. It has been shown that with an uncontrolled random distribution it could happen for some students that the interactions are placed to dense. As a consequence this leads to longer phases of no interactivity. This behavior is not appreciated because of two reasons. At first the goal of interactive components is the support of the attention of the students while watching a video. As shown by other studies [9][10] a regular occurrence of interactions is important. The second reason is the feedback of the students [9]. They are also more satisfied with an even and regular distribution of interactions. These mentioned reasons were confirmed by using the algorithm at lectures to show the recordings of the lecture held in class. This means that these recordings are of a considerable length (one hour or more). In comparison to that the usage of the algorithm for shorter videos (10 to 20 minutes) points out that the distribution is not directly related to the success rate of the students [10]. A reason for the discovered differences regarding the distribution of interactions in videos of different lengths might be that in shorter videos the frequency of interactions is typically higher and because of that a more dense setting of interactive components is more accepted. It can be seen that the remarks about the connection between the distribution of the interactive components and the success rate of the students were gathered in the context of different studies. For that the possibilities of learning analytics provided by the application of the interactions to videos were used [7].

7 Outlook

The discovered discrepancy between the longer and shorter videos regarding the distribution of interactions implies that further research is required. It is planned that recordings of lectures should be presented to students in different ways. Each recording should be presented to one group of students in its complete and therefore long version. A second group should watch the recording in smaller parts. The distribution of interactions in the videos of the two groups should be the same. With that it should be possible to gain further insight in the relevance of the distribution of interactions at videos of different lengths.

8 Conclusion

With this work an algorithm to schedule interactive components in learning videos as well as live broadcastings is introduced. The application of interactivity to videos seems to be important because videos were not well accepted for learning purposes [2]. However, learning videos are more in use today [1] due to the increased offering of MOOCs. To support the attention of the watching students providing interactive components in different forms is a widely accepted tool [6].

To address this issue a web application first introduced by Ebner et al. [8] is developed. This web application adds different methods of interactions to videos and live broadcastings. For that the scheduling algorithm presented by this work is used. This algorithm consists of several parts which are explained by Section 4. The basic principle of the algorithm is a state machine for each individual interaction. Depending on its state an interaction is handled differently. This means that such an interaction could be waiting to be displayed, it could be currently visible or it could be already finished.

The algorithm is developed and evaluated using the mechanisms of “Test Driven Development” and “Rapid Prototyping”. This indicates that TDD is used to define the requirements and to validate the correct implementation of these requirements. To evaluate the requirements a prototype which is passing the unit tests of TDD is used in a productive environment. With the experiences and gathered results of the productive usage the requirements are refined and the circle starts again. The evaluation of the algorithm according to this explained principle pointed out that after some iterations the scheduling produces satisfying distributions of the interactions. With a working algorithm the initial research question (see Section 1) is finally answered.

9 References

[1] H. Khalil and M. Ebner, “Interaction possibilities in moocs–how do they actually happen,” in International Conference on Higher Education Development, 2013, pp. 1–24.

[2] G. Salomon, “Television is easy and print is tough: The differential investment of mental effort in learning as a function of perceptions and attributions.” Journal of educational psychology, vol. 76, no. 4, p. 647, 1984. https://doi.org/10.1037//0022-0663.76.4.647

[3] J. Moran and R. Desimone, “Selective attention gates visual processing in the extrastriate cortex,” Science, vol. 229, pp. 782–784, Aug. 1985. https://doi.org/10.1126/science.4023713

[4] R. M. Shiffrin and G. T. Gardner, “Visual processing capacity and attentional control,” Journal of Experimental Psychology, vol. 93, no. 1, pp. 72–82, 4 1972. https://doi.org/10.1037/h0032453

[5] H. Spitzer, R. Desimone, and J. Moran, “Increased attention enhances both behavioral and neuronal performance,” Science, vol. 240, pp. 338–340, Apr. 1988. https://doi.org/10.1126/science.3353728

[6] A. Carr-Chellman and P. Duchastel, “The ideal online course,” British Journal of Educational Technology, vol. 31, no. 3, pp. 229–241, 2000. [Online]. Available: https://doi.org/10.1111/1467-8535.00154

[7] J. Wachtler, M. Khalil, B. Taraghi, and M. Ebner, “On using learning analytics to track the activity of interactive mooc videos,” 2016.

[8] M. Ebner, J. Wachtler, and A. Holzinger, “Introducing an information system for successful support of selective attention in online courses,” in Universal Access in Human-Computer Interaction. Applications and Services for Quality of Life. Springer, 2013, pp. 153–162. https://doi.org/10.1007/978-3-642-39194-1_18

[9] J. Wachtler and M. Ebner, “Impacts of interactions in learning-videos: A subjective and objective analysis,” in EdMedia: World Conference on Educational Media and Technology, vol. 2015, no. 1, 2015, pp. 1642–1650.

[10] J. Wachtler, M. Hubmann, H. Z¨ohrer, and M. Ebner, “An analysis of the use and effect of questions in interactive learning-videos,” Smart Learning Environments, vol. 3, no. 1, p. 13, 2016. https://doi.org/10.1186/s40561-016-0033-3

[11] J. Wachtler and M. Ebner, “Attention profiling algorithm for video-based lectures,” in Learning and Collaboration Technologies. Designing and Developing Novel Learning Experiences. Springer, 2014, pp. 358–367. https://doi.org/10.1007/978-3-319-07482-5_34

[12] J. Wachtler and M. Ebner, “On using interactivity to monitor the attendance of students at learning-videos,” in EdMedia: World Conference on Educational Media and Technology. Association for the Advancement of Computing in Education (AACE), 2017, pp. 356–366.

[13] J. Wachtler, M. Scherz, and M. Ebner, “Increasing learning efficiency and quality of students homework by attendance monitoring and polls at interactive learning videos,” in EdMedia+ Innovate Learning. Association for the Advancement of Computing in Education (AACE), 2018, pp. 1357–1367.

[14] S. Schwan and R. Riempp, "The cognitive benefits of interactive videos: learning to tie nautical knots," Learning and instruction, vol. 14, no. 3, pp. 293–305, 2004. https://doi.org/10.1016/j.learninstruc.2004.06.005

[15] G. Kovacs, "Effects of in-video quizzes on mooc lecture viewing," in Proceedings of the third (2016) ACM conference on Learning@ Scale. ACM, 2016, pp. 31–40. https://doi.org/10.1145/2876034.2876041

[16] G. Kovacs, "Quizcram: A question-driven video studying interface," in Proceedings of the 33rd Annual ACM Conference Extended Abstracts on Human Factors in Computing Systems. ACM, 2015, pp. 133–138. https://doi.org/10.1145/2702613.2726966

[17] K. Beck, "Test-driven development: by example". Addison-Wesley Professional, 2003.

[18] D. Janzen and H. Saiedian, "Test-driven development concepts, taxonomy, and future direction," Computer, vol. 38, no. 9, pp. 43–50, 2005. https://doi.org/10.1109/mc.2005.314

[19] X. Yan and P. Gu, "A review of rapid prototyping technologies and systems," Computer-Aided Design, vol. 28, no. 4, pp. 307–318, 1996. https://doi.org/10.1016/0010-4485(95)00035-6

[20] P. F. Jacobs, "Rapid prototyping & manufacturing: fundamentals of stereolithography". Society of Manufacturing Engineers, 1992.

[21] J. Wachtler and M. Ebner, "Support of video-based lectures with interactions-implementation of a first prototype," in World Conference on Educational Multimedia, Hypermedia and Telecommunications, vol. 2014, no. 1, 2014, pp. 582–591.

[22] J. Wachtler, "LIVE documentation," online, 2018, https://josefwachtler.wordpress.com.

10 Authors

Josef Wachtler is currently doing his PhD at Graz University of Technology in the area of interactive learning videos. This means that his research is focused on the development and the evaluation of interactive video tools for different purposes. He is working at the Department Educational Technology at Graz University of Technology and is responsible for developing different kinds of learning applications and tools. For more information like further publications, please visit his blog: https://josefwachtler.wordpress.com/

Martin Ebner is currently head of the Department Educational Technology at Graz University of Technology and therefore responsible for all university wide e-learning activities. He holds an Adjunct Prof. on media informatics (research area: educational technology) and works also at the Institute for Interactive Systems and Data Science as senior researcher. His research focuses strongly on seamless learning, learning analytics, open educational resources, maker education and computer science for children. Martin has given a number of lectures in this area as well as workshops and keynotes at international conferences. For publications as well as further research activities, please visit his website: http://martinebner.at

Article submitted 2019-06-06. Resubmitted 2019-07-15. Final acceptance 2019-07-20. Final version published as submitted by the authors.