Guided Selection of IT-based Education Tools

Tommy Kubica[1], Tenshi Hara[1], Iris Braun[1], Felix Kapp[2], Alexander Schill[1]
[1]Chair of Computer Networks, School of Engineering Sciences
[2]Chair of Learning and Instruction, School of Science
Technische Universität Dresden, 01062 Dresden, Germany
Email: {forename.surname}@tu-dresden.de

***Abstract*—University courses in STEM fields benefit tremendously from the introduction of IT-based education tools (IET). Learning is not limited to knowledge of, but extended to knowledge acquisition through IET utilization. With the introduction of IETs, the preparation of lectures and courses becomes increasingly challenging for lecturers. Lecturers must know the IET's functional range and the didactical concept underlying each. They must also consider the interaction of IETs among each other and their behavior under different circumstances (time in semester, audience size, etc.). Additionally, didactics preferences of lecturers gravely influence IET selection. Only few solutions exist with an approach utilizing guided selection of IETs; none of which offers comprehensive explanations of didactics concepts and influencing factors. Attempting to address these problems, we added a guided IET selection to our AMCS platform which includes nine IETs not suitable for utilization at the same time. Our guided IET selection addresses didactical design as well as feasible und suitable combinations of the IETs. Any suggestion to lecturers at lecture or course creation time bases on a simple algorithm, which takes all of the before-mentioned aspects into consideration, while at the same time providing useful hints and explanations to the lecturers.**

I. Challenge

Universities cannot afford to continue to offer courses in STEM fields (science, technology, engineering, and mathematics) as traditional readings, tutorials or seminars. Instead, enhancements by addition of IT tools, such as audience response systems (ARS) [1], [2] or real-time evaluation systems, or evaluation in general [3], has become a common practice. The gist is to have students not only learn in STEM, but also learn through STEM. Of course, this can also be applied outside the scope of STEM fields, by modernizing all kinds of different courses, or selected individual lectures[1] thereof. Such modernized courses and lectures shall be considered tech-enhanced.

One of the first major problems with tech-enhanced lectures lies not within the IT tools themselves, but more in the common approach to these tools. Often lecturers are unsure about the functionality of a certain tool, or whether a certain tool is appropriate for their lecture, or not. This insecurity results in lecturers using specifically suitable tools not at all or, even worse, not properly. Sadly, lecturers are often enough not aware of the didactics underlying a specific tool, or are oblivious to well-established combinations of tools (the reverse, namely forbidden combinations of tools, are also often unknown). Additionally, lecturers new to IET systems are frequently over-strained with the functionality and choices they have to make. This creates additional workload which could be avoidable. The situation worsens when broadening the considerations from individual lectures to entire courses. The interplay of IETs must be observed spanning several lectures, while still serving individual level demands.

There are two approaches to support lecturers in implementing IETs. Similar to the classification used when it comes to the question on how to support students in self-regulated learning [4], lecturers can be either *directly* trained with regard to their didactic competencies and their media use skills or *indirectly* supported by designing the IETs in such a way, that an optimal amount of guidance is provided. Both approaches can be combined and beside resulting in an improvement of teaching quality have there advantages and disadvantages. Whereas the teaching skills and media competencies addressed in a specific training can be used in various situations, indirect support such as a guideline for the implementation of one specific tool might be attached to the very situation and class in which it is implemented. Nevertheless the second one is especially promising for professors and assistant professors at universities as they tend to have very little time and direct support in form of a training is very time consuming.

In this paper we present selected existing solutions for feature selections and elaborate on their shortcomings with respect to guiding lecturers through lecture and course creation. Afterwards, we will present our prototype with an implementation of a guided lecture creation assistant. Lecturers benefit from this due to suggested sets of tools based on input with respect to specific parameters of the lecture. The set of tools was designed in accordance to models of self-regulated learning [5], [6]. We present a few basic experiments before discussing priliminary results.

II. Related Work

The problem of avoidable workload and functional overstraining of lecturers is not uncommon. In fact, this is a well-known issue of software development. Users new to a system require guidance into the full functionality of the system, be it by reducing the initial functionality and aiding the users by means of tutorials within the system, or be it by selective presentation of certain functions, while other functions are disabled or masked out. With respect to IT-based education tools (IET), similar approaches can be found,

[1]A note on the terminology used by us: a course spans several lectures. Individual lectures can be either readings, tutorials, seminars, or practicals.

978-1-5090-5920-1/17/$31.00 ©2017 IEEE

namely *(reduced) fixed functionality*, *unexplained selectable functionality*, *explained selectable functionality*, and *guided selection* (or *proposition-based selectable functionality*). The latter is the mechanism to be found in our tool, Auditorium Mobile Classroom Service (AMCS)[2] [7]–[9].

Distinguishing educational systems into the before-mentioned types is reasonable as it is not always wise to make all functions available in all types of lectures [10]. Instead, some reduction must be processed in order to adapt systems to the actual necessity within a lecture.

A. Fixed Functionality

Many existing systems are limited to supporting selected scenarios only. Their functionality is fixed to a set of IETs specific for their respective scenario. For example, Backchannel[3] only supports instant feedback, whereas polleverywhere[4] also allows the answering of questions (targeted Q&A system).

Especially polleverywhere shows that it is wise to reduce the functionality. The mantra is "quality over quantity." Compared to other targeted Q&A systems, polleverywhere also supports:

- almost any kind of event with up to 50,000 participants,
- different question types such as brainstorming questionnaires, clickable image polls, et cetera,
- different types of participation instruments (website, SMS, Twitter), and
- real-time results that can be shared with the audience through the website, or by real-time embedding in Powerpoint, Keynote, or Google Slides.

Controlling the given fixed functions is possible by having the lecturer manually releasing, pausing, or terminating them.

Fixed functionality systems are designed to serve the demands of lectures as well as courses through their strict binding to specific scenarios. As long as a course has all of its lectures operate within the given scenario, no (additional) IET interplay problems should occur on the course level.

B. Unexplained Selectable Functionality

Systems with more functionality compared to the before-mentioned targeted Q&A systems often allow the selection of said functions by the lecturers.

Letsfeedback[5] allows enabling or disabling of *audience questions* or *welcome notes*, as exemplary shown in Figure 1. As is true for polleverywhere, the audience questions can be controlled by releasing, pausing, or terminating the function.

Due to the very limited set of functions and the self-explanatory function names, the selection of the functions can be considered to be intuitive.

The selection of functions is handled similarly in SMILE[6]. The functions selection menu allows deactivation or activation of so called modules. Each module can include different functional aspects. After activating a module, a new tab is created within which the module can be configured in more detail. The overall functional scope is very limited. Nevertheless, due to ambiguous naming conventions the selection process is unintuitive without extended familiarization. A better approach is presented in the next section.

Unexplained selectable functionality systems are designed to meet the demands of single lectures. The implications for courses are not considered. This makes these systems vulnerable to all drawbacks of IET interplay problems that can occur on course level. To the best of our knowledge, no such system even considers course design or warns users of the drawback. However, all investigated systems promote utilization throughout entire courses as if no IET interplay problems were to be expected.

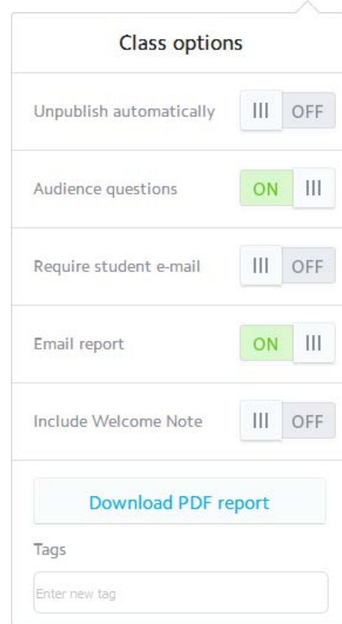


Figure 1. Functions Selection Menu in Letsfeedback

C. Explained Selectable Functionality

A system functionally comparable to SMILE is Tweedback[7]. However, Tweedback describes all functions within the functions selection menu as can be seen in Figure 2.

The descriptions are limited to functional aspects, while didactics and/or learning psychology are omitted/missing. With respect to the available functionality, this way of functions presentation is very intuitive and works well for Tweedback.

A very complex and functionally extensive system is ARSnova[8]. The system allows the selection of different use-cases as shown in the background of Figure 3. Each use-case has a more or less intuitive name and holds additional information after a click on a help symbol as shown in the modal message in the foreground of Figure 3. The use-cases are therein described and partially presented with respect to an underlying didactics concept.

[2]https://amcs.website/
[3]https://backchannel.cnc.io/ – accessed 4 July 2017
[4]https://polleverywhere.com/ – accessed 4 July 2017
[5]https://letsfeedback.com/en/home/ – accessed 4 July 2017
[6]https://www.smile.informatik.uni-freiburg.de – accessed 4 July 2017
[7]https://www.tweedback.de – accessed 4 July 2017
[8]https://arsnova.eu/mobile/#en – accessed 4 July 2017

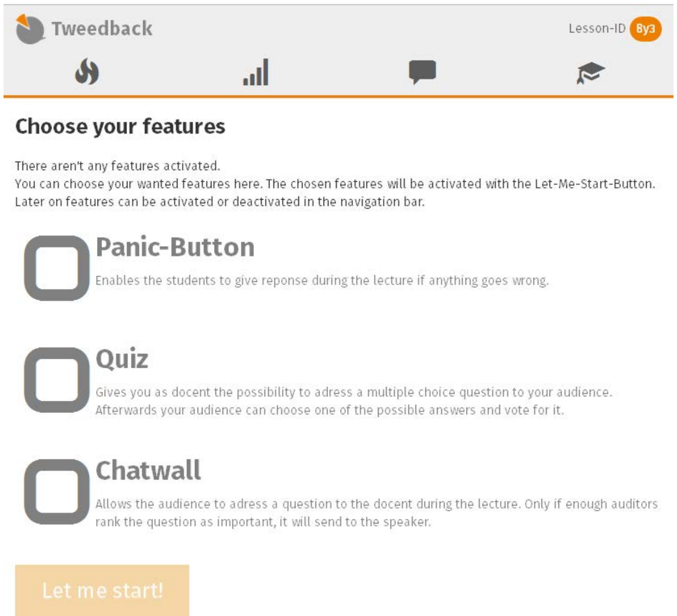


Figure 2. Functions Selection Menu in Tweedback

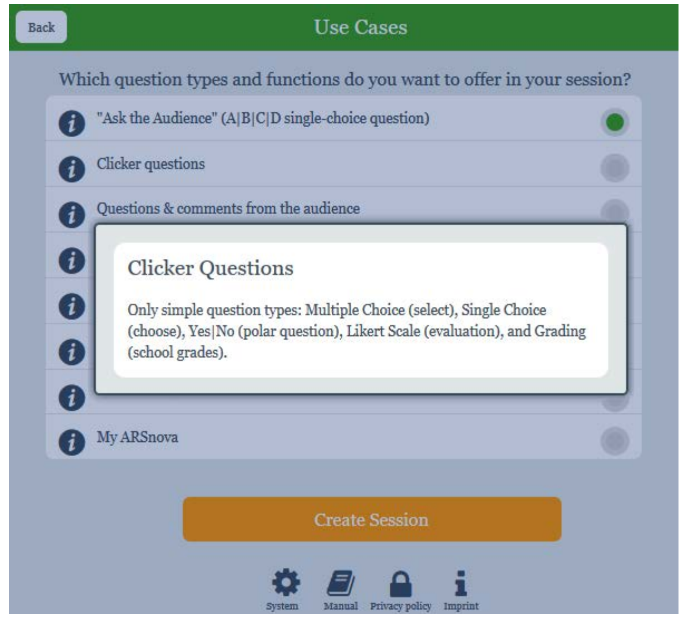


Figure 3. Use-case Selection and Description in ARSnova

Besides the predefined use-cases, ARSnova also supports the creation of "individual function sets." Such sets can be created by selecting from a list of main functions and optional additions. However, when creating individual sets, no functional or didactics description is provided. This is very unfortunate especially as new abstract concepts are likely to be introduced to the lecturer. Their functional range and scope cannot be intuitively derived from the given function names only. A deduction of course-spanning IET interplay effects is impossible that way.

Tweedback only extends Letsfeedback with the addition of the above-mentioned descriptions, whereas ARSnova introduces a variety of combined functions. For example, "interactive presentation" as a means to stage a lecture under Peer Instruction [11]. However, the downside of this approach lies in the exclusive selection of sub-functions. Some sub-functions mutually exclude each other in the pre-set use-cases. While audience questions in the fashion of "Who wants to be a Millionaire" can be useful in one situation, a lecturer may want to use evaluation questions in another. Nevertheless, there is no use-case in which both types of questions can be used in the same lecture. When creating an individual set as mentioned above, lecturers need to derive on their own that these types of questions are "lecture hall questions." Unfortunately, there is no description hinting to the fact that lecture hall questions can realize audience questions and evaluation questions.

D. Guided Selection

As seen in the previous section, many systems attempt to support functions selection for individual lectures by providing descriptions of the individual functions or the use-cases they are used in. However, none of the systems considers the limitation on functions or attempts a sensible pre-selection as proposed in [10]. In general, lecturers are abandoned in assessing the sensibility of functions with respect to their individual lecture. Considerations addressing course-level IET interplay do not exist

A system guiding the lecturer through the selection by providing a sensibility assessment of the functions with respect to the individual lecture as well as courses is – to the best of our knowledge – currently only available in our prototype. Especially in comparison to ARSnova, our prototype supports lecturers in the selection process as sub-functions are always assigned to corresponding feasible use-cases while observing the point in time during a semester (i.e., when during a course a lecture takes place).

III. Prototype

In order to provide a platform for tech-enhanced education research we have been implementing topical aspects of learning psychology, didactics and information technology into a prototype platform over the past several years, namely our before-mentioned Auditorium Mobile Classroom Service (AMCS). The main features of AMCS have been evaluated and constantly developed [7]–[9]. AMCS aims at enhancing the quality of lectures by providing both, support to the students and the lecturer. The goal is to offer lectures that foster an active, constructive and highly individual process as [12] proclaimed. Additionally, we plan to include our research results with respect to tutorials (e.g., [13], [14]) in combination with the results presented in this paper into the production system during the winter semester of 2017. For now, they are limited to the staging prototype system. Overall, AMCS especially focuses on didactics concepts such as Peer Instruction [11] and the possibilities audience response systems (ARS) offer [1], [2]. However, AMCS is based on a psychological framework describing the learning process of students during the lecture. The system's different features are derived from models of self-regulated learning (e.g., [5], [6], [15]). With AMCS the lecturer is able to construct learning questions, surveys and messages for distinctive students in advance of the lecture. These interventions are delivered during the session according to defined rules. AMCS thereby turns the lecturer into a designer of a tech-enhanced learning environment by enabling them to select from a variety of IT-based education tools (IET). In the latest version we focused on an IET that addresses the needs of the lecturer. In order to help students in the auditorium to successfully learn, lecturers have to know more about the audience: their interests, their personal goals, the state of knowledge, their difficulties and their motivation must be considered when designing support. A continuous evaluation in terms of formative evaluation during the course and a summative evaluation at the end of the course is necessary in order to improve the quality of the classes [9]. However, selecting and setting up the components of the system, namely the individual IETs, is demanding towards the lecturers. Often, they become discouraged when confronted with the entirety of IETs offered by AMCS. The same problem also applies to other complex systems such as ILIAS[9] or ARSnova.

AMCS provides its IETs' data processing and storing on a central web server running Ruby on Rails. Representation of information and user interactions are delegated to mobile devices, namely by utilizing web browsers on laptop computers and smartphone apps (an app for Android has been available, and an app for iOS was just recently released). Therefore, utilization of AMCS is primarily made available to lecturers and students through their own devices following a second screen approach [14].

Currently, AMCS contains seven in-use IETs[10] which we present in III-A through III-G. Additionally, a new IET for graphical feedback and discussions [16] is currently in testing in a separate staging environment. All currently implemented IETs aim at supporting students in mastering the demands of the learning process. According to [6], students have to face various demands during the forethought phase, the performance phase and the self-reflection phase. Students differ with regard to the goal orientation, attribution style, prior knowledge, reception of new knowledge, and processing of new knowledge. Evaluating the learning activity during the self-reflection phase as well as the changing relevant strategies for the next learning activity is also influenced by personal experiences. Whether or not students identify a learning strategy as useless depends on their meta-cognitive skills, et cetera.

In order to lay out the foundation to the guided selection, we wish to briefly introduce the main IETs used. The reader may take note that students remain entirely anonymous in AMCS. A de-anonymization concept was recently submitted in a master thesis, but shall be excluded from this paper.

[9]https://ilias.uni-giessen.de/ilias/ – accessed 4 July 2017

[10]There are several disabled IETs considered instable or "too early beta".

A. Interests / Personal Goals

Students are asked about their personal goals and interests – or in more general terms, why they are attending – at the beginning of the lecture. E.g., they are asked whether they are "interested in the topic or just need the credit points for the course." Similar questions can be posted on lecture-level. The answers are stored for each student in a database and are used as triggers for later-on interventions, such as messages and learning questions. At the same time, the short survey at the beginning helps students to reflect about their goals.

B. Learning Questions distributed over the Lecture

AMCS is able to deliver learning questions at different points of time during the lecture. However, in contrast to other ARS, AMCS provides individual feedback. Students can answer second-attempt single-best-choice questions on their mobile device. This means they have two attempts to select a single correct answer out of several offered options, whereas the incorrect options serve as plausible answer distractors. The students receive feedback after choosing an option. After the second incorrect attempt AMCS displays the correct answer, ideally extended by an explanation why a plausible distractor is actually wrong. The lecturers are still able to display the aggregated audience results via LCD projector in case they want to discuss in public. Along with AMCS comes a tutorial helping lecturers to design learning questions and feedback according to certain construction rules. This makes them a powerful IET supporting the learning process, both in the necessary cognitive and meta-cognitive processes.

Due to high demand, multiple choice questions are being tested in the moment. However, it has not been determined how to scale[11] answers, yet.

C. Meta-cognitive Prompts

Depending on the students' preference (e.g., exam preparation or interest in the subject), strategic guidance is delivered during the lecture. If students stated that their main goal in the present class is to pass the exam, they might receive the following message on their mobile device: "The issue on the current slide is relevant for the exam. The examiner may ask about X" The intention of such meta-cognitive prompts is to help students to regulate their attention and motivation in order to reach their personal learning goals.

D. Cognitive Prompts

The learning questions at the beginning, in the middle and at the end of the lecture contain the possibility to identify knowledge gaps of the students. For example, students who have made mistakes in a learning question at the beginning of the lecture, receive message containing a cognitive prompt at a later point of time such as: "You have made a mistake

[11]How to count incorrect versus correct answers? For example, if there are two correct choices out of five, how to count a submission with one correct and one incorrect choice selected? Is it 50% correct (as in one out of two correct ones), or 17% (as in one out of two correct ones *minus* one out of three incorrect ones), or something totally different?

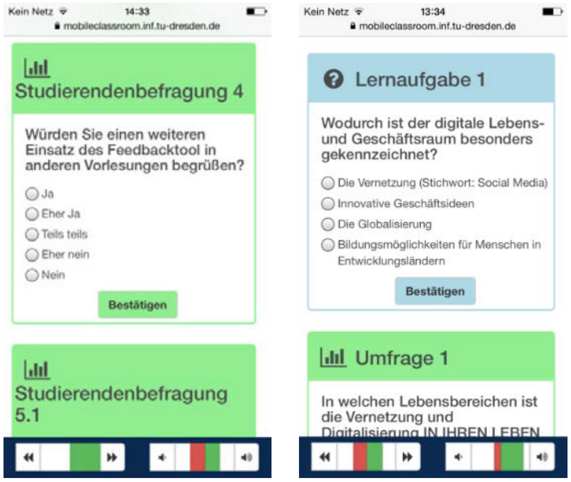


Figure 4. Exemplary Interface on Smartphone

in the learning question on topic X. You thought that concept Y was the answer to the question instead of concept Z. – Pay attention NOW; concept Z is explained on the current slide."

Variations in tone and urgency are possible, but currently need to be designed by the lecturer during set-up of their lectures. We may add a system function to automatically derive the urgency of a message based on a student's learning progress at a later time.

E. Providing further Material

AMCS allows sending additional information to the students, such as links, documents, or continuative presentation slides. This happens according to the personal learning goals of the students. For example, a student may receive a message such as "You have indicated that you are interested in writing a thesis on this topic. The chair is doing research on the topic presented on the current slide. You can find possible research queries for a thesis on the subject under the following link: edu.example.com/theses"

F. Scripted Discussion – Animation of Questions

In preparation of a lecture, the lecturer can design messages by providing content as well as a target group and point in time during class. The target group can be either the entire group of students or any arbitrary subset derived through assessment of the answers provided to personal goals and learning questions (ref. III-A and III-B). Designing messages contains the possibility to enhance slow discussions. The time reserved for questions and answers can be used in an optimized way by sending messages like "Stand up right now and ask the following question loudly into the room: 'What's the practical use of this theory?'" Students can be animated to pose questions which allow them to reach the next knowledge level.

G. Immediate substantial Lecture Evaluation

Finally, AMCS also offers formal evaluation of lectures. Compared to traditional ways of evaluation it allows to get

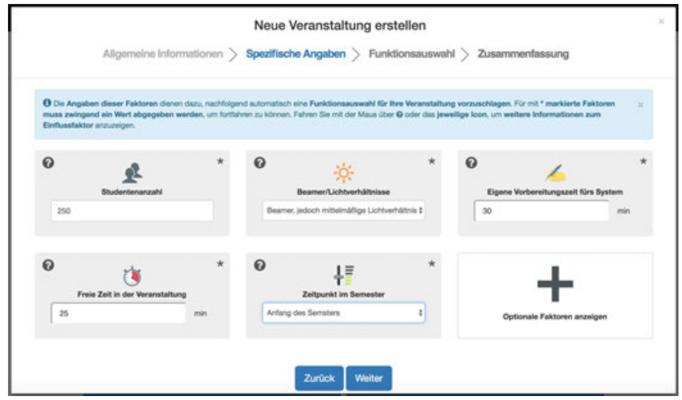


Figure 5. Query of Lecture Setting and Circumstances

more information by the means described earlier this section. Providing learning questions, surveys with different formats and messages allow gathering data relevant for evaluation.

Additionally, AMCS also contains an extra function for immediate feedback to the lecturer, namely a traditional ARS with dimensions on volume and speed of the lecture. The interface for the immediate feedback is displayed in Figure 4, which also shows evaluative questions (ref. III-A) on the left and learning questions (ref. III-B) on the right.

IV. Guided Selection Interface

The newly introduced guided selection graphical user interface (GS-GUI) focusses on easily answerable questions. It aims at determining the lecture setting and circumstances (Figure 5) with few as possible questions while guiding the lecturers with explanatory feedback whenever required. The feedback contains information on functionality as well as didactics conception of the IETs. Additionally, the GS-GUI provides feedback as soon as one or more IETs are designated for utilization. Such feedback especially includes information on why certain IETs boost or hinder each other. In the end, lecturers are presented with an assessment of IET suitability for their individual lecture setting and circumstances (Figure 6). Currently, our prototype is partially aware of the course setting (i.e., sets of lectures and times during the semester), but our GS-GUI does not provide course setting selections, yet.

In order to reach the final assessment, AMCS relies on a simple but expandable algorithm to calculate the result. Each of the queried factors (currently five, but more can be added later), namely the audience size, availability of lecture hall equipment (e.g., LCD projector), the lecturer's available set-up time, the available on-site time, as well as the point in time during the semester, is matched into one of three factor categories each (finer granular categories can be utilized later). For example, the factor "audience size" currently distinguishes between "1 to 25 students" (typically a tutorial), "25 to 50 students" (typically a seminar or small reading), and "more than 50 students" (in general a reading). As another example, the available on-site time for IET utilization on site distinguishes

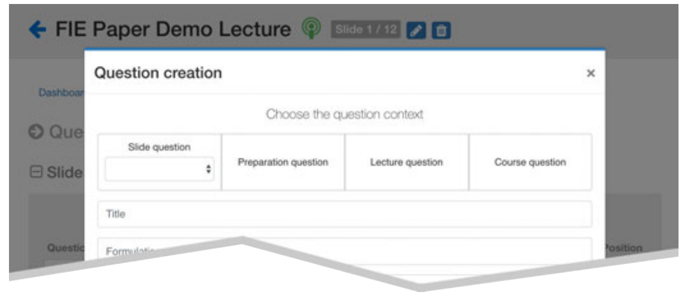
Figure 6. Summary of the IET Selection for the Lecture Setting

between “0 to 10 minutes,” “11 to 30 minutes,” and “more than 30 minutes.” The boundaries can be configured to match the demands of different education institutions.

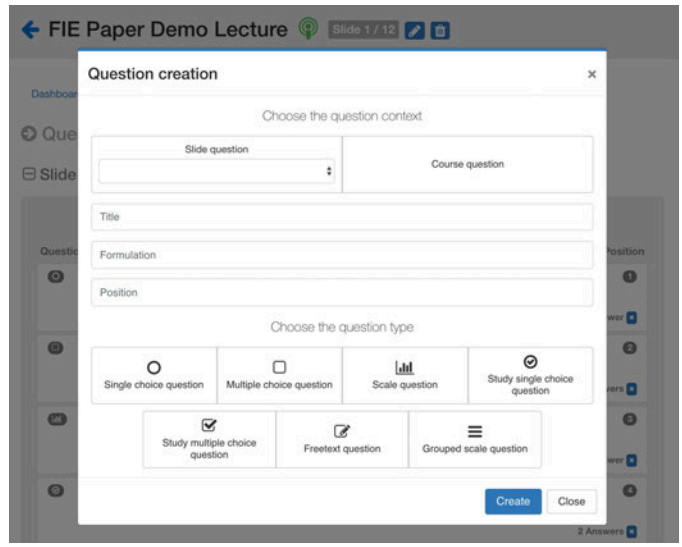
In addition to the before-mentioned factors, which are all objectively observable, three optional factors that are of a more subjective nature can be input (once again, more can be added later). These cover the students’ attitude toward participation, the lecturer’s attitude toward student involvement, and the fraction of expected participating students. Again, the algorithm distinguishes three categories for each optional factor.

After the entered factors and the input’s categorization is computed, our selection algorithm modifies the calculated result depending on the lecturer’s and students’ experience with the system (how often the system has been used in a lecture series), and –most importantly– how selected IETs influence each other. This influence is once again categorized threefold, namely ones precluding one another, IETs often used together (low dependency), and IETs that rely on or even require other IETs (strong dependency). Only after this final modification of the algorithm’s computation a result output is generated and presented to the lecturer. An exemplary output is depicted in Figure 6.

Finally, the lecturer has the opportunity to change the selection based on their personal assessment of the computed result. This allows lecturers to set-up a lecture based on their preferences. For example, a lecturer who dislikes discussion-based IETs can deselect them. Any alteration of the suggested selection is commented by the system in order to explain why such alteration can or shouldn’t be conducted. In general, these automatically generated comments focus strongly on didactics



(a) slightly reduced context



(b) strongly reduced context; all types available

Figure 7. Different reduction levels for available question contexts

reasoning, but also include reasoning based on functional aspects of the IETs (this is required for the “strong dependency” category mentioned earlier).

Based on the resulting set of IETs, our prototype may reduce the options available to the lecturer when preparing actual lectures. For example, when creating questions that the students will see during a lecture, certain types of questions that are inept for the specific setting or certain contexts may be disabled or unavailable. The reduction of available question contexts is shown in Figure 7.

V. Experiments

We conducted a few basic experiments in order to ascertain a first ballpark figure of whether we a progressing in the correct direction with our ideas. As the presented prototype and our research are still a work in progress, we are currently running an extensive semester-spanning evaluation with several lecturers in multiple lectures. Therefore, the expected results on suitability, impact on lecture design, et cetera must be postponed to a follow-up paper.

A. Experimental Setup

We investigated the described guided IET selection with respect to functionality, usability, and quality of the resulting selection. For that, we relied on System Usability Scale (SUS) [17], Thinking Aloud, and the actual selection output of the system. All tests involving users were conducted on the same machine, a MacBook Pro with a 3.1 GHz Core i7 and 16 GB of RAM. We ran the test under OS X 10.11.3 in a 64-bit instance of Chrome 52.0.2743.116 displayed on a 13.3ďisplay at a resolution of 1440×900 px. We conducted tests with 11 users (thereof 5 "experts" who were already familiar with a subset of the system functionality) in two iterations. Each test was conducted around noon and lasted for about one hour. The system's output, namely the result of the selection algorithm, was noted after each test. Afterwards the system was reset to the initial test settings to ensure all test subjects started with the same conditions.

We had our test subjects play through three distinct scenarios. First, an introductory lecture of a well-attended mathematics course (something like *Math 101*); second, a smaller tutorial for the mathematics course during the semester; and last, the final seminar of the mathematics course before the exam at the end of the semester.

The test subjects were asked to play through each scenario thinking aloud, as well as answer specific questions related to the quality of the system's automated decisions. After the simulation they were invited to freely alter the system's output and comment on the result. Finally, each test subject was asked to complete a SUS questionnaire.

B. Results

Based on the feedback we received from the test subjects, a set of best practices could be derived. Most importantly, lecturers always need to know how much time they are supposed to invest into the system. This is true for the preparation time, as well as for the actual utilization of the system during or after a lecture or tutorial. Therein, it has to be clearly distinguished how much time is actually available on-site, and how much of that can be made available for the system's IETs. Regardless of how self-explanatory an IET seems to be, it is imperative to always add hints and descriptions. These should include assessments as well as actual impact on the selection algorithm. If an IET collides with another IET, a clear discouraging statement should be provided. Analogously, IETs supporting each other should be highlighted. Based on the parameters of the lecture or tutorial (such as time in the semester, room capacity, et cetera), each IET must be clearly marked whether it positively or negatively contributes to the selection algorithm. This is equivalent to a statement whether an IET is didactically useful in certain setting, or not, without actually explaining the didactics to the lecturer (who might not be firm in didactics theory). Similar to the described dependencies between IETs, functional dependencies (potentially spanning multiple IETs) must be made clearly visible to the lecturer during selection. This further emphasizes each other supporting/hindering IETs.

We adapted the system's guided selection process accordingly between the two test iterations. Consequentially, the system's SUS was expected to rise; however, it decreased from 88.3% after the first iteration to 80.6% after the second iteration. This needs to be further investigated, but for now we assume the decrease originates in the first iteration consisting of expert users only (ones who were familiar with AMCS before testing), while the second iteration had a significant portion of non-expert users (ones who were newly introduced to AMCS).

From an algorithmic perspective, the system's output (the selection result) was as predicted and expected. For the three scenarios (first lecture, tutorial during semester, final seminar), the IET selection was overall correct.

In the first scenario, all question-based IETs with the exception of preparatory question were suggested by the system. This is correct as we should assume students have no prior knowledge. Hence, repetition of previously taught material is not possible (there are no previous lectures), and a current state of knowledge or learning progress cannot be determined. Due to the expected audience size the system also strongly discouraged the utilization of tutor questions (one where students directly ask the lecturer) as well as discussions. The potential of spam as well as distraction of the lecturer and the students would be to high. In the second scenario, all IETs with respect to students' interests as well as learning progress were discouraged by the system. This is sensible as the idea of tutorials is to apply the knowledge presented in lectures and deepen the understanding thereof. Additionally, instant feedback functions (namely ARS IETs) were discouraged as they cannot be applied to small groups in a sensible manner.

Finally, for the third scenario the system strongly encourages utilization of feedback and question IETs. This makes sense as students in preparation of the exam have potentially many questions with respect to individual topics of a course that they want to ask the lecturers and tutors directly, as well as discuss with their fellow students. Additionally, the system discourages utilization of evaluative feedback as it is too late for changes to the lectures in the course. Such feedback may help improve future installments of the same course by fixing issues in lectures or course material, but it does not benefit the current students.

The before-mentioned algorithm output results as well as further details are summarized in Table I.

VI. Assessment

Overall, a guided selection is a helpful tool for lecturers in preparation of teaching events such as lectures and tutorials. When a few simple design guidelines are followed, the results of the selection algorithm can be easily understood by the lecturer. However, some results can and should be questioned. In our assessment, the guided selection can help improve time-consuming preparation tasks, and the algorithmic results can be trusted in great parts. Nevertheless, closer observation of the results under didactics perspective shows that they should be double-checked in some instances. Therefore, we conclude that the guided selection is an improvement as it helps to make an initial selection with good explanations. Lecturers can then fine-tune the selection based on their didactics expertise. This

Table I
SELECTION RESULTS PER SCENARIO

Function	Lecture at beginning of semester	Tutorial in middle semester	Seminar at end of semester
Course Questions	yes	no	yes/no*
Event Questions	no	yes	no
Slide Questions	yes	yes	yes
Meta-cognitive Prompts	yes	no	(yes)
Cognitive Prompts	yes	no	(yes)
Provision of further Links/Material	yes	no	(yes)
Instant Feedback	yes	no	no
Tutor Questions	no	yes	no
Voting Questions	yes	no	(yes)
Discussion Questions	no	no	(yes)

* suggested in some instances; suggestion is however questionable for the scenario (it is too late in the semester for interest-based IETs)

() suggestion is only valid under special boundary constraints

eliminates the initial time-consuming selection, especially if the system supports storing of preferred IET combinations for quick and template-based creation of lectures, tutorials, or seminars.

The overall very positive SUS (above 80%) shows that our approach indeed helps lecturers intuitively select IETs suitable for their demands.

VII. CONCLUSION AND FUTURE WORK

In this paper we discussed challenges with respect to the selection of didactically sensible and useful IETs. We presented an approach for guided selection of IETs especially aimed at reducing the time required for the selection and freeing the lecturers of considerations with respect to didactics sensibility of selections. Our algorithm's output was discussed with respect to functionality, usability and sensibility.

In the future an interactive tutorial is required. Even though the system is designed to guide lecturers through the selection of IETs, the tools themselves are still too complex. We think, an exemplary walk-through or visual aid could help lecturers in understanding the IETs themselves. Even if lecturers absolutely understand how and why certain IETs were selected, they might not understand how the selected IETs function and perform within in real lectures or tutorials. Currently, this decoupling of guided selection and actual IET utilization is a major drawback of ACMS. Utilizing IETs should be same as easy as selecting them.

As we have not yet had the opportunity to evaluate the system intensively, future work must be based on intensive evaluation. For this, we have secured funding for tests in multiple lectures at our university. Overall, we will test the system in at least four courses in the summer semester of 2017, as well as at least three courses in the following winter semester. The tests will be accompanied by interviews with lecturers and students, and intensive analysis of server logs. At the end, we hope to be able to fine-tune our list of best-practices and hence further increase the quality of our algorithm's suggestions.

We should investigate how successful initial learning of the system's functions can be achieved by automated tutorials in comparison to training sessions. Also, the influence of the availability of a manual compared to integrated functional descriptions should be analyzed. For this, an investigation into the time component could be rewarding. We may want to log the time passed between a lecturer's first log-in and the first successful and complete creation of a lecture. This may provide hints with respect to the benefit of trainings and manuals, as well as the automated introductions. It would allow an in-use perspective rather than the delayed perspective ascertained through interviews conducted with lecturers in retrospective. However, retrospective interviews should not be omitted. Instead, we could use them in order to determine whether lecturers with progressing time start using IETs not initially suggested to them on their own, or not.

The addition of weights to the selection algorithm could improve result quality. In the moment, all IETs factor in equally. This does not allow lecturers to customize the algorithm to individual preferences. For example, a lecturer preferring Peer Instruction lectures may want the algorithm to benefit question-based IETs for the pre-reading and ConcepTest phases of Peer Instruction. With weights, such preferences could be accounted for.

Finally, we want to add lecturer feedback to the system. When lecturers are able to provide feedback on IETs, especially the sensibility of their selection as well as their usefulness, the automated selection algorithm could *learn* from this feedback and improve the selection results accordingly.

ACKNOWLEDGMENT

We wish to acknowledge the contributions of the AMCS team, namely our graduate students Lucas Braeschke, Patrick Buchholz, Markus Heider, and Samuel Knobloch.

Disclosure: This paper summarizes the core research contributions of Tommy Kubica's master's thesis. It is amended with new results and corrections since the thesis' filing date.

REFERENCES

[1] R. E. Mayer, A. Stull, K. DeLeeuw, K. Almeroth, B. Bimber, D. Chun, M. Bulger, J. Campbell, A. Knight, and H. Zhang, “Clickers in college classrooms: Fostering learning with questioning methods in large lecture classes,” *Contemporary educational psychology*, vol. 34, no. 1, pp. 51–57, 2009.

[2] K. Weber, B. Becker *et al.*, *Formative Evaluation des mobilen Classroom-Response-Systems SMILE*, ser. E-Learnng zwischen Vision und Alltag. Waxmann, 2013.

[3] D. Kirkpatrick, *The four levels of evaluation*. American Society for Training and Development, 2007, vol. 701.

[4] H. F. Friedrich and H. Mandl, “Analyse und Förderung selbstgesteuerten Lernens,” in *Enzyklopädie für Psychologie, Themenbereich D, Pädagogische Psychologie, Bd. 4 Psychologie der Erwachsenenbildung*, H. Mandl and H. F. Friedrich, Eds. Göttingen: Hogrefe, 1997, pp. 237–276.

[5] P. H. Winne and A. F. Hadwin, “Studying as self-regulated learning,” *Metacognition in educational theory and practice*, vol. 93, pp. 27–30, 1998.

[6] B. J. Zimmerman, M. Boekarts, P. R. Pintrich, and M. Zeidner, “Attaining Self-Regulation: a social cognitive perspective,” *Handbook of self-regulation*, vol. 13, pp. 13–39, 2000.

[7] F. Kapp, I. Braun, H. Körndle, and A. Schill, “Metacognitive Support in University Lectures Provided via Mobile Devices - How to Help Students to Regulate Their Learning Process during a 90-minute Class,” in *Proceedings of the 6th International Conference on Computer Supported Education (CSEdu 2014)*. ScitePress, 2014, pp. 194–199.

[8] T. Hara, F. Kapp, I. Braun, and A. Schill, “Comparing Tool-supported Lecture Readings and Exercise Tutorials in Classic University Settings,” in *Proceedings of the 7th International Conference on Computer Supported Education (CSEdu 2015)*. ScitePress, 2015, pp. 244–252.

[9] F. Kapp, I. Braun, and T. Hara, “Evaluating Lectures Through the Use of Mobile Devices - Auditorium Mobile Classroom Service (AMCS) as a Means to Bring Evaluation to the Next Level,” in *Proceedings of the 8th International Conference on Computer Supported Education (CSEdu 2016)*. ScitePress, 2016, pp. 251–257.

[10] B. Leiding, J. Vetterick, and C. H. Cap, “Exploring Classroom Response Systems in Practical Scenarios,” 2014, https://www.researchgate.net/profile/Benjamin_Leiding/publication/280008669_Exploring_Classroom_Response_Systems_in_Practical_Scenarios/links/55a3773a08aeaefdb97bba3b.pdf.

[11] E. Mazur, *Peer Instruction: A User’s Manual*, Series in Educational Innovation ed. Upper Saddle River, NJ: Prentice Hall, 1997.

[12] N. M. Seel, *Psychologie des Lernens: Lehrbuch für Pädagogen und Psychologen*. UTB, 2003, vol. 8198.

[13] T. Hara, “Analyses on tech-enhanced and anonymous Peer Discussion as well as anonymous Control Facilities for tech-enhanced Learning,” Ph.D. dissertation, Technische Universität Dresden, 2016, http://nbn-resolving.de/urn:nbn:de:bsz:14-qucosa-205517.

[14] T. Hara, I. Braun, F. Kapp, and A. Schill, “Toward a Second Screen Peer Discussion Derivative,” in *Integrated STEM Education Conference (ISEC), 2016 IEEE*. IEEE, 2016, pp. 235–240.

[15] A. F. Hadwin, P. H. Winne, D. B. Stockley, J. C. Nesbit, and C. Woszczyna, “Context moderates students’ self-reports about how they study.” *Journal of educational psychology*, vol. 93, no. 3, p. 477, 2001.

[16] T. Hara, K. Chen, I. Braun, and F. Kapp.

[17] J. Nielsen and R. Molich, “Heuristic evaluation of user interfaces,” in *Proceedings of the SIGCHI conference on Human factors in computing systems*. ACM, 1990, pp. 249–256.