Know the Knowledge of your Students: A Flexible Analytics Tool for Student Exercises

Sören Aguirre Reid, Frank Kammer, Daria Schüller, Markus Siepermann, and Jonas Wölfer

Technische Hochschule Mittelhessen, Wiesenstraße 14, 35390 Gießen
`<firstname>.<surname>@mni.thm.de`

Abstract. Intelligent tutoring systems (ITS) have the power to influence lecturers' practices in the classroom and to improve students' learning. Many ITS provide standardized analytical methods to evaluate classroom performance or the results of the exercises. But they lack more sophisticated, flexible and individual analyses. This paper presents an e-learning Analytics Tool (EAT) for an ITS that provides a dashboard with key figures and a flexible analytics board (FAB). This enables lecturers to analyze students' performance in detail and identify misconceptions. For instance, the FAB allows us to classify student solutions, reveal conceptual errors in exercises, and analyze each part of an exercise. By this, common mistakes can be identified, and tailored feedback can be given to the students. Following the design science approach, the platform is designed in a general way so that it can be used for different types of exercises (e.g., Math, Excel, SQL). To assess the artifact, we conducted group interviews with German University lecturers from various courses. The results show that lecturers require a good overview of the submitted student solutions to provide timely feedback. They also appreciate the flexible analytics tool for detailed analyses of student solutions to understand student mistakes better. The employed architecture allows general analyses of exercises and the content of students' solutions. In addition, the EAT is not bound to a specific kind of exercise, but can cope with different kinds like SQL and Excel.

Keywords: Intelligent Tutoring System · ITS · Learning Analytics and Evaluation · Dashboard · Flexible Analysis

1 Introduction

E-Learning platforms mainly support five different kinds of tasks: (1) course administration, (2) provision of course material, (3) communication and cooperation, (4) providing exercises for practicing, as well as (5) assessment and grading [36]. Platforms for categories 1 and 2 like Blackboard, Canvas, Moodle, Ilias, or WEB-CT exist for many years. By and by, they have been extended with additional features of categories 3, 4 and 5. However, particularly category 4 still remains underdeveloped. Mostly, simple types of exercises are implemented like multiple-choice or fill-in-the-blank. More sophisticated exercises are left to specialized platforms, known as Intelligent Tutoring Systems (ITS). ITS provide challenging exercises that are mostly specialized to a certain field of application like mathematical conversion (e.g. [29]), graphical modeling (e.g. [35,37]), or SQL (e.g. [27]) etc. Nowadays, students' submissions are no longer just classified as correct or wrong. Students receive a detailed analysis of their submission stating which parts are incorrect and why. Such a detailed analysis helps students to better understand the learning subject so that they can improve their skills.

While students are the number one addressee of ITS, also the needs of lecturers are considered beyond the possibilities of easily providing exercises, which are automatically reviewed and assessed. This certainly relieves lecturers from a huge amount of recurring standard tasks. But apart from that, ITS can accommodate lecturers with information about students, courses, and exercises. No matter what kinds of exercises are provided, there is a lot of data concerning the usage of the system or the processing of the exercises by students. Provided that this data is analyzed in the right way, valuable insights concerning the learning behavior of students, their level of knowledge and individual performance in the course, the general status of courses but also concerning single exercises and subsequently the knowledge transfer of lectures can be generated. Already the detailed analysis of individual student submissions helps lecturers with providing feedback on students' performance. This in turn improves and accelerates students' learning processes and empowers them as self-regulated learners [31].

However, most ITS provide only general (meta-)information concerning courses [4,16,32], students and exercises [10,23]. The students' solutions are usually not analyzed and interpreted. Although ITS try to "understand" the content of a student solution so that they can give detailed feedback to students, they hardly use the content of students' solutions for advanced analyses (see also next section). The most advanced analysis of data collected by ITS consists in the forecast of possible grades based on the historic performance a student has shown during a course [10,18]. This is quite surprising as Business Intelligence methods are mature, widely known and applied in business applications ([5], p. 1-3). There exist standard procedures that can be adapted to the context of e-learning [34]. Whatsoever, existing ITS do not compare student solutions, do not provide any filter for a deeper analysis of solutions not to mention an analysis of which solutions parts are often done wrong.

Interestingly, although general metrics like the number of attempts, or the average processing time, cannot only be used for one course or exercise, but on an aggregated level for a set of courses and among exercises used in different courses, existing ITS even do not provide this feature. Yet, such information on an aggregated level and/or as a time series can provide interesting insights to lecturers. Lecturers usually try to improve their teaching and change explanations and teaching methods. In addition, courses may be divided into subgroups taught by different lecturers. Hence, information aggregating different courses and/or exercises over time can be helpful to lecturers to better adjust their teaching methods. Thus, this paper aims to answer the following research questions:

RQ1: Which kind of analyses of student solutions are useful for lecturers?
RQ2: How can such analyses be realized in a generalized analytics tool?
RQ3: How can different exercises be analyzed in a unified way?

To answer these research questions, we developed an e-learning Analytics Tool (EAT) that acts as an add-on for ITS. Our implementation is open source and can be found at github [14]. Particularly, EAT is built for SQL and Excel exercises but can be adapted to analog kinds of exercises like programming or mathematical transformations. EAT is an IT-artifact that provides general key performance indicators (KPI) concerning courses, exercises, and students and allows sophisticated analyses of the student solutions with regard to their content. It consists of two parts: a dashboard that displays pre-calculated KPIs and a flexible analytics board with which it is possible to analyze student solutions in detail concerning solution parts that were done right or wrong. The analyses can be done for courses, exercises, students, and different time periods without limitations in any of those dimensions. They are not bound to a specific kind of exercise, but can easily be transferred to new types of exercises. However, we will demonstrate EAT for SQL exercises. In addition, the analyses are flexible and can be defined by any lecturer. The analyses are oriented to classic BI systems and therefore allow the commonly known BI functions like drill-down etc. To the best of our knowledge, EAT is the first analytics tool that provides BI analyses to lecturers which focus on the content of student solutions and not only the metadata of courses, exercises, and student performance.

For the development of the IT-artifact, we used the design science approach (DSA) introduced by Hevner et al. [19]. For this, we orientated to the DSA processes of Conboy, Gleasure, and Cullina [8] and Peffers et al. [30] as well as to the recommendations given by Gregor and A. Hevner [17]. Hence, the remainder of this paper is organized as follows: In the next section, we review the related literature and which analytic tools have already been introduced in ITS. In section 3, requirements for the IT-artifact are derived, and the architecture of EAT is presented. Particularly, as the analyses of EAT are not restricted to a special kind of exercises, a general representation of e-learning exercises has to be developed. Additionally, some technical solutions are chosen. section 4 introduces the IT-artifact in detail which is evaluated in section 5 by interviewing 16 lecturers. The paper closes with a discussion of the results and an outlook on future developments.

2 Related Work on Intelligent Tutoring Systems

2.1 Structured Literature Review

For the structured literature review [38,39], we developed the relevant key terms for the search string and identified the database for our structured literature review in the first step. In the second step, we run the literature review with the following search string for published journal papers and conference proceedings between 2010 and 2021: *"("Intelligent Tutoring System OR Learning Analytics ("Intelligent Tutoring System" AND "Learning Analytics") OR ("Learning*

Analytics" AND "Dashboard") OR ("Intelligent Tutoring System" AND "Dashboard") OR ("Intelligent Tutoring System" AND "Education") OR ("Learning Analytics" AND "Education") OR ("Intelligent Tutoring System" AND "Feedback") OR ("Learning Analytics" AND "Feedback") OR ("Dashboard" AND "Education")". In total, we identified 393 publications. After excluding duplicates and non-peer-reviewed papers, the analysis of title and abstract reduced the number of papers to 64 that were read entirely. Of these 64 papers, only 16 papers are relevant for our study. An additional backward and forward search found four additional publications leading to a total of 20 relevant publications.

2.2 Literature Discussion

Former studies investigated the application of an analytics tool for e-learning exercises in different courses like math courses [1,20,28,42], information system courses [3,4,9,21], computer science and programming courses [2,10,16,18,23,32] and general science courses [6,7,11,13,25,26]. However, all studies used generic KPIs for the analysis like the number of participants, downloaded course material, the number of text messages between students [4,16,32], usage of group work or comment options [6,7,10,16,23,32,42], students course working activity in hours, the processing time for an exercise, results (correctness) of students submission, or the number of code posts or commits [10,23]. Although such generic KPIs are applicable for different courses or exercises, none of the former studies investigated this aspect of aggregation but solely on one single course.

Furthermore, former studies neither tried to analyze the content of student solutions nor utilized flexible adjustable metrics (e.g., drill-downs). Subsequently, analyses of multiple exercises types (e.g., Excel, SQL) also do not exist. In contrast, EAT focuses on the content of student solutions and analyzes for example the different parts of a submitted SQL query. This helps lecturers identify typical mistakes like students' misunderstanding of the GROUP BY clause. Instead, platforms of former studies can only sort students' results according to their exercise performance (pass or fail) [1,2,4,7,10,11,13,18,20,21,28,32,42], filter exercises which create the most problems generally [20,42], caclulate average performance of students regarding the exercises [20,42], count the number of trials, or show the course progress of students [2,4,13]. Only two studies [10,18] utilized a more sophisticated approach to forecast the students' final result based on their current course performance.

3 Objectives, Requirements & Architecture

3.1 Objectives

The main goal of EAT is to support lecturers with useful information about their courses and to provide them with a flexible analytics tool with which it is possible to examine the content of student solutions in depth. An EAT shall help lecturers to get a complete picture of students' knowledge and summarize the evaluation and assessment of student solutions. The assessment of student solutions (e.g. by running SQL queries on test databases, comparing the structure and formulas of an excel sheet, or executing unit tests on student programs) is not done by EAT but provided to EAT by the underlying ITS.

In more detail, the requirements can be distinguished into *general requirements* and *specific analysis abilities*. The *general requirements* comprise usability and technical factors. An ITS should be user-friendly, ease of use and simple [24]. But, usability does not only mean that the tool is easy to use but content wise that it delivers immediate benefits for the lecturer with standard analyses (e.g., overview of processing rates of exercises) which answer typical questions of the lecturer (see section 4.1). Often, standard analyses can be tailored with the help of further parameters. But too many such parameters make a frontend confusing. Hence, parameters should initially be set to useful default values. Only in a separate display, additional parameters should be adaptable. (For similar reasons, almost all search sites have a standard search and an advanced search [40]) Then, individual, flexible in-depth analyses can start from this second display on [32]. Like business information, e-learning data and the according in-depth analyses can become very complex ([5], p. 113). Also, the data volume is quite big. During one course, it is not uncommon that several ten thousand data sets are generated. Hence, suitable methods are necessary to cope with this complexity. Business Intelligence (BI) has brought up several techniques like Pivot Tables and Pivot Charts with which it is possible to summarize, analyze, explore, and present complex data accordingly. Therefore, an EAT for ITS should orientate to these well-established methods (see section 3.2) in order to ensure acceptable performance of analyses within a few seconds ([5], p. 112).

This also helps to deal with different exercise types. The EAT presented in this paper is not bound to one but several types (e.g. SQL, Excel, mathematics). Thus, the user interface, the underlying analysis architecture and data structure have to be generic so that different data can be stored, processed and analyzed.

The specific *analysis abilities* concern the content of analyses, i.e. what kind of information should be provided so that lecturers are supported in the teaching task. Besides a quick overview of what happens in courses (e.g.[7]), answers to the following questions are frequently demanded.

- Which mistakes occur most often? What is the share of students who understood the topic? [20,42]
- How many attempts and time do students need for solving an exercise? [7]
- How many students have solved which exercises? Which are difficult? [20]
- What are the expected (final) results of students based on their current performance concerning the exercises? [10,18]

3.2 Evaluation Platform Architecture

A main requirement for the analytics tool is to deal with different kinds of exercises. This can be compared with an integration of different information systems into one analytics solution. In both cases, the data of the different sources may differ greatly. Therefore, a generic architecture is needed with which it is possible to incorporate all the different data from different sources. A common approach to solve this problem is to use a reference architecture. However, in the area of learning analytics, we only found the concept of a value chain [34] which does not suit the needs of EAT.

Therefore, we employ a so-called data warehouse used for BI systems to get a single point of view (SPOT) combining all relevant data. A common reference architecture for BI systems is described in [33]. It consists of so-called *source systems*, a *data warehouse system* and a *BI platform* for data analyses. The *source systems* are the systems that generate the data to be analyzed. The *data warehouse system* is composed of four layers. First, incoming data is loaded from a *staging area* and restructured/corrected in a *cleansing area* before being stored in a *core* in a subject-oriented manner. In preparation for evaluation data is aggregated and stored in *data marts.* In addition, the architecture consists of a meta-data manager that stores information on all operations with a timestamp.

Comparing the BI reference architecture with our use case EAT, the source systems are represented by the different exercise types which are realized by so-called checkers in the underlying ITS. These contain the business logic for assessing student exercise submissions and provide the results in a database. APIs to the ITS are the staging area in EAT. The cleansing area has the goal to prepare the data. In other words, the process consists of two core aspects. The first is the detection and correction of defects within the data. Much more important is the second point, the transfer into a generic data schema. Technically, the cleansing area of a BI system and EAT is very similar.

The part of EAT corresponding to the BI core is the database where all information is stored. The data scheme is depicted in Fig. 1. Exercises belong to courses which lie in the responsibility of lecturers. Students can submit solutions to an exercise. As exercises can be of different subjects (e.g. SQL, Excel) with different structures, solutions (i.e. submissions) to an exercise are generically deconstructed into their basic parts, called attributes. Attributes are subject specific. In the case of SQL exercises (e.g. "SELECT * FROM employees"), a possible attribute can represent the table names of an SQL query. Thus, the characteristic "attribute" of the class "Attribute" gets the value "tables". The tables that are really used in a solution are then stored in the characteristic "values", in our example, characteristic "value" has the value "employees". Whether the attribute can be used to get a valid solution is stored in the "status" field. This helps to analyze if parts of solutions are correct or wrong. With this deconstruction of exercise parts, it is possible to represent different exercise types within the model and to analyze the exercises then in a unique way.

The counterpart of data marts in the BI world, is the analytics level of EAT. First, analysis-related subsets are extracted from the core and made available. Aggregated key figures are calculated in order to make them instantly available. As is also common in BI systems, there is also a metadata management in place. This is important, for example, in order to record the checker versions used to evaluate the student solutions. In this way, the evaluation of student solutions

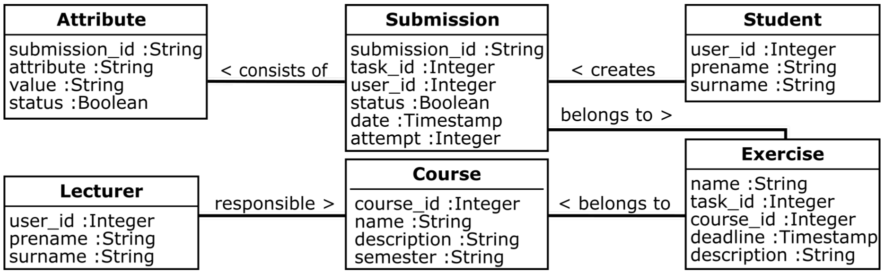


Fig. 1. Generic Data Model

can be updated if a better checker is available. Furthermore, the evaluation data is considered as a so-called *slowly changing dimension* of *Type 2* where a BI system stores history data on all information. In doing so, the progress in the checkers can be analyzed, too.

3.3 Capabilities for an E-Learning Analytics Tool (EAT)

For the realization of the functional requirements, several technical solutions already exist that differ in several aspects. Therefore, the following technical requirements have been set up to choose an application that suits EAT:

Embedding. The application should be able to be integrated into an existing web application in order to be able to use EAT within the ITS.

Customizing. In particular, it is important that the graphical components can be adapted to the existing design in the ITS system. In addition to visual integration,the product must be technically integrable into the existing frontend.

Data volume. Various interface options should allow for the connection of any data source. Data size should not be a limiting factor in any way that confines the usability of potential data sources.

License model. The application should be free of charge as this increases the attractiveness and usability and because the existing ITS is also open source.

Functional requirements. The application should enable the graphical display of data, the visualization in the form of tables as well as complex filtering without having to resort to other applications.

Based on these requirements, KNIME (Individual)[1], Grafana (Free)[2], PlotlyJS[3], ChartJS[4], WebDataRocks[5], and Plotly Dash[6] were analyzed to evaluate their usage for EAT. Any available paid functions were not taken into account.

In particular the license model is a limiting factor. KNIME and Grafana cannot provide the desired functions in the free variants. Since PlotlyJS and ChartJS

[1] https://www.knime.com/
[2] https://grafana.com/
[3] https://plotly.com/javascript/
[4] https://www.chartjs.org/
[5] https://www.webdatarocks.com/
[6] https://plotly.com/dash/

are only charting libraries, these must be supplemented by a table function. Web Data Rocks could serve as such a table tool, but due to the limited data size it represents a bottleneck so that these combinations are excluded from further considerations. PlotlyDash provides all requested features. It is a low-code framework programmed with Python. Being not an application, additional programming efforts result. Nonetheless, this has a positive effect on customization. Despite being a free version, there are no limitations on the amount of data. These large quantities can also be processed efficiently and in real time. The Dash DataTable module enables pivot tables to be displayed, which offer a second form of analysis in addition to the integrated graphics. The development of the prototype is therefore carried out in Plotly Dash.

4 The E-Learning Analytics Tool (EAT)

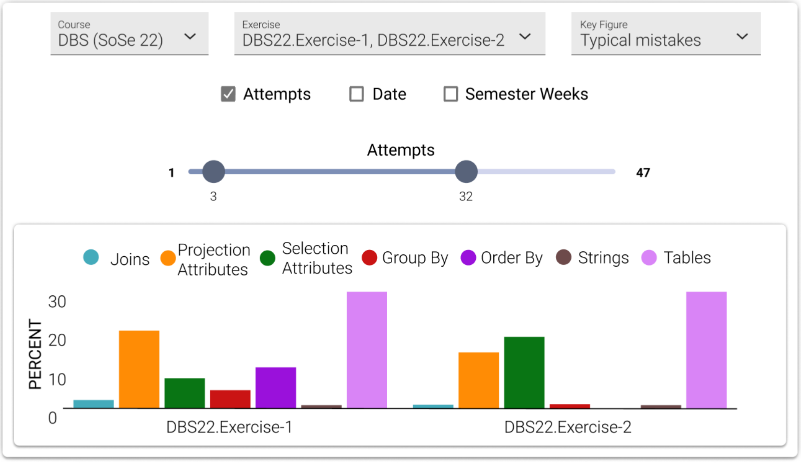
4.1 Dashboard for General Information

The first part of EAT consists of a dashboard where standardized information about courses is presented in a Pivot Table and via a Pivot Chart (see Fig. 2). In the first step, the course(s) and the exercise(s) have to be selected for the analysis. Then, a key figure is chosen and the details of which are displayed in the Pivot Chart. For simplicity, data is not displayed in a time series. Instead, the analysis can be restricted to different time periods (number of attempts, date, semester week) with the help of a slider. This helps to check exercises done during a lesson or to monitor the learning progress during a semester by changing the slider and observing the changes in key figures. The key figures below are supported.

Processing rate, average attempts and facility index. These key figures are presented in a matrix showing students (possibly grouped by tutorial, degree program, etc.) in the rows, exercises (possibly grouped by learning target, issue time, etc.) in the columns and the key figures in the cells. At a glance, you can see which exercises are solved by which students and how many attempts students have needed. Student and exercise groups in the matrix allow us to focus on different tutorial and learning goals, respectively. The difficulty index is defined similarly to [15] as the ratio of correct to incorrect submissions. This provides lecturers with an indication if exercises have an appropriate level of difficulty.

Earned points and grades. The gradings of the student solutions are also shown in a matrix. Exercises can be subdivided into mandatory, voluntary, and bonus exercises as well as grouped by learning goals, weekly, etc. Moreover, rules can be set up to define when an exam admission or course is passed. For each student, the grading overview shows the result from evaluating the rules and summarizes the points in one exercise group, but one can also see the detailed points.

Typical errors. To identify typical weaknesses of students and to be able to align a course more precisely to the individual students, lecturers can have a look at the frequency of errors students made when solving exercises. The errors investigated in the dashboard are specific to an exercise type like SQL or Excel. They correspond to the attributes described in the previous section. For each



Course: DBS (SoSe 22)

Exercise: DBS22.Exercise-1, DBS22.Exercise-2

Key Figure: Typical mistakes

☑ Attempts ☐ Date ☐ Semester Weeks

Fig. 2. Dashboard

selected exercise, a bar chart is generated, displaying the percentage of errors for each attribute. There are various additional parameters for better insights.

4.2 Flexible Analytics Board

The Flexible Analytics Board (FAB) allows lecturers to dive deeper into the data than it is possible with the dashboard. Similarly to the dashboard, filters can be defined for courses and exercises. Additionally, it is possible to use filters on subject-specific characteristics. The example shown in Fig. 3 shows a flexible analysis for SQL exercises where the possible SQL-specific attributes are displayed as filters. For Excel, attributes such as specific operators (e.g. sum) are displayed. Regardless of the subject-specific exercises, the various attributes can be moved to a so-called background filter or to the columns.

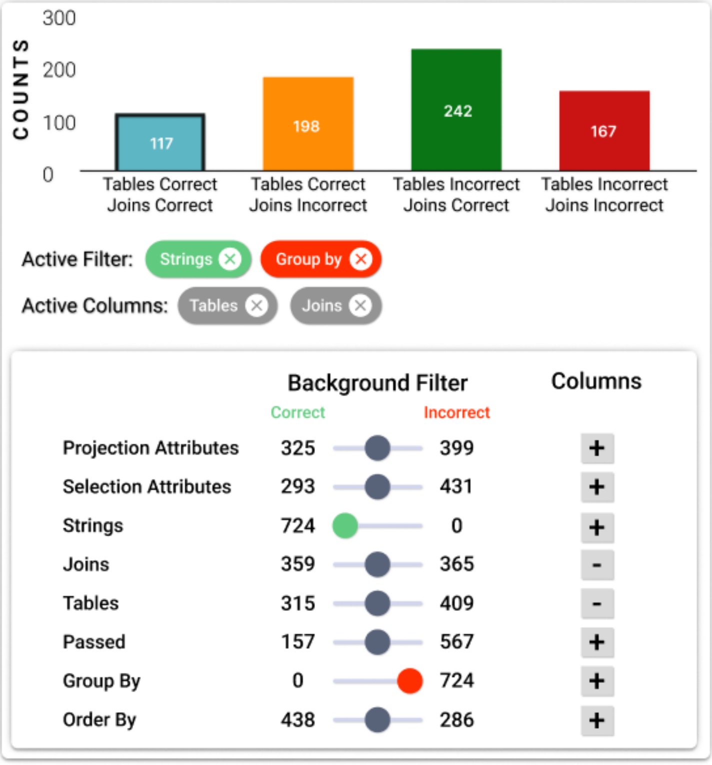
Background filter. By adding a background filter, it is possible to set conditions for the displayed data. For example, all erroneous GROUP BY-clauses can be filtered on the condition that the used STRING-comparisons are correct. Thus, attributes GROUP BY and STRING are placed in a background filter. Another background filter is the possibility to filter by the attempts (e.g., consider only attempts 1-3) or by time aspects. This is done similarly to the dashboard but displayed in Figure 3.

Columns. Attributes can be further analyzed with the help of a bar chart. If one attribute is selected, one bar displays the number of correct and another one the number of incorrect student solutions. If several attributes are included as columns, there is a bar for each possible combination of the attributes.

To analyze the data in more detail, it is possible to drill down to the concrete entries behind one of the bars. To do this, the bar must be clicked (first bar in Fig. 3), whereupon a table opens with all concrete student submissions. Also, checker related information can be provided that is specific to a particular type of exercise. Concerning SQL queries for example, the checker compares the result set of a student's SQL query with a correct query in a test database to check the correctness of the student's SQL query. These result sets can be displayed. Such data is also generated during other subject-specific exercises. For example in Excel exercises, consequential errors are analyzed to identify the source of the problem. The goal is to provide the lecturer with further information at this point, which has been generated by the subject review. Typical use cases are:

Classify typical solutions. Students often make the same mistakes. To know these mistakes is a prerequisite to react and explain the according topic during courses. With the help of the FAB, it is possible to isolate these typical errors by filtering the solutions with a combination of different attributes. For example, Figure 3 shows that students make frequent errors on the Joins attribute. Digging deeper and looking additionally at the tables used in a query, the join mistakes also appear when the correct tables are used. Hence, a misconception of the join clause is quite likely

Focus on important errors and hide unimportant errors. Student mistakes stem from different causes. While some mistakes reveal a lack of under-


COUNTS
300
200
100
0
117
198
242
167
Tables Correct Joins Correct
Tables Correct Joins Incorrect
Tables Incorrect Joins Correct
Tables Incorrect Joins Incorrect

Active Filter: Strings ⊗ Group by ⊗

Active Columns: Tables ⊗ Joins ⊗

	Background Filter		Columns
	Correct	Incorrect	
Projection Attributes	325	399	+
Selection Attributes	293	431	+
Strings	724	0	+
Joins	359	365	-
Tables	315	409	-
Passed	157	567	+
Group By	0	724	+
Order By	438	286	+

Fig. 3. Flexible Analytics Board

standing of the general concepts, some mistakes are read or transcription errors. To distinguish between these categories, solutions can be filtered by fixing the value of an attribute and moving this attribute to a background filter. Looking again at the example in Figure 3, before focusing on the join operator, it should be clear that the correct tables are used in the SQL query. Otherwise, the join operator can hardly be correct without using the correct tables. Thus, it makes sense to set tables to ”correct” in a background filter before analyzing the joins. In the field of Excel, this function can help to detect the source of errors in case of subsequent errors.

Show concrete solutions and/or explanations. Often, mistakes can easily be seen when looking directly at the student solution. Hence, for SQL, the comparison between the result sets on the test database of a student solution and a correct solution can be very helpful. For Excel, the incorrect fields should be shown with additional information. This includes, for example, a comparison of the dependency structure in the student solution and a correct solution.

5 Evaluation

Following the design science approach, we evaluate the validity and utility of our proposed artifact [17]. The evaluation was conducted in two ways: First, we collected lecturers' comments regarding the artifact with a semi-structured group interview [12]. The group interviews were conducted by two researchers and lasted 10 between 20 minutes. In each interview, five lecturers participated. Five interviews were conducted online via zoom and seven in presence. Second, after the group interviews, each participating lecturer was invited to fill out the standardized questionnaire regarding the perceived usefulness and value of the artifact. In total, 16 lecturers from two different modules (business administration and information systems: 4; database systems: 12) participated in the group interviews and the questionnaire.

In the group interview section, the lecturers were asked to evaluate their expectations and personal options regarding EAT. Generally, the lectures demand the provision of a good overview of EAT (e.g., to see the submitted exercise or processing time of each student for each exercise). They also expect a high level of ease of use so that their interaction with the artifact is clear. Lecturers want to make only a few settings (e.g., select filters) to analyze students' exercise solutions. Hence, aside from usability, EAT should help them to quickly analyze students' submissions. Moreover, lecturers want to analyze exercises according to the level of difficulty with an accurate error display. In particular, they highly expected a detailed analysis of students' typical mistakes to promote students' learning progress. Additionally, the lecturers also expected that the EAT should be suitable for various courses at the university.

Most of the interviewed lecturers have a positive opinion about the EAT (average score for each statement of the questionnaire in parentheses). The lecturers assessed EAT as useful in general. They agreed that the dashboard interface is intuitive (3.6) and not overloaded with information (4.2). Moreover, they consider the interface as easy to understand (3.7). In their eyes, EAT provides a good overview which helps to identify what needs to be improved in the learning content and in optimizing existing exercises (average 4.2). They regard EAT as being useful for tracking students' learning progress so that they can determine how they can help students to improve their learning outcomes (3.8) and individually give students more efficient feedback related to their learning situation (3.6). The lecturers valued the flexible filtering options like filtering for students' attempts or the time period (e.g., the third week in a course) as being useful to carry out flexible analyzes for individual students or tasks (3.6) and to better understand and evaluate the solutions that have been submitted (4.2) (e.g., detailed filtering for part of an SQL query). Nevertheless, criticism was also expressed, namely the missing design aspects (e.g., front, labels unclear), or the inappropriateness of the color scheme for people with red-green color blindness.

6 Conclusion

To answer our research questions, we developed an IT artifact (EAT) that aims to analyze different kinds of exercises. Following the design science approach, several iterations of design, development, and evaluation have been run through leading to a gradually improved system according to the needs of lecturers.

Concerning *RQ1: Which kinds of analyses of student solutions are useful for lecturers?*, the results of our two surveys show that standard analyses as well as flexible analytics options are expected and valued by lecturers in equal measure. With regard to standard analyses, this aligns with former studies which highlighted the necessity of a good overview with exercise-related information like processing time, results (correctness) of students' submission [10,23], sorting students' results according to their exercise performance (e.g., [1,2,4,7]) or counting the number of trials [2,4,13]. In contrast, flexible analytics options have been investigated for the first time and were considered as highly practical and helpful. Lecturers valued the possibility to better assess their students' domain knowledge (e.g., what the students need to know) and in particular the flexible filtering option. This helps lecturers to understand students' mistakes better and subsequently to tailor their feedback on students' performance and how to proceed, which is extremely valuable for the student's learning process [22].

The answer to *RQ2: How can such analyses be realized in a generalized analytics tool?* consists in the usage of a reference architecture that is derived from BI applications. Already one kind of exercise like SQL is composed of a variety of elements, not to speak of several kinds (see RQ3 below). In addition, different courses contain a plethora of different exercises. To successfully cope with this situation, the BI concept was employed to the e-learning analytics tool. For this, a generic data schema was developed which disassembles exercises into their basic parts. This not only allows general analyses of exercises on a dashboard concerning average attempts, processing time, or facility index for example like in many other e-learning tools [1,2,4,7,13,18,20,21,28,32,42], but also analyses of the content of students' solutions with the help of a flexible analytics board. For the first time, profound analyses of student mistakes are possible with the help of drill-down functions.

With regard to RQ3: *How can different exercises be analyzed in a unified way?*, a side effect of the generic data schema is that the analytics tool is not bound to a specific kind of exercise but can cope with different kinds like SQL, Excel, or mathematics. The reason is that the exercise structure is disassembled into its basic parts and generalized. These basic parts are called attributes. For example, in SQL exercises, the joins are an attribute while in Excel exercises an attribute may be the color scheme of cells. This approach opens the analytics board to a variety of exercises. Each exercise type just needs to be decomposed into its parts. Integrating EAT into ITS creates a single touchpoint for lecturers. Student submissions are collected and evaluated within a single tool. EAT relieves lecturers from a huge amount of recurring standard tasks with the standard and flexible analyses. It accelerates the evaluation process of student submissions as submissions are automatically assessed by the ITS and lecturers can directly make use of the analytics functions to intervene in a course's learning process and improve their teaching. The flexible analyses enable tailored feedback to single students and the whole course. This in turn improves students' learning outcomes and empowers them as self-regulated learners.

Future developments concern lecturers as well as students. For students, notification functions are planned which remind students to keep up with their learning (e.g., continue with your course-related exercise) or which highlight knowledge gaps (e.g., "you need to improve your understanding of the GROUP BY clause"). Additionally, due to the positive effect of learning partners [41] a recommendation function for possible learning partners shall be integrated which is based on students' performance. For lecturers, downloads of anonymized reports shall be available that can be used for example in lectures to show and discuss common mistakes (e.g., If function problems in Excel). Regarding the limitations of our study, first, the data sample revealed vital aspects for the development of our tool, but is also small and needs to be increased in future research. Second, we only interviewed lecturers from the information system context. Future studies should include lecturers from different subjects like mathematics.

References

1. Aleven, V., Xhakaj, F., Holstein, K., McLaren, B.M.: Developing a teacher dashboard for use with intelligent tutoring systems. technology **34**, 44–50 (2010)
2. Ali, L., Hatala, M., Gašević, D., Jovanović, J.: A qualitative evaluation of evolution of a learning analytics tool. Computers & Education **58**(1), 470–489 (2012)
3. Alzoubi, D., Kelley, J., Baran, E., B. Gilbert, S., Karabulut Ilgu, A., Jiang, S.: Teachactive feedback dashboard: Using automated classroom analytics to visualize pedagogical strategies at a glance. In: CHI EA '21 (2021)
4. Arnold, K.E., Pistilli, M.D.: Course signals at purdue: Using learning analytics to increase student success. In: LAK '12. pp. 267–270 (2012)
5. Baars, H., Kemper, H.G.: Business Intelligence & Analytics - Grundlagen und praktische Anwendungen: Ansätze der IT-basierten Entscheidungsunterstützung. Lehrbuch, Springer Vieweg (2021)

6. Barr, J., Gunawardena, A.: Classroom salon: A tool for social collaboration. In: SIGCSE '12. p. 197–202 (2012)
7. Cobos, R., Gil, S., Lareo, A., Vargas, F.A.: Open-dlas: An open dashboard for learning analytics. p. 265–268. L@S '16, ACM (2016)
8. Conboy, K., Gleasure, R., Cullina, E.: Agile design science research. In: Donnellan, B., Helfert, M., Kenneally, J., VanderMeer, D., Rothenberger, M., Winter, R. (eds.) DESRIST 2015. pp. 168–180. Springer (2015)
9. Dawson, S., Bakharia, A., Heathcote, E., et al.: Snapp: Realising the affordances of real-time sna within networked learning environments. Networked Learning (2010)
10. Diana, N., Eagle, M., Stamper, J., Grover, S., Bienkowski, M., Basu, S.: An instructor dashboard for real-time analytics in interactive programming assignments. In: LAK '17. p. 272–279 (2017)
11. Dickler, R.: An intelligent tutoring system and teacher dashboard to support mathematizing during science inquiry. In: Isotani, S., Millán, E., Ogan, A., Hastings, P., McLaren, B., Luckin, R. (eds.) Artificial Intelligence in Education. pp. 332–338. Springer (2019)
12. Döring, N., Bortz, J.: Forschungsmethoden und evaluation. Wiesbaden: Springer (2016)
13. Dourado, R.A., Rodrigues, R.L., Ferreira, N., Mello, R.F., Gomes, A.S., Verbert, K.: A teacher-facing learning analytics dashboard for process-oriented feedback in online learning. In: LAK21. p. 482–489 (2021)
14. Feedbacksystem: Intelligent, personalized feedback for students using artificial intelligence, https://github.com/thm-mni-ii/feedbacksystem
15. Gamage, S.H.P.W., Ayres, J.R., Behrend, M.B., Smith, E.J.: Optimising moodle quizzes for online assessments. Int J STEM Educ **6**(1) (2019)
16. Govaerts, S., Verbert, K., Duval, E., Pardo, A.: The student activity meter for awareness and self-reflection. CHI EA '12, ACM (2012)
17. Gregor, S., Hevner, A.R.: Positioning and presenting design science research for maximum impact. MIS Q **37**(2), 337–355 (2013)
18. Herodotou, C., Maguire, C., McDowell, N., Hlosta, M., Boroowa, A.: The engagement of university teachers with predictive learning analytics. Computers & Education **173**, 104285 (2021)
19. Hevner, A.R., March, S.T., Park, J., Ram, S.: Design science in information systems research. MIS Q. **28**(1), 75–105 (2004)
20. Holstein, K., Xhakaj, F., Aleven, V., McLaren, B.: Luna: a dashboard for teachers using intelligent tutoring systems. Education **60**(1), 159–171 (2010)
21. Hu, X., Hou, X., Lei, C.U., Yang, C., Ng, J.: An outcome-based dashboard for moodle and open edx. In: LAK '17. p. 604–605 (2017)
22. Keuning, H., Jeuring, J., Heeren, B.: A systematic literature review of automated feedback generation for programming exercises. ACM Trans. Comput. Educ. **19**(1), 1–43 (2019)
23. Leony, D., Pardo, A., de la Fuente Valentín, L., de Castro, D.S., Kloos, C.D.: Glass: A learning analytics visualization tool. In: LAK '12. p. 162–163 (2012)
24. Li, D., Zhou, H.H.: An intelligent tutoring system with an automated knowledge acquisition mechanism. In: 2015 IEEE International Conference on Computational Intelligence & Communication Technology. pp. 88–91. IEEE (2015)
25. Martinez-Maldonado, R.: A handheld classroom dashboard: Teachers' perspectives on the use of real-time collaborative learning analytics. Int J Comput Support Collab Learn **14**(3), 383–411 (2019)

26. Martinez Maldonado, R., Kay, J., Yacef, K., Schwendimann, B.: An interactive teacher's dashboard for monitoring groups in a multi-tabletop learning environment. In: Cerri, S.A., Clancey, W.J., Papadourakis, G., Panourgia, K. (eds.) Intelligent Tutoring Systems. pp. 482–492. Springer (2012)
27. Mitrovic, A.: An intelligent sql tutor on the web. Int J Artif Intell Educ **13**(2-4), 173–197 (2003)
28. Molenaar, I., Knoop-van Campen, C.A.N.: How teachers make dashboard information actionable. IEEE Trans Learn Technol **12**(3), 347–355 (2019)
29. Patel, A., Kinshuk: Intelligent tutoring tools-a problem solving framework for learning and assessment. In: Fie '96. vol. 1, pp. 140–144 vol.1 (1996)
30. Peffers, K., Rothenberger, M., Tuunanen, T., Vaezi, R.: Design science research evaluation. In: Peffers, K., Rothenberger, M., Kuechler, B. (eds.) Design Science Research in Information Systems. Advances in Theory and Practice. pp. 398–410. Springer (2012)
31. Sadler, D.R.: Formative assessment: revisiting the territory. Assess Educ **5**(1), 77–84 (1998)
32. Santos, J.L., Govaerts, S., Verbert, K., Duval, E.: Goal-oriented visualizations of activity tracking: A case study with engineering students. In: LAK '12. p. 143–152 (2012)
33. Schnider, D., Jordan, C., Welker, P., Wehner, J.: Data Warehouse Blueprints: Business Intelligence in der Praxis. Hanser (2016)
34. Shankar, S.K., Prieto, L.P., Rodriguez-Triana, M.J., Ruiz-Calleja, A.: A review of multimodal learning analytics architectures. In: ICALT 2018. pp. 212–214. IEEE (2018)
35. Siepermann, M.: Lecture accompanying e-learning exercises with automatic marking. In: E-Learn: World Conference on E-Learning in Corporate, Government, Healthcare, and Higher Education. pp. 1750–1755. Association for the Advancement of Computing in Education (AACE) (2005)
36. Siepermann, M., Börgermann, C., Lackes, R.: Question-and-answer based explorative elearning exercises. In: ECGBL2010. p. 360. Academic Conferences Limited (2010)
37. Thomas, P., Smith, N., Waugh, K.: Automatic assessment of sequence diagrams (2008)
38. Vom Brocke, J., Simons, A., Riemer, K., Niehaves, B., Plattfaut, R., Cleven, A.: Standing on the shoulders of giants: Challenges and recommendations of literature search in information systems research. Commun. Assoc. Inf. Syst. **37**(1), 9 (2015)
39. Webster, J., Watson, R.T.: Analyzing the past to prepare for the future: Writing a literature review. MIS Q **26**(2), xiii–xxiii (2002)
40. White, R.W., Morris, D.: Investigating the querying and browsing behavior of advanced search engine users. In: Kraaij, W., de Vries, A.P., Clarke, C.L.A., Fuhr, N., Kando, N. (eds.) SIGIR '07. pp. 255–262. ACM (2007)
41. Woolf, B.P., Arroyo, I., Muldner, K., Burleson, W., Cooper, D.G., Dolan, R.P., Christopherson, R.: The effect of motivational learning companions on low achieving students and students with disabilities. In: Intelligent Tutoring Systems (1). pp. 327–337 (2010)
42. Xhakaj, F., Aleven, V., McLaren, B.M.: Effects of a dashboard for an intelligent tutoring system on teacher knowledge, lesson plans and class sessions. In: André, E., Baker, R., Hu, X., Rodrigo, M.M.T., du Boulay, B. (eds.) Artificial Intelligence in Education. pp. 582–585. Springer (2017)