

Extending the AAT Tool with a User-Friendly and Powerful Mechanism to Retrieve Complex Information from Educational Log Data^{*}

Stephen Kladich, Cindy Ives, Nancy Parker, and Sabine Graf

Athabasca University, Canada

sjkladich@hotmail.com, {cindy, nancyp, sabine}@athabascau.ca

Abstract. In online learning, educators and course designers traditionally have difficulty understanding how educational material is being utilized by learners in a learning management system (LMS). However, LMSs collect a great deal of data about how learners interact with the system and with learning materials/activities. Extracting this data manually requires skills that are outside the domain of educators and course designers, hence there is a need for specialized tools which provide easy access to these data. The Academic Analytics Tool (AAT) is designed to allow users to investigate elements of effective course designs and teaching strategies across courses by extracting and analysing data stored in the database of an LMS. In this paper, we present an extension to AAT, namely a user-friendly and powerful mechanism to retrieve complex information without requiring users to have background in computer science. This mechanism allows educators and learning designers to get answers to complex questions in an easy understandable format.

Keywords: Learning Analytics, Learning Management System, Log Data.

1 Introduction

Learning management systems (LMSs) collect and store a great deal of data about learners and learning materials/activities along with how learners interact and utilize these materials/activities [1]. As learners access more and more learning material/activities online, and as LMSs collect more and more data about learners and their interactions, analysing and understanding that data has become critical for many reasons. For instance, this data can inform educators and learning designers about how students learn and how they use the provided materials/activities, leading to valuable insight into student behaviours and enabling educators and learning designers to better adapt learning content and teaching strategies to student needs. Such insights may lead to better understanding of the effectiveness of course designs and teaching strategies, and thus to better student satisfaction and increased student performance from improved learning designs and teaching strategies.

^{*}The authors acknowledge the support of Athabasca University and NSERC.

The concept of analysing learning system data is not new. Rudimentary analytics have been employed since the beginning of computer-based training [2], for example to add a degree of personalization to the learner's experience. However, when dealing with aggregated data of a high number of users in a large instructional setting, more advanced techniques and tools are required to effectively understand learners' interactions with the course materials and resources of the LMS, particularly when analyzing such data across entire programs of study. The large amount of data generated is often stored in complex log files or databases, and therefore, specialized techniques are required to process the data and derive meaningful information. These techniques are often not within the skill-set of instructional designers and educators as they are highly technical or computational in nature and require specific skills and tools (such as knowledge of database schema/SQL). Therefore, such educational log data are typically not accessed by learning designers and educators, and thus, little feedback is available for them on whether their teaching strategies and learning designs are actually helping students. Siemens [3] writes that the field of learning analytics (and its related research) must make such activities relevant to educators and administrators – this is only possible if tools are developed that specifically aim at this group to help them understand, evaluate and act on the data that learning systems generate.

Given this skills gap, researchers have started to design and build solutions to assist educators in understanding the data in their learning systems. Tools such as GLASS [4], LOCO-Analyst [5], and CosyLMSAnalytics [6] have been developed to meet this need, however they all have either complex user interfaces, are specific to only one LMS, only extract data without providing even basic analytical functionality, or require other specialized software to be installed.

This paper focuses on the Academic Analytics Tool (AAT) [7]; a browser-based application that can access and report on the data generated by any LMS. While a general description of the tool is provided by Graf et al. [7], the aim of this paper is to introduce an extension of the tool which increases the user-friendliness of AAT and at the same time provides users with advanced access to complex information from educational log data.

The paper is structured as follows: Section 2 provides a brief overview of AAT. Section 3 presents the proposed extension of the tool and Section 4 concludes the paper and discusses future work.

2 Overview of AAT

AAT allows users to access and analyse educational log data from any LMS. It uses input data from one or several databases of an LMS, extracts and analyses the data that are specified by users, stores the results within the Academic Analytics database, and allows exporting the results as CVS and HTML files to be further used by visualization or advanced statistical tools. In the following paragraphs, the main design decisions and building blocks of the tool are described to provide an overview of AAT.

In order for users to be able to specify what data they are interested in accessing and analysing, one of the main building blocks of AAT is the notion of *concepts*, which refer to logical constructs of interest to the user (such as a course, discussion forum, quiz etc.). Most of these concepts are *learning objects* (e.g., discussion forum,

quiz, resource, etc.) and therefore, have a pedagogical type which defines their educational purpose (e.g., a quiz could be for self-assessment or graded).

While concepts only allow users to specify the area in which they are interested, *patterns* allow them to specify in more detail what data they exactly are looking for. Patterns are similar to reports or queries and can be simple lists of concepts (e.g., a list of students who have posted more than 5 times in a forum) or an analysis/calculation on concepts (e.g., the average amount of time students spent on quizzes).

In order to make the tool applicable for different LMSs, *templates* have been introduced, which can be seen as the interface between the tool and the databases. While patterns specify what data should be extracted from a database, templates specify where (i.e., what tables and columns) the respective data resides within the database of a particular learning system, considering the version of the system (e.g., Moodle 2.2). Each LMS version/instance has its own template, and the template is configurable to encompass any alterations or modifications to the out-of-the-box schema. Therefore, the tool can be used for an LMS even if its database schema has been modified to meet the needs of the institution.

In the context of AAT, a *database* is the underlying repository for an LMS. AAT can connect to different databases as well as to multiple databases at the same time.

Another central entity to AAT is the *dataset*. A dataset specifies one or more courses that the user is interested in. The dataset defines on which data the patterns are executed (e.g., all courses in Nursing, the course “Introduction to Biology”, etc.).

In order to allow a user to run patterns repeatedly over time on the same or different datasets and databases, *profiles* have been introduced. A profile can be seen as an experiment for extracting and analysing particular data. In a profile, a user specifies which LMS is used (through selecting a template), which courses and time spans should be investigated (through selecting the dataset), and which data the user is interested in (through selecting patterns). AAT guides the user through this specification process. Once the profile is created, it can be used to extract and analyse the specified data.

Another important feature of AAT is *user management*, including user roles and the authentication process. All users authenticate to the tool via built-in login or via CAS [10]. The authentication method is set via the application's configuration file. There are two principle types of users in AAT: Administrators and Standard Users. Administrators can add or edit templates and are assumed to have an understanding of the LMS database's schema. Administrators also define the database connections the tool uses to access LMS data, including the authentication credentials. Standard Users (e.g., educators, course designers) can use the tool to select an LMS to connect to, select a dataset, create/edit/store/delete patterns, and create/edit/store/delete profiles. Standard users are not expected to have any knowledge of database schema or SQL.

3 Pattern Creation Mechanism

While the previous section provided a brief overview of AAT, this section presents an advanced pattern creation mechanism that extends AAT to provide users with easy-to-use functionality to extract and analyse complex information from educational log

data. The mechanism includes three components to improve the user-friendliness and the capacity of the tool to enable the user to ask complex questions (and therefore build complex patterns) while still remaining easy to use for users without computer science background. These components include: (1) an ontology for easy use of the tool, (2) a refined approach for pattern chaining, and (3) further options and a wizard-style user interface for pattern creation. In the following subsections, these three components are presented in more detail.

3.1 AAT Ontology

Ontologies in the information systems context are utilized to represent the knowledge about a system (i.e., its parts, relationship of parts, activities, etc.), to share that knowledge, and house a vocabulary about that system [11]. Such an ontology forms the reference point for design and implementation considerations for AAT. The AAT system domain knowledge is represented by the AAT Ontology illustrated in Fig. 1. In this figure only the most relevant attributes are included as examples since a complete visualization of all the attributes is space-prohibitive. There are three key aspects to the AAT Ontology: Concepts, Databases and Templates.

Concepts represent the entities that are considered in the system to get information about and perform analysis. Each concept can have one or more attributes that contain data about this concept. In this ontology, there are five main types of concepts: User, Course, Learning Object, Message and Log. The User concept consists of three subtypes which represent common roles in an LMS: Student, Teacher, and Course Creator. Similarly, the Learning Object concept has several subtypes that encompass the types of learning objects typically used by LMSs: assignments, quizzes, forums, books, resources, wikis, blogs, surveys, and lessons. The list of learning objects, users and concepts can be easily extended if needed, making the ontology extensible.

A *Database* is the repository that includes data about the learners and their behaviour/performance (i.e., a database of an LMS) and is mapped by the template to facilitate pattern creation.

A *Template* is composed of Concept Mappings, one for each concept that is supported by the LMS. A Concept Mapping not only maps the relationship between the concept and the concept's location in the LMS database, but also the relationship between concepts. Specifically, the concept mapping includes four types of information. First, it provides information on where a concept and its attributes are stored in an LMS database, including the concrete information in terms of tables and columns. Second, it also includes information about the relationships between concept attributes and the database schema, which is critical when a concept is physically stored in multiple database tables. For example in Moodle, data about quizzes is stored in several tables, so the concept mapping for the concept quiz provides information about how these quiz tables are related to each other (e.g., which question belongs to which quiz) and how information in these tables can be found. Third, a concept mapping stores the relationships between the respective concept and other concepts, as well as indicates the unique id of each concept. For example, information about how the concept *Student* is related to other concepts such as *Discussion Forum*, *Quiz*, etc. is stored (e.g., to find out which student has posted messages in a discussion forum). Fourth, a

concept mapping stores labels for the concept and its attributes. Such labels are names for the concept and its attributes that can be easily understood by users of the tool without knowledge about the database schema and how tables and columns are called in the database. These labels are then used throughout the system as names for concepts and their attributes (e.g., in the pattern creation process) so that users are able to employ terminology they are comfortable with. These labels are LMS/template specific, so that different terminology can be used for different learning systems. This is an important feature to make the system user-friendly, as different vendors use their own terminology, to which users of that LMS become accustomed.

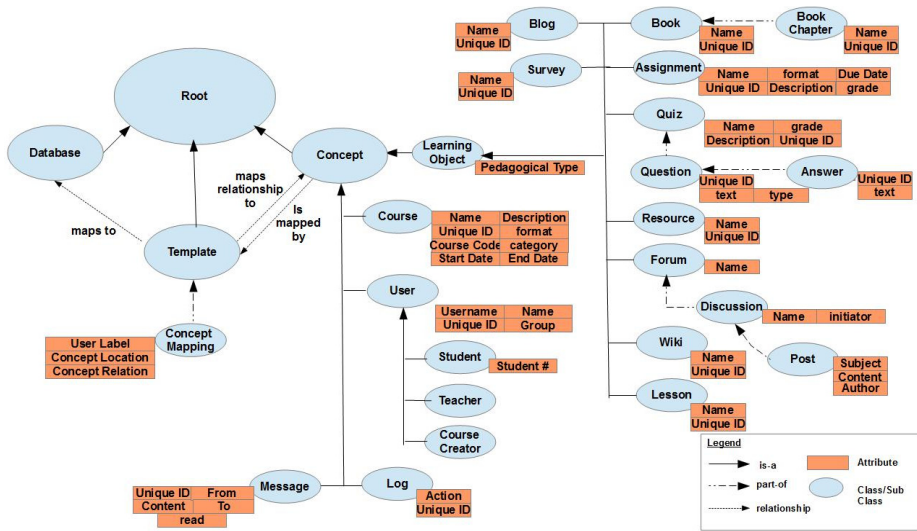


Fig. 1. AAT Ontology

3.2 Pattern Chaining

Pattern chaining facilitates the creation of complex patterns through utilizing simpler patterns as inputs. This way, complex patterns can be easily created. A user can start with a simple pattern (e.g., a list of students' grades on assignments) and then refine the pattern more and more, deepening his/her investigation about learners' behaviours/actions or the quality of learning material. This functionality allows a user to incrementally and progressively build increasingly complex patterns. For example, if a user wishes to identify difficult quiz questions, he/she can build a pattern that extracts data about the average performance of students on questions within quizzes. On top of this pattern, the user can create another pattern that outputs all quiz questions where the average performance of students is lower than, for example, 70%.

There are two types of pattern chaining. First, one pattern can be used as input for another pattern, and therefore restrict the result set of the base pattern. Second, two existing patterns can be merged, leading to a combination of the result sets of these two patterns.

In order to make it possible to chain patterns (for both types), particular additional data as well as meta-data need to be stored for each pattern. Such additional data include identifies of the table(s) from which the data have been retrieved. Such identifiers are indicated in the ontology as the unique id of a concept. For example, if we retrieve a list of all assignment grades that have been achieved by students, then additional data such as the assignment id has to be stored (even if the user is not interested in this id). Furthermore, meta-data are needed that indicate from which location the respective data have been retrieved. This information can be easily retrieved from the AAT ontology, in particular the concept mappings.

3.3 Advanced and User-Friendly Pattern Creation Interface

In this section, an easy to use wizard-style interface for creating patterns is presented. From technical point of view, the end result of a pattern is a SQL statement that is used to retrieve the data the user specified. This SQL statement is built from the template as it is the result of joins defined in a template's concept mappings. Patterns can be reused across different LMS instances.

The interface of the pattern creation wizard is mainly based on questions that users can answer in order to "tell" the system what data they want to access and how they want to analyse them. As a first step, users can decide if they want to: (1) create a new pattern from scratch, (2) use a pattern as input for building another pattern, (3) merge two patterns, or (4) perform an analysis on an existing pattern.

To create a new pattern from scratch, five steps are needed. First, the user can *select the concepts* he/she wishes to investigate. These concepts are (as per the AAT Ontology) linked to the concept mappings for a particular LMS' template. If an LMS does not support that concept (i.e., there is no concept mapping for it) then that concept is not available for selection. Second, the user can *select the concept attributes* he/she wishes to investigate (Fig. 2). These attributes are presented based on the selected concepts. In both the first and second steps, the text (i.e., names of concepts and attributes) comes from the ontological labels for each concept and attribute rather than directly from the database and therefore, is easy to understand for users without knowledge about the database structure. Third, the user can define filters for each of the attributes. The filter type is based on the attribute's data type (as per the ontology). Thus, a user can filter a string attribute by exact match or by a sub-string; an integer can be filtered by an exact match to a number or by a lower and/or upper bound (or both). The user also specifies if he/she wishes to apply all the filters (logical AND) to the pattern or not (logical OR). Fourth, the user can define the sort order of the result set and how each attribute will be sorted (ascending or descending). Fifth, the user can save the pattern, with the option to share the pattern with other users.

In each step, a Pattern Result pane and SQL pane is shown (as in Fig. 2), presenting the user with the tabular result of the query (limited to the top 10 rows) and the exact SQL statement that will be run against the source database to obtain the result set. The SQL pane is more for advanced users/administrators and can be hidden via the application configuration file.

Active Database: OldMoodle
 Selected Datasets: COMP200, COMP301
 Choose Concept > **Choose Attributes** > Add Limits > Define Sorting > Save

What concept attributes you are interested in?

Quiz Quiz Name
 Quiz Grade

Course Course Name

<< Choose Concepts Add Filters >>

Pattern Result (top 10 rows only)

Quiz Name	Quiz Grade	Course Name
Quiz 1	45	COMP200
Quiz 1	50	COMP301
Quiz 2	75	COMP200
Quiz 2	100	COMP200

SQL Pane [Copy](#)

```
SELECT mdl_quiz.name, mdl_quiz.grade, mdl_course.fullname
FROM mdl_quiz
JOIN mdl_course on mdl_quiz.course = mdl_course.id
WHERE mdl_course.id in (47,39)
```

Fig. 2. The Pattern Creation Interface: Selecting Concept Attributes

In order to use a pattern as input or to merge two patterns, the user first selects the base pattern or the two patterns to be merged. After that, the same wizard as for creating a new pattern is used to go through the five steps of selecting concepts, concept attributes, filters, and the sort order, as well as storing the newly created pattern.

To perform an analysis (or calculation) on an existing pattern, the user can select the base pattern, the type of analysis (i.e., counting, calculating the sum or average, and presenting the minimum or maximum) and the concept attributes on which the respective analysis should be performed. Such analyses can either be performed for one attribute, resulting in a single value (e.g., the number of forum postings in a course), or for one attribute per concept, resulting in an additional column of the result set of the base pattern (e.g., the average number of postings per student). This new column can then be named by the user.

4 Conclusions and Future Work

AAT is an innovative tool that enriches smart learning environments by reducing the complexity to access educational log data and therefore enabling educators and course designers to perform simple and complex analytical queries on students' behaviour across courses and programs. In this paper, a user-friendly and powerful mechanism is introduced that extends the AAT tool to be more intuitive but at the same time, advances the functionalities to retrieve complex information from a learning system's log data. In order to do so, an extensible ontology has been built, a refined approach for pattern chaining has been developed and a wizard-style user interface with further options for pattern creation has been implemented.

By extending the functionality and user-friendliness of AAT, the tool becomes more relevant for its target group, namely course designers and educators. On one hand, this user group typically does not have background in computer science or knowledge about the database schema of a learning system. Therefore, advanced user-friendliness in form of a quick and easy way to access data is required. On the other hand, users want to perform comprehensive analyses on the log data in order to get meaningful insights into how students learn, how well teaching strategies work, how much a particular course design supports students, etc. While increased functionality and flexibility often comes at the cost of usability, the proposed mechanism addresses both of these needs and provides users with a user-friendly and powerful way of accessing and analysing educational log data. By offering such access and possibilities for analysing data, the tool provides a smart environment that facilitates course designers' learning about the effectiveness of their courses as well as educators' learning about student behaviours that impact on learning outcomes.

Future work will deal with advanced visualization of data, adding statistical functionality (e.g., regression, correlation), and conducting a study where learning designers and educators evaluate AAT with respect to its usability and usefulness. Another direction of future work will be to investigate the potential of AAT to increase learners' awareness of their learning processes through providing them with access to their behaviour and performance data.

References

1. Aljohani, N.R., Davis, H.C.: Significance of Learning Analytics in Enhancing the Mobile and Pervasive Learning Environments. In: International Conference on Next Generation Mobile Applications, Services and Technologies (NGMAST), pp. 70–74 (2012)
2. Ross, S.M., Anand, P.G.: A computer-based strategy for personalizing verbal problems in teaching mathematics. *Educational Tech. Research & Development* 35(3), 151–162 (1987)
3. Siemens, G.: Learning analytics: envisioning a research discipline and a domain of practice. In: International Conference on Learning Analytics and Knowledge (LAK), pp. 4–8. ACM, New York (2012)
4. Leony, D., Pardo, A., Valentín, L., Castro, D., Kloos, C.D.: GLASS: a learning analytics visualization tool. In: International Conference on Learning Analytics and Knowledge (LAK), pp. 162–163. ACM, New York (2012)
5. Liaqat, A., Hatala, M., Gasevic, D., Jovanovic, J.: A Qualitative Evaluation of Evolution of a Learning Analytics Tool. *Computers & Education* 58(1), 470–489 (2012)
6. Retalis, S., Papasalouros, A., Psaromiligkos, Y., Siscos, S., Kargidis, T.: Towards Networked Learning Analytics – A concept and a tool. In: International Conference on Networked Learning, Lancaster (2006)
7. Graf, S., Ives, C., Rahman, N., Ferri, A.: AAT: a Tool for Accessing and Analysing Students' Behaviour Data in Learning Systems. In: International Conference on Learning Analytics and Knowledge (LAK), pp. 174–179. ACM, New York (2011)
8. Fowler, M.: Patterns of enterprise application architecture. Addison-Wesley, Boston (2002)
9. PHP Manual–Database Extensions,
<http://www.php.net/manual/en/refs.database.php>
10. CAS, <http://www.jasig.org/cas>
11. Chandrasekaran, B., Josephson, J.R., Benjamins, V.R.: What are ontologies, and why do we need them? *Intelligent Systems and their Applications* 14(1), 20–26 (1999)