This document is published in:

José A. Ruipérez-Valiente, Pedro J. Muñoz-Merino, Carlos Delgado Kloos. An architecture for extending the learning analytics support in the Khan Academy framework. ***Proceedings of the First International Conference on Technological Ecosystem for Enhancing Multiculturality,*** **277-284, Salamanca, Spain. ACM, 2013.**

DOI: 10.1145/2536536.2536578

An architecture for extending the learning analytics support in the Khan Academy framework

José A. Ruipérez-Valiente
Universidad Carlos III de Madrid
Avenida Universidad 30
28911 Leganés (Madrid) Spain
jruipere@it.uc3m.es

Pedro J. Muñoz-Merino
Universidad Carlos III de Madrid
Avenida Universidad 30
28911 Leganés (Madrid) Spain
pedmume@it.uc3m.es

Carlos Delgado Kloos
Universidad Carlos III de Madrid
Avenida Universidad 30
28911 Leganés (Madrid) Spain
cdk@it.uc3m.es

ABSTRACT

The Khan Academy platform enables powerful on-line courses in which students can watch videos, solve exercises or earn badges. This platform provides an advanced learning analytics module with useful visualizations for teachers and students. Nevertheless, this learning analytics support can be improved with recommendations and new useful higher level visualizations in order to try to improve the learning process. In this paper, we describe our architecture for processing data from the Khan Academy platform in order to show new higher level learning visualizations and recommendations. The different involved elements of the architecture are presented and the different decisions are justified. In addition, we explain some initial examples of new useful visualizations and recommendations for teachers and students as part of our extension of the learning analytics module for the Khan Academy platform. These examples use data from an undergraduate Physics course developed at Universidad Carlos III de Madrid with more than 100 students using the Khan Academy system.

Categories and Subject Descriptors

K.3.1 [**Computer Uses in Education**]: *Distance learning;* H.1.2 [**User/Machine Systems**] *Human information processing;* D.2.11 [**Software Architectures**] *Domain-specific architectures*;

General Terms

Design, Performance, Algorithms, Experimentation, Human Factors

Keywords

Learning analytics, architectures, recommenders, visualizations, data processing

1. INTRODUCTION

Education is being boosted by new tendencies to improve the learning process and learning analytics is one of the most promising tools. Some examples of applications of learning analytics are the ability to early detect failures, generate feedback, or report on the learning process so that teachers can act to help students succeed [2].

On the other hand, the use of MOOCs (Massive Online Open Courses) is emerging as a new paradigm. In this context, the use of learning analytics becomes more necessary because it is required to analyze and interpret the learning process of thousands of students in a course. Automatic learning analytics tools that give insights about this learning process are required because a teacher cannot take care of so many students in an efficient way without technological help.

The Khan Academy[1] platform is one of the pioneer systems for running MOOCs. This platform is one of the more complete at the moment taking into account the provided functionality. The Khan Academy system offers a great learning analytics support by default. Some of the provided features are related to the skill progress, the exercise report, or the student activity report. Although there is a debate about the definition of learning analytics, and different researchers have different points of view, we consider the learning analytics term in a broad sense as introduced in the CFP of the 1st LAK conference[2]: "Learning analytics is the measurement, collection, analysis and reporting of data about learners and their contexts, for purposes of understanding and optimizing learning and the environments in which it occurs". According to this definition, the Khan Academy platform provides different learning analytic features by default.

Even though the Khan Academy system offers this great learning analytics support, there is interesting information which is not included in its module. Therefore, an extension is required to achieve this goal. In previous work [8], we presented some new interesting parameters which can be included as part of the learning analytics support for the Khan Academy platform. Moreover, ways and specific formulas to infer this higher level information from low level data were presented [8].

In this work, we present our implemented architecture for our Add-on of the Learning Analytics Support in Khan Academy (ALAS-KA). This architecture enables to process the huge amount of educational low level data (in the form of events) and obtain higher level learning information which can be presented in the form of visualizations and recommendations.

The remainder of this paper is organized as follows. In Section 2, related works are presented. Section 3 gives an overview of the

Conference'10, Month 1–2, 2010, City, State, Country.

[1] http://www.khanacademy.org

[2] https://tekri.athabascau.ca/analytics/

learning analytic support provided by the Khan Academy platform by default. Section 4 explains the implemented architecture, describing its different elements and their relationships, as well as the data model and the data processing design. Section 5 exposes some specific examples of implemented visualizations and recommendations that are working in this architecture. Finally, Section 6 is devoted to the conclusions.

2. RELATED WORK

The application of learning analytics techniques is increasing. These techniques are very valuable to try to understand the learning process. In this direction, several systems incorporate visualization tools like CAMERA [12] which enables social network visualizations, GLASS [6] which shows the most used events by students, CourseVis [7] which is integrated in the on-line WebCT platform, or other works applied in Moodle such as Moodog [15] that tries to make an analysis of some interactions in a course through visualizations, or the application of visual analytics techniques to logs using tag clouds [5]. The work in [4] shows several learning dashboards and recommender examples. Furthermore, a comparison between educational and non-educational user tracking environments is done [4]. It is important to find the most efficient way to transfer the knowledge to the user.

As in our ALAS-KA proposal, several works analyzed learning analytics applications covering different layers such as in article [11]: with data, processing, and visualization dashboards. In our work, we present an architecture and solution for the specificities of the Khan Academy framework.

There are also many educational recommender systems. In this direction, there are tools that assist the learning process with forum posts that might result helpful for students, or messages which might result of benefit from former learner's knowledge [3]. There are also other several recommendation tools that help people by offering resources and papers that could be of their interest (e.g. [10] [14]).

The Khan Academy system includes an educational framework which incorporates innovative concepts like gamification or learning analytics techniques. The framework can be used for applying methodologies such as 'flipping the classroom'. The main focus of the platform is on exercises and videos, which are grouped by different topics forming a knowledge tree. One of the main features of the system is related to gamification techniques (e.g. a student can earn badges or points by watching videos or solving exercises correctly).

Although the Khan Academy platform already provides a powerful learning analytics functionality, there is path to improvement. The Khan Academy system has different visualizations for knowing the students' progress in different skills or the students' last activity in the different resources. Despite these visualizations are very useful for students and professors, there are still a lot of new possibilities and interesting information that can be inferred from the educational data in the Khan Academy platform. Another important issue is how to make the information understandable. Sometimes, the visualizations can be overwhelming for the users and the conclusions might not be clear. Moreover, sometimes it is hard to know how to intervene and understand these visualizations. Therefore, a recommender is also a useful tool which is not included in Khan Academy. Thus, in our work in ALAS-KA, we have designed an add-on that expands the learning analytics support with new visualizations and recommendations, enabling professors and students with more possibilities.

3. OVERVIEW OF THE LEARNING ANALYTICS SUPPORT IN THE KHAN ACADEMY PLATFORM

The Khan Academy platform has a powerful learning analytics module. Khan Academy was one of the pioneer platforms to implement a great learning analytics support. The system allows accessing to a big amount of educational data. The Khan Academy system stores most of the low level events and users' interaction during their learning path. In this way, all the required data is available for processing.

The learning analytics module has individual visualizations so that students can access their own information. There are also some global class visualizations which can only be accessed by teachers.

One example of individual visualization is called *Activity,* which allows users to access their activities organized by time. With this feature, they are able to see what resources they have been used each day for different intervals of time. Another type of individual visualization is shown in figure 1. This visualization allows students to know their time distribution for the different skills or videos (shown as percentage). In addition, figure 1 distinguishes between time solving exercises (outer circle) and time watching videos (inner circle). Moreover, users can access the option *Skill Progress* to check their progress status in the different skills. In addition, the option *Progress Over Time* enables students to check their progress during the time that the course took place.

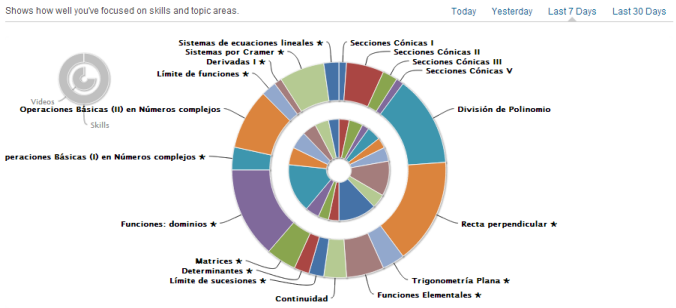

Figure 1. Individual visualization of activity focus in Khan Academy

The Khan Academy platform provides useful visualizations for teachers. In the Khan Academy system, there is the role of a coach which makes the function of a teacher. Any user can select other users as coaches. When this happens, the coaches (usually teachers) will be able to monitor the learning information of these students that selected this user as a coach. A coach is able to access class visualizations for progress or skills.

Coaches can do a close tracking of each student individually in the *Daily Activity Report*. This way they can check the individual activities done in that day for the whole class. In addition, the coach can access a much more in depth *Activity Report* for each student. A screenshot of this detailed report is shown in figure 2. The presented information includes all the exercises a student has attempted, with the number of attempts, the time invested or if the user asked for hints.

Figure 2. Detailed exercise report in Khan Academy.

Figure 3 shows two screenshots of another type of visualization: one with visualizations of each student, and another with class visualizations of the entire class as a whole. The screenshot on the left side of figure 3, which is called *Progress Report*, has as rows the different students and as columns the different exercises in the course. For each pair user/exercise, the color of the matrix indicates if the student started the task, obtained proficiency or struggled. This is helpful to know an overall class progress. The screenshot on the right side of figure 3, which is called *Progress Summary*, shows per each exercise and in this order: the number of students who struggled in that exercise, who did not do the task, who did the task correctly at least once, and who obtained the proficiency in this exercise. This allows keeping track of the class progress in each exercise separately, and also helps to detect problems in certain skills.

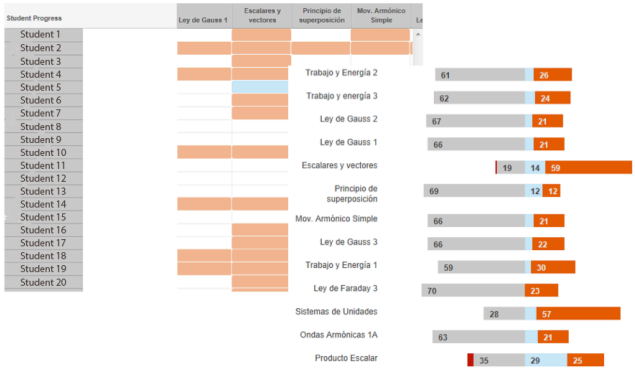

Figure 3. Skill progress of the class in Khan Academy.

Several of the learning analytics visualizations in the Khan Academy system are the representation of events which are stored in the Datastore but without further processing. Although they are very useful for self-awareness and class tracking, much more information can be inferred and processed. Furthermore, once the data is processed, a recommender system can be implemented and this is not available in the Khan Academy system.

4. DESCRIPTION OF THE ARCHITECTURE

This section presents our architecture to extend the learning analytics module for the Khan Academy platform. Two types of applications can be thought for the extension:

- Standalone applications that can only work for a specific software. They are usually designed for a certain purpose like e.g. teaching the multiplication table [13]. They are usually small and limited applications.

- Applications designed as a plug-in for other platforms. They usually retrieve data from the platform database and the computations are done as independent processes. This way, minimum changes in the platform are required. Some examples of this type are [7] for the WebCT or [15] for Moodle.

Our work is based on a plug-in development for the Khan Academy platform. We have implemented a system that works with the data generated by the users in the Khan Academy platform. The information is extracted from it.

4.1 SYSTEM ARCHITECTURE

Our solution runs over a Google App Engine (GAE) server using the python programming language. Figure 4 represents this architecture with its related elements.

Our implemented ALAS-KA module is a plug-in for the Khan Academy platform. ALAS-KA allows new visualizations and recommendations. The different elements of the architecture are the following:

- Students: They interact with the Khan Academy platform by watching videos, solving exercises and using other elements. They can also access their own individual visualizations and recommendations of ALAS-KA but not the information of other students or the global class. In this way, self-reflection and help is enabled for students.

- Professors: One of their assigned tasks is usually to design the videos and exercises for the Khan Academy courses. Moreover, they usually have to supervise the work of the students. ALAS-KA enables teachers to access visualizations of every student individually but also of the entire class. The ALAS-KA plug-in allows professors to control the progress and habits of their students much easier, because it includes new visualizations and intelligent information. Teachers can make interventions for students with problems with the purpose of preventing dropouts, making early warnings, detecting exercises or videos with problems and making recommendations in an easier way.

- Datastore: The Google App Engine Datastore provides storage for the Khan Academy platform data. Most of the events of the user interactions during their learning path are stored as a *Model Class*, which is a kind of entity that includes the types and properties. Since we have designed our add-on to run in the same GAE server as the Khan Academy instance, because of its initial simplicity, we also use this Datastore for data persistence. Therefore this GAE Datastore also contains the ALAS-KA models. Other options might be used and a different database system outside GAE server might be selected. There is a key difference between the Khan Academy models and the ALAS-KA ones. The Khan Academy models are accessed to extract the student's data required to make the processing, while the ALAS-KA models are used to store and retrieve the results of the information, as a result of the data processing. Further explanation about this aspect will be given in section 4.2.

- Data processing: This module is in charge of making the proper computation to transform from different low level data from the Khan Academy models into higher level information that is stored as ALAS-KA models. The processing task is performed by the same GAE server where the Khan Academy platform runs. However, this processing might be done outside the Khan Academy server as an external service, which would not be tied to the GAE server. Due to its importance, the data processing will be thoroughly explained in subsection 4.3.

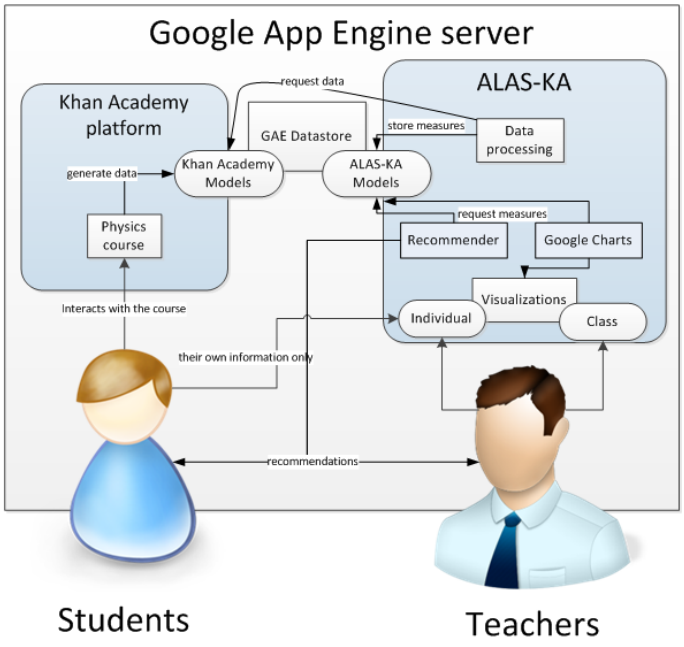

Figure 4. Interaction diagram of the system architecture.

- Visualizations: The Google Charts API [3] was selected for the visualizations because of its simplicity and variety of charts. This API only needs to load their javascript libraries to be ready for use. It is widely spread and tested and there is a lot of documentation which make easy the use of this API. Another important matter is that charts are rendered using HTML5/SVG. Therefore, this technology provides cross-browser and cross-platform compatibility (e.g. with tablets or smartphones). Google Charts only renders the graphics in the client side and does not make any processing. In our case, the data needed to build the visualizations are requested to the ALAS-KA models in the Datastore. Therefore, this required data could also be received from an external source such as a web service.

- Recommender: Once the processing of the data is done, the information is available in the form of ALAS-KA models through the Datastore, which are ready to be accessed. The Khan Academy system already includes a recommender of exercises and videos. But there are not recommendations related to the learning process. The function of the recommender is to analyze the results and send warnings to students or professors based on some rules. These recommendations can be based on conditions and configurations designed by experts. Recommendations can range from telling a student that it is better to work in the mornings because the student's efficiency is higher during this time, or communicating a professor that some students have not engaged with the platform recently to prevent their dropout.

4.2 DATA MODEL

Our solution permits to offer information of different courses in the same Khan Academy instance. This is a difference with respect to the initial learning analytics support given by the Khan Academy platform. In case there are exercises and videos from different courses (e.g. Mathematics, Chemistry and Physics) in the same Khan Academy instance, and there are different students in each course, ALAS-KA allows managing the information of each course and visualize it separately. This feature improves the accuracy of the information since there is not a mixing among courses.

In addition, since the Khan Academy and ALAS-KA data models are different and modifications in the Khan Academy platform were not made, then there are not further problems in installing the ALAS-KA add-on. We are working with a fixed version of Khan Academy, so the Khan Academy models do not change. Anyway, the development can be made totally independent of the Khan Academy system, because ALAS-KA only needs of Khan Academy models to work. In this way, an ETL (Extract, Transform and Load) process can be configured to extract the data needed from the Khan Academy Datastore, transform it in order to be usable and finally load it into ALAS-KA Datastore. Using this configuration, the Khan Academy platform and ALAS-KA could be running in different servers without any problems. Furthermore, the *Transform* step could be adaptable as a *middleware* layer in case there are changes in the Khan Academy models.

Figure 5 shows the data model and the relationships between entities. The Datastore models of the Khan Academy platform are represented in yellow color, while the new added ALAS-KA models are represented in blue color. Figure 5 only represents the Khan Academy models that are more important in our add-on. Next, we make a brief review of the different models of the system:

- *UserData*: This model represents a user in the Khan Academy. It includes a variety of information such as the email, nickname, profile avatar, points and badges, among many others.

- *Exercise* and *Video*: These models represent the exercises and videos of the Khan Academy platform. They store information about titles, descriptions and other data required for the Khan Academy platform.

- *UserExercise* and *UserVideo*: These models represent the interaction in global terms of a user with a certain exercise or

[3] https://developers.google.com/chart/?hl=en

video. For example, *UserExercise* includes the number of times a user has accessed an exercise or the total progress in that exercise. *UserVideo* has properties to know if the video has been completed or the time the user spent in that video.

- *ProblemLog* and *VideoLog*: These models store the data of each time a user accesses an exercise (*ProblemLog*) or a video (*VideoLog*). The storage of all these events (timestamps, attempts, hints asked and more variables) about the interaction flow solving an exercise (*ProblemLog*) makes possible the processing of most of our measures. Similarly, the number of seconds invested by a user each time the user accesses a video, the timestamp with the exact access time and other events (*VideoLog*) also allow to detect other behaviors, such as for example if a user tried to solve an exercise before watching the related video first.

- *AlaskaUser*: This is the user entity in ALAS-KA. One of its properties is related to the same user as in the *UserData* entity in the Khan Academy data model. We also distinguish between professors and students for the different courses. In this way, we accomplish to configure the different roles and permissions in the add-on. Professors are not considered as students in the computations, and their data is not used for the processing phase.

- *AlaskaExercise* and *AlaskaVideo*: These models represent exercises and videos in ALAS-KA. Sometimes, there are introductory or non-mandatory exercises and videos, which are not required for completing a course. Therefore, these entities only store the exercises and videos that are going to be processed, and the course they belong to.

- *UserProfile*: This is the main entity in ALAS-KA. It stores all the metrics that have been processed for a user. This entity is updated each time that a cron job is triggered with the new values. The same user could be in different courses, which is also supported.

- *ClassMeanProfile*: This model is used to check if a student knowledge level is better or worse than the average of the class. As an example, visualizations that compare a student progress with the mean value of the class might be presented. The entity *ClassMeanProfile* is used to store the mean value of the class. In this way, it is not required to calculate this mean each time that a visualization is requested.

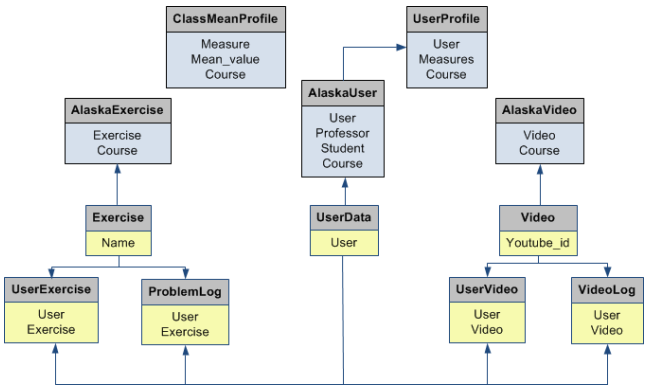

Figure 5. Overview of the data model of the complete system. In yellow KA models and in blue ALAS-KA models.

Our ALAS-KA system works in parallel to the Khan Academy platform without disturbing it but adding new functionality and learning analytics metrics. ALAS-KA allows different courses in the same Khan Academy instance and different functionality for professors and students.

4.3 DATA PROCESSING DESIGN

One of the challenges in learning analytics is the extraction of enormous amount of data from databases and the further processing that is usually required. It is challenging how to make a design that can efficiently work without collapsing the database or the server processing capacity. Moreover, it is a challenge the need of making the extraction and processing of the data in a regular interval.

Each database system is different, so the most adequate architecture might differ a little from others. For example, GAE Datastore has several restrictions. One example is the need to use pre-indexes to run queries that make it more restrictive. Another restriction is that user queries can run only for 60 seconds. However, on the positive side, GAE Datastore is designed to scale very well and allows applications to maintain a high performance, being transparent for the developer. Figure 6 shows the data processing design. A justification of the different decisions is explained next:

- Cron jobs: The heaviest queries and processing could last from 30 to 60 seconds. This fact makes unaffordable to provide real time requests because the user experience would be so bad. In order to solve this problem, we use the App Engine Cron Service to configure the system in order to process the metrics of each user in the platform at regular intervals. Once the processing is done, the results are stored as presented in the previous subsection (within the *UserProfile* ALAS-KA model). Thus, when a visualization is requested by a user, the system only needs to get the value of the metrics that were previously processed and stored. It is also important to find an appropriate tradeoff between the computational load and the time interval chosen. This is also because the computational and Datastore operations have a billing cost in GAE. Our decision has been to make the processing 4 times per day (every 6 hours). We have created 5 different cron jobs, one for each group of metrics that we have designed [8]. As we divide functionality in parts, we make it more scalable and a more balanced processing. These jobs are automatically triggered and they invoke a URL using a HTTP GET request. This code that is invoked is in charge to generate a *Task* entity for each student in the platform and add it to its corresponding *Queue* as explained in the next list item.

- Queue: Their function is to execute background work in the application outside a user request. The Task Queue API system internally handles whether the execution of tasks is possible. As the system makes it automatically, then this process is totally transparent for the developer, which makes very easy to use this service. There are several configuration parameters like the number of task/second that can be executed. Five different *Queue* entities have been configured: one for each different cron job of ALAS-KA. This also helps to divide the background work in smaller *Queue* models and by functionality.

- Tasks: A *Task* entity is a small and discrete unit of work in GAE. Each *Task* entity (which is a processing unit of a

student metrics) is added into its *Queue*. They are executed in the same order that were added. The App Engine system handles the *Queue* system executing *Task* entities in background whenever is possible. Thus, this feature does not cause a server overload. Each *Task* entity gets the required data from the Khan Academy Datastore models, and makes the custom processing of the proposed measures. Once it is finished, the results are stored in the *UserProfile* model. This method allows the system to retrieve the data quickly when users are watching visualizations.

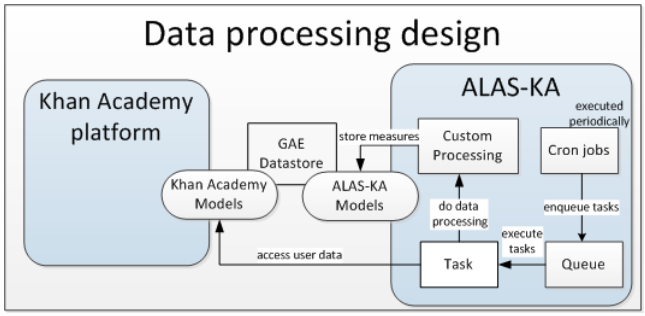

Figure 6. Diagram for the data processing design.

The design allows dividing the extraction and processing flow into small Task entities that are executed by the system gradually without overloading the server. Another addressed issue is the answer time for a request. If a heavy processing metric comes, the computation could take more than 10 seconds. Obviously, this is absolutely unacceptable for the user's experience. With the proposed solution, the user response time is very short because there is no need to make the processing of the metric, but only retrieving the stored value. This solution avoids the request timeout problem (10 minutes maximum in GAE), but this issue would arise if we had not divided the work into small *Task* entities.

This architecture could be applicable for other educational systems that run over GAE if the processing metrics would be adapted to the platform.

5. VISUALIZATION AND RECOMMENDATION EXAMPLES

This section presents some innovative examples of visualizations and recommendations that are implemented in our architecture. These examples are extensions of the learning analytics module of the Khan Academy platform.

In August 2012, an instance of the Khan Academy platform was used at Universidad Carlos III de Madrid to offer an undergraduate Physics course. The goal of this course was to review concepts that students required for their first year in an engineering degree. The entire course comprised a set of 27 videos and 35 exercises which were developed by professors from Universidad Carlos III de Madrid. These contents were uploaded to the own Khan Academy server instance at Universidad Carlos III de Madrid. The following visualization and recommendation examples use educational data generated in that course.

5.1 PERSISTENCE MODEL

The first example gives insight about the students' persistence when watching videos (also easily extensible on doing exercises). Persistence shows the capacity of focusing on a goal and accomplishing objectives. We measure the student's video persistence by comparing the number of the videos that a student started and finished.

Figure 7 presents a visualization that shows the students' video persistence. In the X-axis, the different users are represented. In the Y-axis, the number of videos for each user is provided. A green circle represents the number of videos that have been accessed by a user. A blue circle represents the number of these videos that have been finished by that user. These two points can be compared to infer a persistence profile on watching videos. Larger lines represent users that started lots of videos but did not end many of them. While shorter lines represent users who are more persistent on finishing videos. An extreme case is for example user number 30, who started to watch all videos but did not end any of them. On the contrary, we can find users with only one green point, which means that the student ended all the videos that he started. Examples of this type of users are number 2 to 6 from figure 7.

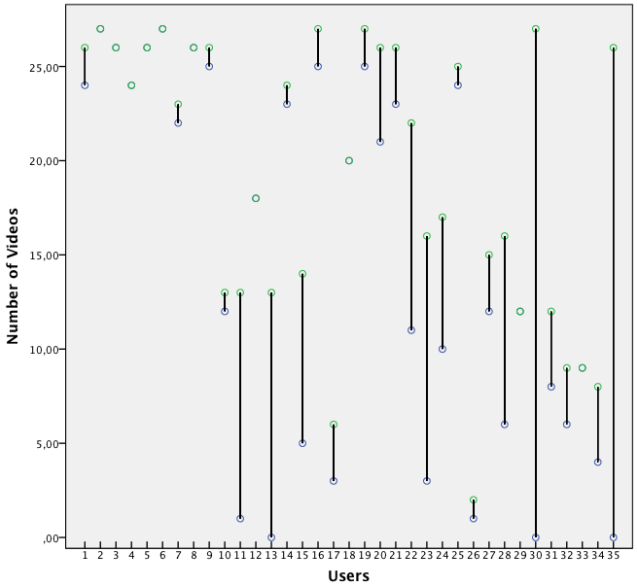

Figure 7. Persistence model in videos. Started videos (green) and finished videos (blue).

As an example of an easy condition to configure the recommender module for non-persistent users, a rule might be to check the ratio between the number of videos started and finished. If this ratio is lower than a chosen threshold (e.g. 0.4), the system will send a warning. A possible recommendation might be to advise students to focus in a resource before attempting others. In addition, warnings to professors might be sent advising about students' behavior.

An important issue is that the same visualization information can have different interpretations. For example, a student can start a lot of videos but the student might not end most of them. This might be because the student is struggling or because the student already knows the concepts explained in the videos. The final interpretation and recommendation should depend on other variables such as if the student already knows how to solve the related exercises. In some cases, it is not possible to determine the exact causes of some information visualizations.

5.2 RESOURCE FOCUS

Students can focus more on watching videos or solving exercises. The parameters of time invested by users on solving exercises and watching videos are calculated separately in order to know their resource focus. Next, a comparison is done between both parameters. This ratio might give a clear indicator of students'

preferences. As an example, students who devote greater time on solving exercises might imply more active learners. But the final interpretation and recommendation should depend on the educational context.

Figure 8 presents the information of learners' time distribution for videos and exercises. Blue bars represent the time devoted by users on solving exercises. The green bars represent the time devoted by users on watching videos. An example of different behaviors can be observed in users 16 and 17. The first one has spent much more time watching videos while the second has focused the learning on resolving exercises.

Regarding recommendations, communications might be sent to teachers for advising them about students who are focused on videos or exercises. In this way, a teacher can make an analysis and act properly depending on the specific educational context.

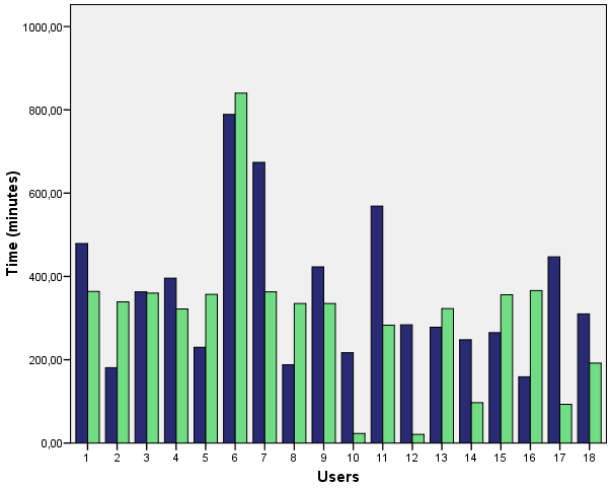

Figure 8. Total time distribution for each user: exercise time (blue) and video time (green)

6. CONCLUSIONS

In this paper, we propose an architecture to extend the learning analytics support for the Khan Academy platform. The interactions among the different elements are explained, as well as our data processing design. We have also presented some initial examples of recommendations and visualizations that are working in this architecture. These examples are illustrated with data of a real course in Physics at Universidad Carlos III de Madrid.

The presented architecture enables to make the processing of low level data and transform it into higher level learning information. The architecture can work independently from the Khan Academy platform. This implies that the processing modules can receive the data extracted from the database and make the calculations externally, invoking the proper web services. This solution enables that some of the metrics might be reused by other platforms. In the future, we aim to integrate our solution into more generic architectures for e-learning such as OKI [9] or Knowledge Tree [1].

Nevertheless, in order to reuse some of the metrics, the semantics of each platform should be taken into account. While there are some parameters that are very specific for the Khan Academy system (e.g. for the process of badges calculation), however there are others that might be common to other platforms (e.g. some related to videos since video watching is very similar in other systems). Moreover, several of parameters about exercises might be reused e.g. in Google Course Builder[4], which has recently incorporated a player for Khan Academy exercises.

7. ACKNOWLEDGEMENTS

Work partially funded by the EEE project, "Plan Nacional de I+D+I TIN2011-28308-C03-01" and the "Emadrid: Investigación y desarrollo de tecnologías para el e-learning en la Comunidad de Madrid" project (S2009/TIC-1650)".

We would like to thank the professors in Universidad Carlos III de Madrid who developed the physics materials for the course (videos and exercises).

The last author wishes to acknowledge support from Fundación CajaMadrid to visit Harvard University and MIT in the academic year 2012/13

8. REFERENCES

[1] Brusilovsky, P. "KnowledgeTree: A distributed architecture for adaptive e-learning". *In Proceedings of the 13th international World Wide Web conference*, 104-110, 2004.

[2] Campbell, J. P. Oblinger, D. G. Academic Analytics. *EDUCAUSE White Paper, October 2007.*

[3] Chen, W. and Persen, R. A recommender system for collaborative knowledge. *Proceedings of the 2009 conference on Artificial Intelligence in Education*, 309-316.

[4] Duval, E. 2010. Attention Please! Learning Analytics for Visualization and Recommendation. *Proceedings of LAK11: 1st International Conference on Learning Analytics and Knowledge 2011*.

[5] García-Peñalvo, F. J. Conde, M. Á. Bravo, S. Gómez, D. A. and Therón, R. Visual Analysis of a Moodle-based Object Oriented Programming Course. *International Journal of Computers Applications. Proceedings on Design and Evaluation of Digital Content for Education (DEDCE),* 2011, 8-14.

[6] Leony, D. Pardo, A. de la Fuente, L. Sanchez, D. Delgado, C. GLASS: A Learning Analytics Visualization Tool. *LAK '12 Proceedings of the 2nd International Conference on Learning Analytics and Knowledge,* 162-163.

[7] Mazza, R. Dimitrova, V. Visualising Student Tracking Data to Support Instructors in Web-Based Distance Education. *Proceedings of the 13th international World Wide Web conference on Alternate track papers & posters*, 154-161.

[8] Muñoz-Merino, P. J. Ruipérez, J. A. Delgado, C. Inferring higher level learning information from low level data for the Khan Academy platform. *Proceeding LAK '13 Proceedings of the Third International Conference on Learning Analytics and Knowledge*, 112-116.

[9] OKI (Open Knowledge Initiative), http://sourceforge.net/projects/okiproject/?_test=beta

[10] Recker, M. M. Walker, A. Lawless, K. What do you recommend? Implementation and analyses of collaborative information filtering of web resources for education. *Instructional Science 31*: 299–316, 2003.

[11] Santos, J. L. Verbert, K. Govaerts, S. Duval, E. Visualizing PLE Usage. *Proceedings of EFEPLE11 1st Workshop on*

[4] https://code.google.com/p/course-builder/

Exploring the Fitness and Evolvability of Personal Learning Environments, 34-38.

[12] Schmitz, H. Schefel, M. Friedrich, M. Jahn, M. Niemann, K. Wolpers, M. CAMera for PLE. Learning in the Synergy of Multiple Disciplines. *Proceedings of the 4th European Conference on Technology Enhanced Learning: Learning in the Synergy of Multiple Disciplines*, 507-520.

[13] Schön, M. Ebner, M. Kothmeier, G. 2012. It's Just About Learning the Multiplication Table. *Proceedings of the 2nd International Conference on Learning Analytics and Knowledge*, 73-81.

[14] Ya Tang, T. McCalla, G. Smart Recommendation for an Evolving E-Learning System: Architecture and Experiment. International Jl. on E-Learning (2005) 4(1), 105-129.

[15] Zhang, H. Almeroth, K. Knight, A. Bulger, M. Mayer, R. Moodog: Tracking Students' Online Learning Activities. *Journal of Interactive Learning Research*, 21(3), 407-429.