Tommy Kubica, Tenshi Hara, Iris Braun, Alexander Schill

An Approach to Support Interactive Activities in Live Stream Lectures

Erstveröffentlichung in / First published in:

15th European Conference on Technology Enhanced Learning (EC-TEL 2020). Heidelberg, 14.-18. September 2020. Springer, S. 432-436. ISBN 978-3-030-57717-9

DOI: https://doi.org/10.1007/978-3-030-57717-9_40

Final edited form was published in "15th European Conference on Technology Enhanced Learning (EC-TEL 2020). Heidelberg, 2020".
S. 432-436. ISBN 78-3-030-57717-9
https://doi.org/10.1007/978-3-030-57717-9_40

An Approach to Support Interactive Activities in Live Stream Lectures

Tommy Kubica[1], Tenshi Hara[2], Iris Braun[1], and Alexander Schill[1]

[1] Faculty of Computer Science, Technische Universität Dresden, Dresden, Germany
{prename.surname}@tu-dresden.de
[2] Saxon University of Cooperative Education, State Study Academy Dresden, Dresden, Germany
tenshi.hara@ba-dresden.de

Abstract. Higher education is still mainly based on traditional face-to-face teaching formats such as lectures. Although known for a long time, the demand for alternative lecture designs has only recently grown significantly due to the present pandemic crisis, which requires a physical separation between the students and also the lecturer. This separation results in challenges of its own: While most solutions already allow the lecture to be held via streaming, it is even more challenging to involve students, or to create interactions with them or between them. Some approaches allow to activate students through simple polls, chats, or collaboration in breakout sessions. However, either the functional scope of these activities is severely limited or they are restricted to small scenarios. The contents of this paper have been under investigation for some time, but in lieu of the current CoViD-19 pandemic, the authors have decided to apply their preliminary findings in actual lectures and present the results here. The main goal is to investigate the addition of more advanced activities that can be adapted to the current scenario. Therefore, an approach is presented that enables lecturers to create custom workflows of interactive activities such as several types of polls, question and answer, instant feedback, or group formations with corresponding interactions. The application in real scenarios is currently being investigated.

Keywords: Live Stream Lecture · Synchronous Learning · Audience Response System · Group Formation · Collaborative Learning

1 Introduction

In recent months, the demand for remote alternatives to traditional teaching formats has grown significantly due to the development of the Corona crisis [9]. Therefore, many universities recommend using asynchronous formats, such as the provision of lecture videos or teaching materials, and their support by synchronous formats, such as webinars or live stream lectures – following the mantra "asynchronous over synchronous". However, the majority of lecturers transfer their lectures nearly entirely synchronously into the virtual space, either for the sake of simplicity or because it is not otherwise feasible for a certain topic.

Final edited form was published in "15th European Conference on Technology Enhanced Learning (EC-TEL 2020). Heidelberg, 2020". S. 432-436. ISBN 78-3-030-57717-9
https://doi.org/10.1007/978-3-030-57717-9_40

Especially for live stream lectures, this leads to didactic challenges, as the communication takes place unidirectional from the lecturer to the students. While it is already very difficult to motivate and involve the students in a traditional 90-minute lecture (cf. [8]), it is an even more challenging or impossible task in this setting [1]. However, in order to tackle these issues, most systems provide chat or simple poll opportunities to interact with the students and receive feedback on whether they understood the topic or not. Anyway, more advanced bidirectional communication, as it takes place in webinars, is limited to small scenarios and often replaced by asynchronous alternatives such as forums.

Thus, our goal is to investigate functionality that is targeted to support live stream lectures more efficiently. As a solution, we propose the adoption of an approach called scenario-tailored Audience Response System (stARS) that allows lecturers to support their personal teaching strategies [4]. Therefore, it provides a variety of activities, such as polls, question and answer, instant feedback, or group formations with corresponding interactions within these groups, that can be combined in customized workflows. In order to illustrate its applicability in live stream lectures, an example scenario is modeled, which involves the combination of chat and polling activities with group formations (based on students' previously given answers) and subsequent interactions within these groups.

2 Related Work

A common suggestion to properly implement remote teaching is the integration of interactive activities [9]. In this section, we will explore three groups of activities, which are supported by current live stream systems, such as BigBlueButton[3], GoToMeeting[4] or Zoom[5]. Additionally, we present approaches found in the literature that might be suitable to improve these systems.

The first group of activities is represented by polls that are integrated during the ongoing lecture and give students the opportunity to check their gained knowledge. Moreover, the lecturer receives feedback on whether the majority of students have understood a certain topic or not. Therefore, most systems support basic question types, e.g., *Single Choice*, *Multiple Choice* or *Short Answer* questions. However, the selection of a correct answer is often just as impossible as a comprehensive evaluation of the results. In contrast, current *Audience Response Systems*, which are commonly used to support traditional lectures, offer a variety of different question types, such as *Matching*, *Order* or *Region* (cf. [7]) as well as extensive means of evaluation (cf. [2]).

The second group of activities allows students to ask questions or give feedback on the ongoing lecture. Therefore, current systems provide *chat* or *question and answer* options that are moderated by either the lecturer or an assistant. In addition, some systems allow students to raise their hands in order to ask questions directly using their microphones. In comparison, there exist a variety

[3] `https://bigbluebutton.org/` – accessed 6/22/20
[4] `https://www.gotomeeting.com/` – accessed 6/22/20
[5] `https://zoom.us/` – accessed 6/22/20

Final edited form was published in "15th European Conference on Technology Enhanced Learning (EC-TEL 2020). Heidelberg, 2020".
S. 432-436. ISBN 78-3-030-57717-9
https://doi.org/10.1007/978-3-030-57717-9_40

of approaches providing advanced *question and answer* opportunities including the voting for both questions and answers, e.g., Pigeonhole[6] or Tweedback[7]. Tweedback additionally allows to give *instant feedback* as soon as something went wrong. Furthermore, [3] presents an approach that combines the *question and answer* activity with *instant feedback* by allowing different posts to be assigned on specific positions of the slides to highlight the cause of the problem.

The last group of activities focuses on collaboration with or between students. Therefore, current systems allow to create *breakout rooms*, which split the students into several groups. It is possible to manually or randomly form groups of students. Within these groups, different interactions can be performed, such as *audio*, *video* or *textual* chats. In addition, an interactive *multi-user whiteboard* can be used. Although the possibilities for collaboration are extensive, the implementation is often limited to small scenarios due to the high consumption of resources. Moreover, the formation of goal-oriented groups is not possible.

[4] presents an approach that allows to combine functions of all three groups of activities in a customized workflow. In addition to classic *audience response* functionality with different questions types, both *question and answer* and *instant feedback* are supported. Moreover, it provides novel, collaborative functions to support the formation of groups based on students' previously given answers. Furthermore, it allows for interactions within these groups using a chat or a voting to determine a group answer. In the following, we will present this approach in more detail and discuss its applicability as an addition to live stream systems.

3 Support Advanced Activities in Live Stream Lectures

3.1 The stARS Approach

The stARS approach builds on top of a uniform metamodel that is used as an abstraction to define the lecture and all its interactive activities [5]. In addition to common workflow elements, it specifies a variety of function blocks to support different kinds of communication, as described in the previous section. Furthermore, a set of parameters for each function block is defined to customize even those functionalities. Using a graphical editor allows lecturers to create their personal teaching scenarios in an intuitive manner. The execution of these scenarios is performed in a distributed, fully scalable infrastructure[8], which is able to run multiple scenarios simultaneously without interference [4].

3.2 Using stARS to Support Live Stream Lectures

In this section, we will visualize the applicability of stARS to support different groups of activities that take place in live stream lectures. Therefore, we present an example model that integrates all three groups in a single scenario,

[6] `https://pigeonhole.at/` – accessed 6/22/20
[7] `https://tweedback.de/?l=en` – accessed 6/22/20
[8] A prototype is provided on `https://stars-project.com/` (accessed 6/22/20).

Final edited form was published in "15th European Conference on Technology Enhanced Learning (EC-TEL 2020). Heidelberg, 2020".
S. 432-436. ISBN 78-3-030-57717-9
https://doi.org/10.1007/978-3-030-57717-9_40

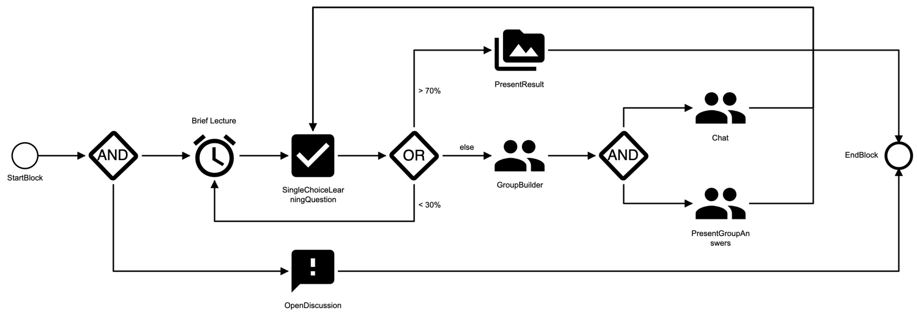

Fig. 1. An example scenario that integrates all three groups of activities.

as depicted in Figure 1. As usual in workflows, the scenario starts with a *start* block. Next, the control-flow is splitted into two sub-flows, resulting in the parallel execution of both the *question and answer* functionality (represented by a *OpenDiscussion* block) and an iteration of the teaching method *Peer Instruction* (represented by all other function blocks) [6]. The first phase of *Peer Instruction* is a *brief lecture*. During this phase, we do not want to enable any other functionality as specified in the parallel sub-flow. Therefore, we added a *PauseBlock* with a meaningful name. The second phase of *Peer Instruction* is the *concepTest*, in which the lecturer checks students' gained knowledge. In our example model, this is represented by a *SingleChoiceLearningQuestion*. For the method of *Peer Instruction*, it is essential to set the parameter *answerFeedback* to false – otherwise, students will receive feedback on whether their given answer was correct or not. Next, the conditional progression of *Peer Instruction* is modeled: If the majority of students performed very well (>70% correct answers), the topic is concluded (represented by a *PresentResult* block that displays a result chart on the students' devices) and the current sub-flow moves forward to the *end* block. However, if the majority of students performed very bad (<30% correct answers), the *brief lecture* has to be held in another version and the *concepTest* is repeated afterwards. If some students performed well, while others performed bad (*else*), a *peer discussion* should be executed. While this discussion is usually conducted offline, stARS overcomes this limitation by the introduction of novel, collaborative functions. First, a *GroupBuilder* is added, which represents the start of each group interaction. As we want to create groups that include students with different opinions, we set the parameter *buildSchema* to *differentAnswers* and refer the group builder to take into account the previous question (i.e., *SingleChoiceLearningQuestion*). By setting the parameter *groupSize* to 3, groups of three students are created. The *GroupBuilder* is followed by different interactions within these groups. In our example, the group members can discuss their answers with each other in a *Chat*. Simultaneously, their previously given answers are displayed, which is modeled by a parallel *PresentGroupAnswers* block. After completion of the *Peer Discussion*, the *concepTest* is repeated. The scenario ends as soon as both sub-flows proceed to the *end* block.

Final edited form was published in "15th European Conference on Technology Enhanced Learning (EC-TEL 2020). Heidelberg, 2020".
S. 432-436. ISBN 78-3-030-57717-9
https://doi.org/10.1007/978-3-030-57717-9_40

4 Results and Outlook

Up to now, two user studies have been carried out to validate the comprehensibility of both the metamodel (cf. [5]) and the editor (cf. [4]). It was shown that users with varying modeling abilities were able to express their personal teaching scenarios. However, two challenges became obvious: First, inexperienced users had difficulties to cope with the extensive functionality, and second, modeling of more complex scenarios, as they will occur in real scenarios, demands a high effort. Therefore, we are currently investigating concepts to support lecturers in the initial phase and the selection of suitable functions on the one hand and in the modeling of complex scenarios on the other. To assess the proper functionality of the prototype, simulations of the group interaction blocks were performed and their correct execution was verified. The prototype is currently being evaluated in smaller lectures. For the future, we plan to include it in larger and more complex scenarios.

Acknowledgments This work is funded by the German Research Foundation (DFG) within the Research Training Group "Role-based Software Infrastructures for continuous-context-sensitive Systems" (GRK 1907). Special thanks to Ilja Shmelkin, Robert Peine, Lidia Roszko and Sinthujan Thanabalasingam for their participation in the research project.

References

1. Arnaud, C.H.: Tips for teaching in the time of the coronavirus (2020)
2. Braun, I., Hara, T., Kapp, F., Braeschke, L., Schill, A.: Technology-enhanced self-regulated learning: Assessment support through an evaluation centre. In: COMPSAC 2018 – Proceedings of the 42nd Annual Computer Software and Applications Conference. IEEE (2018)
3. Bry, F., Pohl, A.Y.S.: Large class teaching with backstage. Journal of Applied Research in Higher Education (2017)
4. Kubica, T., Shmelkin, I., Peine, R., Roszko, L., Schill, A.: stARS: Proposing an Adaptable Collaborative Learning Environment to support Communication in the Classroom. In: CSEDU 2020 – Proceedings of the 12th International Conference on Computer Supported Education. INSTICC (2020)
5. Kubica, T., Shmelkin, I., Schill, A.: Towards a Development Methodology for adaptable collaborative Audience Response Systems. In: ITHET 2019 – Proceedings of the 18th International Conference on Information Technology Based Higher Education and Training. IEEE (2019)
6. Lasry, N., Mazur, E., Watkins, J.: Peer instruction: From harvard to the two-year college. American journal of Physics (2008)
7. Mader, S., Bry, F.: Audience response systems reimagined. In: ICWL 2019 – Proceedings of the 18th International Conference on Web-Based Learning. Springer (2019)
8. Stuart, J., Rutherford, R.D.: Medical student concentration during lectures. The Lancet (1978)
9. Sund, K.J.: Suggestions for moving teaching rapidly online in the face of the corona crisis (2020)