This document is published in:

Proceedings of the: 12th IEEE International Conference on Advanced Learning Technologies ICALT 2012, Rome 4-6 July 2012. IEEE, pp. 652 - 653.

DOI: 10.1109/ICALT.2012.199

© 2012 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.

Coverage metrics for learning-event datasets based on client-side monitoring

Derick Leony, Raquel M. Crespo, Mar Pérez-Sanagustín, Hugo A. Parada G., Luis de la Fuente Valentín,
Abelardo Pardo
Department of Telematic Engineering
Universidad Carlos III de Madrid
Leganés, Madrid, Spain
e-mail: {dleony, rcrespo, mmpsanag, hparada, lfuente, abel}@it.uc3m.es

Abstract: The collection of learner events within a server-client architecture occurs either at server, client or both complementarily. Such collection may be incomplete due to various factors, particularly for client-based monitoring, where learners can disable, delete or even modify their event logs due to privacy policies. The quality and accuracy of any analysis based on such data collections depends critically on the quality of the subjacent dataset. We propose three initial metrics to evaluate the completeness of a learning dataset: client-to-server ratio, event-to-activity ratio and subjective ratio. These metrics provide a glimpse on the coverage rate of the monitoring and can be applied to distinguish subsets of data with a minimum level of reliability to be used in a learning analytics study.

Keywords: *learning analytics; metric; coverage; completeness.*

I. Introduction

The collection of events in a learning environment that follows a client-server architecture can occur either at the side of the server, the client or both complementarily. A common example of a server-based collection is a Learning Management System (LMS) that records the interactions of learners with its functionalities (e.g. posting in forum, reading class resources, and answering a quiz) into *logs*. In a client-based collection, the logs are generated by the client applications, used by the students (or any monitored learning actor) and usually stored locally in a distributed manner. This paper focuses on the latter scenario. In our specific case, a virtual machine (VM) providing the set of required tools is given to the learners. The VM is configured to record and store the interaction of the learners with these tools; eventually, the log files are uploaded to a server and they are added to a dataset of events. A comprehensive explanation of this case study can be found in [1].

A client-based monitoring provides advantages such as gathering events related to tools commonly used offline (e.g. compiler and debugger in a computer science course) and gathering events generated while the learner was offline. On the other side, a relevant issue arises when applying a client-based collection because of the learner having the option to deactivate the recording of events. Then, the learner events are not collected in this case. This issue should be taken into account from the learning analytics point of view in order to identify from a raw dataset those subsets of events that are less reliable. This concern can be perceived in many learning analytics experiments as a pre-filtering stage of the experiment; some examples of data filtering are the ones performed in [1], [2].

These can be applied to different subsets of the dataset of learning events, in order to assess the coverage rate for different types of activities (browsing, edition, code developing, etc.) but also to characterise individuals, distinguishing those learners that provide more complete datasets. The usage of metrics for the classification of users has been already proposed in areas such as mass hyper-personalization [3].

II. Coverage Metrics

We define three generic metrics to characterise a learning event dataset: *client-to-server ratio*, *event-to-activity ratio*, and *subjective ratio*. To help their understanding, each metric is accompanied by examples related to the particular educational environment already described. These metrics have been defined according to a data-set extracted from the VM employed in the context of an actual programming course. The data-set includes events capturing URLs accessed from the web browser, program compilations, and the use of a versioning control system.

For the sake of brevity, we will refer to a tool installed in the VM and that generates events in both the server and the client simply as *tool*. Furthermore, we will refer to the recorded interactions of a learner with a *tool* as *tool events*.

A. Client-to-Server Ratio

Client-to-server ratio is the ratio of *tool events* recorded within the client to *tool events* recorded at the server. A value close to zero implies that the learner either disabled the recording of events from the *tool* in the VM or that a replacement of *tool* was used instead, either within or outside of the VM. This metric requires the *tool events* to be logged in both the clients and a server.

In our scenario one of the client-to-server ratios is the application on *tool events* generated by the web browser. In this case, client-to-server ratio is the ratio of accesses to the course web servers collected in the VM browser to web hits collected from the course website servers. This value provides an estimate of the amount of web browsing that the learner performed from the VM; the greater the value the greater the probability that students performed most of their course related browsing from the provided VM.

B. *Event-to-activity Ratio*

The next metric, *event-to-activity ratio*, is the ratio of all of the events to the occurrences of a specific learner action with a *tool*. Even though this metric can be applied with many purposes, a value of this close to zero commonly indicates a low usage of the learning environment.

In the scenario described above, the selected *tool* is the Subversion version controlling system and the specific action is the upload of files to the server, a mandatory activity for the learners. In this case the event-to-activity ratio provides an estimate of the amount of programming tasks done by learners within the VM.

C. *Subjective Ratio*

The metric *subjective ratio* is obtained by surveying a participant of the learning activity. The value of this ratio can be defined as a percentage or following a Likert scale. For example, learners can be asked to provide the ratio of the times they worked within the VM to the times they worked on any task related to the class.

Another example of the subjective ratio is surveying an external observer such as the class instructor. They can be asked to provide an estimate of the ratio of VM usage to total course-related activity for a given learner, based on the interactions that have occurred in face-to-face sessions.

The main purpose of the subjective ratio is to complement the information provided by the previous metrics. Thus, a subjective ratio provides a different perspective through the comparison with qualitative metrics.

D. *Level of granularity for the application of metrics*

The metrics presented above can be applied at two levels of granularity. First, when they are applied at an individual level, the metric helps to identify and filter individual learners in a dataset. All of the examples presented above are within this level.

The second level of granularity is the collective level, where the same metrics are applied over events generated by a group of two or more learners, providing a different meaning. An example of this level is the application of a client-to-server ratio on the interactions performed by dyads of learners. This metric then identifies those dyads whose combined activity in the VM seems reliable enough to be included in further analysis.

Finally, the application of any of the metrics explained above over a dataset includes the comparison of the metric value with another given value. In the case of a filtering process, the metric value is compared with a constant range of values; those learners whose metric is within the range of acceptance are included in the final filtered dataset for further analysis.

III. Discussion

The proposed metrics will enable us to filter large datasets by identifying learners with events reliable enough to be analysed. The application of these metrics will help practitioners in the field of learning analytics to bootstrap their analysis through an initial step of data filtering.

The definition of a metric threshold relies on the desired level of reliability for the resulting dataset and it varies between scenarios. The optimal values to use as point of comparison should be defined by means of experimentation. Also, experimentation could provide an insight of other ways to use the metrics, such as analysing the behaviour of learners through the comparison or combination of metrics applied on different *tools* as shown in Fig. 1.

Metrics can be defined from many perspectives, depending on the subject of study in a learning experiment. We propose a preliminary set including objective metrics, derived from the dataset itself, as well as subjective ones based on users' evaluation. Such generic metrics can be applied to specific subsets, filtered out by activity, or by the tracked actor. Also, they can be applied to different levels of granularity, covering individual as well as collaborative learning datasets.

Documenting and sharing the definition and application of metrics in experiments will help to enclose other similar scenarios. As future work, we propose the validation of this type of metrics using the current data-set and additional data-sets obtained from other educational contexts.

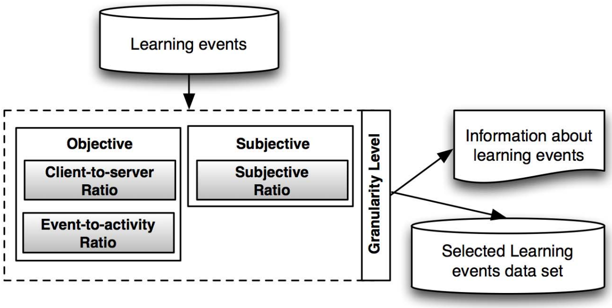

Figure 1. Metrics applications on a learning-event dataset

Acknowledgment

Work partially funded by EEE project "Plan Nacional de I+D+I TIN2011-28308-C03-01", Learn3 Project (TIN2008-05163/TSI), "Emadrid: Investigación y desarrollo de tecnologías para el e-learning en la Comunidad de Madrid" project (S2009/TIC-1650), and "Consejo Social - UC3M".

References

[1] V.-A. Romero-Zaldivar, A. Pardo, D. Burgos, and C. D. Kloos, "Monitoring student progress using virtual appliances: A case study," *Computers & Education*, vol. 58, no. 4, pp. 1058-1067, 2012.

[2] L. P. Macfadyen and S. Dawson, "Mining LMS data to develop an early warning system for educators: A proof of concept," *Computers & Education*, vol. 54, no. 2, pp. 588–599, Feb. 2010.

[3] E. Duval and K. Verbert, "The snowflake number," in *Proceedings of the WebSci'09: Society On-Line*, 2009.