

This document is published in:

Computers & Education (2012). 58 (4), 1058–1067

DOI: 10.1016/j.compedu.2011.12.003

© 2011 Elsevier Ltd.

Monitoring student progress using virtual appliances: A case study

Vicente-Arturo Romero-Zaldivar^{1,*}, Abelardo Pardo², Daniel Burgos³, Carlos Delgado Kloos²

¹ AtoS Research & Innovation, Albarracín, 25 28037, Madrid, Spain

² Department of Telematics Engineering, University Carlos III of Madrid, Av. Universidad, 30, 28911 Leganés (Madrid), Spain

³ International University of La Rioja, Gran Vía Rey Juan Carlos I 41, 26002 Logroño, Spain

* Corresponding author.

E-mail addresses: vicente.romero@atosresearch.eu (V.-A. Romero-Zaldivar), abel@it.uc3m.es (A. Pardo), daniel.burgos@unir.net (D. Burgos), cdk@it.uc3m.es (C. Delgado Kloos).

Abstract: The interactions that students have with each other, with the instructors, and with educational resources are valuable indicators of the effectiveness of a learning experience. The increasing use of information and communication technology allows these interactions to be recorded so that analytic or mining techniques are used to gain a deeper understanding of the learning process and propose improvements. But with the increasing variety of tools being used, monitoring student progress is becoming a challenge. The paper answers two questions. The first one is how feasible is to monitor the learning activities occurring in a student personal workspace. The second is how to use the recorded data for the prediction of student achievement in a course. To address these research questions, the paper presents the use of virtual appliances, a fully functional computer simulated over a regular one and configured with all the required tools needed in a learning experience. Students carry out activities in this environment in which a monitoring scheme has been previously configured. A case study is presented in which a comprehensive set of observations were collected. The data is shown to have significant correlation with student academic achievement thus validating the approach to be used as a prediction mechanism. Finally a prediction model is presented based on those observations with the highest correlation.

Keywords: Educational data mining, Learning analytics, Virtual appliances, Educational systems, Predictive systems

1. Introduction

An important factor that contributes toward the effectiveness of a learning experience is the ability of instructors to monitor the overall learning process and potentially act based on the observed events. In the ideal situation, an instructor monitoring all the events taking place in a learning environment would have a privileged position to adjust whatever parameters are available to improve the overall experience for the students. But this hypothetical scenario is still very far from reality in today's educational institutions and, even worse, there are several forces pulling away from this objective.

When moving away from traditional lecture-based teaching models into active learning strategies, the level of student participation increases. In active learning scenarios the student is at the center of the process, and the level of interaction among students, students and instructors, and students with course resources is essential for the success of the process. Deploying a comprehensive monitoring scheme in this scenario poses a serious challenge and would likely require an increase in the costs (either in time or resources) attached to the experience.

Orthogonally to this tendency, the presence of information and communication technology (ICT) in learning has been growing steadily in the last years. Technology is gaining ubiquity in the learning arena. Recent reports state that cloud-based technologies are gaining momentum in the educational arena (Johnson, Smith, Willis, Levine, & Haywood, 2011). With more material and activities being hosted in the cloud, the possibilities of sharing these resources increase enormously. Students and instructors may share a learning space in which information flows with virtually no effort.

Concepts such as "Learning Ecosystems" appear as the new holistic paradigm under which multiple aspects of the learning environment are considered (Uden, Wangsa, & Damiani, 2007). Such a rich set of interactions may seem as a fertile ground where comprehensive monitorization can be deployed, but this abundance comes at a cost. The ideas presented in this document aim at comprehensive monitorization in highly interactive, heterogeneous learning experiences.

1.1. Interactions in a learning scenario

In this emerging scenario where students and instructors have a high intensity interaction, tutoring is more challenging. We are evolving from scenarios where course activities were totally contained in a Learning Management System (henceforth LMS), to a set of heterogeneous (and usually distributed) tools used by all stakeholders during the learning process. This new environment may even host learning activities throughout a person life-span and in very different settings such as school, university, work, etc.

From an abstract point of view, a significant part of the activities that take place in any learning experience are interactions. Students interact with numerous resources (documents, forums, chats, specific tools), with the instructors, among themselves, etc. Interaction is identified as a crucial component of the learning process. Anderson and Garrison first categorized the different modes of interactions that can take place in a distance learning environment (Anderson & Garrison, 1998) (that can be applied to more generic learning scenarios) and later Anderson proposed an “equivalency statement” (or theorem as referred by the author) with two observations: deep and meaningful learning is supported if one of these forms of interactions is at its high level; and high levels of more than one of these modes will likely increase the quality of the experience although at a significantly higher cost (Anderson, 2003; Miyazoe & Anderson, 2010). Thus, following the equivalency statement we may assume that a successful learning experience has a high level of interaction in one of its aspects.

This theoretical result can be applied to improve the quality of a learning experience. Most learning experiences produce some measure of the student accomplishments. By first monitoring the events that occur during the experience and then analyze the relation between these observations and the derived measures of performance, we may gain insight into the details of the overall process and propose improvements. In other words, questions such as “How effective is a learning experience?”, “Is it adequate for each student?”, “How can it be improved?” as proposed in Elias (2011) apply to any experience and are desirable to foster improvement.

1.2. Academic/learning analytics and educational data mining

The combination of the previously described aspects (a move toward increasing student participation, the ubiquity of ICT, and the potential benefit of observing the learning process) has prompted the appearance in recent years of a body of research in which technology is used to monitor, analyze, deduct and act upon the deployment of a learning experience. As with most research areas, terminology is not as homogeneous as it would be desirable, but we could say that the three terms “Educational Data Mining” (EDM), “Academic Analytics” (AA), and “Learning Analytics” cover most of the work related with this scenario.

The term “Academic Analytics” became relevant in the research community after being the title of the article by Goldstein and Katz in which it is described as the application of business intelligence tools already in use in business and marketing to the area of learning (Goldstein & Katz, 2005). The main application of AA is to go beyond simple reporting and suggest better decisions in academic institutions by observing analyzing vast amounts of events occurring in their activities. The article characterizes AA as an “engine” working with five steps: capture, report, predict, act and refine. The examples provided in this work refer mostly to the problem of detecting “at risk students”, that is, those students that might drop out of a course or abandon their studies. There is a detailed description of how detecting these situations may benefit the institutions. An example of the use of AA is the Signals project in use at Purdue University (Arnold, 2010; Tanes, Arnold, King, & Remnet, 2011). A system has been developed to monitor the events recorded in the institutional LMS and apply measures to improve student success, retention and graduation rates.

Educational data mining (EDM) is a field that pursues a similar goal. It proposes the use of various techniques (statistical analysis, machine-learning, data mining) to resolve educational research issues and understand the setting in which students learn (Baker & Yacef, 2009). The use of ICT has prompted the appearance of vast amounts of events recorded during the learning process. Some of these data sets are being collected, labeled and organized to be shared within the research community (Koedinger, Cunningham, Skogsholm, & Leber, 2008). Data mining techniques are already used in other contexts (Fayyad, Piatetsky-Shapiro, & Smyth, 1996), but their application to educational contexts has been the target of an increasing number of research groups. Fortunately, various authors have written state of the art reviews that are very helpful to understand the topics, techniques and trends within this area (see Romero & Ventura, 2010) for an excellent review). The work in this area has emerged as a combination of the application of psychometric techniques to learning data, the widespread use of LMS with their recording capabilities and the presence of intelligent tutoring systems. The system described in this document can be included in the “Student Performance Prediction” category.

More recently, a third term “Learning Analytics” (LA) appeared to denote another view of the problem. As opposed to the view of AA where the captured data is used to make decisions at the institution level, the objective of LA is to use this data and any other additional observations that can be obtained, and use it to impact directly on the students, the instructors and the details of the learning process. For example, analyzing the interaction among students and prompting an immediate change or adaptation of the course material.

The difference among these three fields (AA, EDM and LA) is far from clear. Some authors claim that AA and LA are based on the premise that a hypothesis is proposed and the data is used to verify such hypothesis, whereas EDM works with no pre-conceived model and aims at discovering causalities among large portions of data (Baepler & Murdoch, 2010). Other authors see EDM as a generic process that can be applied to any aspect of a learning scenario whereas AA is a more concrete subset of techniques aiming at administrative and governing bodies, whereas LA is a subset aiming at teachers and students (Siemens, 2011). Both AA and LA may use data mining techniques to achieve their goals, so the lack of a clear distinction among these terms or areas is a sign of their proximity.

The aspect that all three areas have in common is that need for a vast amount of observations of the learning process. These observations may come from different sources, in different formats, at different times, and need to be processed so that any of the useful outcomes claimed by each area can be obtained. For example, student mental models can be detected by text analysis techniques (Rus, 2009). But one of the most important sources of information for these techniques are the events recorded by the LMS. If the course activities (specially in distance education) are all contained within the LMS, a detailed account of all the steps taken by the students can be derived from the recorded information. A description of the required steps to apply data mining techniques to this data is described in (Romero, Ventura, & Garcia, 2008). For example, in (Macfadyen & Dawson, 2010), Macfadyen and Dawson use the information stored by the LMS to first select only the factors that are more significant for prediction purposes and then propose a predictive model. Another example is presented in (Lin, Hsieh, & Chuang, 2009) where on-line discussions are automatically classified as announcements, questions, clarifications, etcetera, to simplify the cost of supervising these discussions. There are numerous contributions that rely on data gathered by the LMS to detect specific situations and derive interventions in the learning process.

But LMSs are proving to have limitations in their monitoring capabilities. On one side LMSs offer a limited reporting capability compared with what would be desirable, as stated in (Dawson, McWilliam, & Tan, 2008; Mazza & Dimitrova, 2007). Furthermore, the migration previously discussed of course interaction to cloud services makes the recording of events a more challenging task. This is especially true for courses using open-ended artifacts (Blikstein, 2011). Aspects such as the formulation of open activities, the lack of restrictions on the tools to use, the flexibility with respect to the environment used, etcetera. are seen as beneficial for the students but at the same time hinder the development of a simple monitoring mechanism.

An alternative to use the data obtained in an LMS is to embed mining techniques inside a tool designed to host the activities of a learning experience. An example is the use of mining techniques inside the adaptive web-based educational system Aha! (Romero, Ventura, Zafra, & Bra, 2009). In this case, the tool hosting the interactions is extended to collect the required data, analyze it and deduct the proper actions. But adaptive systems tend to be highly specialized and capture only a portion of the interaction that takes place during learning.

The variety of approaches and techniques for monitoring seem to point at a trade-off between coverage and generality. The more generic the technique, the poorer coverage of the learning activities. The work described in this document proposes to deploy a set of generic "sensors" closer to the environment in which students interact while maintaining the generality. In other words, instead of preparing a generic platform to record the interactions and then force activities to be carried out on that platform, the proposed approach consists on providing a monitoring mechanism as close as possible to the user environment and let the user interact with any available tool.

1.3. Virtual appliances in learning environments

Virtualization is the technique by which one computer is used to simulate, instruction by instruction, to a second computer. The simulated machines are called "virtual machines" or "virtual appliances" (the two terms will be used interchangeably in the document), and the software in charge of the simulation is called a "virtual machine monitor" (Goldberg, 1974). This technology has been present since the 1960s, when different systems needed to be simulated but required the execution of privileged instructions. Virtualization was the solution to offer a self-contained replication of the execution environment that allowed the simulation of complex systems. Although the concept and theoretical underpinning for this concept were solid, only a few virtualization platforms were implemented during the 1960s and 1970s mainly due to the lack of appropriate functionality in the hardware platforms.

It is during the 1990s when new techniques emerge to perform a more efficient mapping between the simulated machine and the underlying hardware. This, together with the need to execute applications in different platforms that can be achieved also with virtualization techniques, prompted what it has been called a "reincarnation" of the virtual machines (Rosenblum, 2004). This renewed interest has translated into the appearance of commercial and open source virtualization platforms as desktop applications that can be executed in a conventional personal computer.

Virtualization in education has also gone through a rebirth. Institutions used to deploy laboratories with costly equipment for students. But the increasing variety of tools and environments translated into cost increases. Virtualization is now being used to cope with this variety while maintaining equipment cost at reasonable values. Additionally, the features derived from using virtualization, isolation, compatibility and encapsulation, are perfect to deploy realistic learning environments while keeping their effects within the bounds of the experience. Bullers and colleagues noted that this situation was in fact a trend in education and showed several examples (Bullers, Burd, & Seazzu, 2006). As a consequence, the possibility of students working on a fully equipped machine designed for a course while executing in their conventional computers has widen the landscape of learning scenarios. Virtual machines are used for instance for debugging programs (King, Dunlap, & Chen, 2005), or operating systems courses (Laadan, Nieh, & Viennot, 2010) (to mention only two applications).

Given the encapsulated environment in which virtual appliances execute, they offer the possibility of monitoring all the events occurring internally. This possibility has already been used to record all the operations occurring in the desktop (Laadan, Baratto, Phung, Potter, & Nieh, 2007). In this document, virtual appliances are proposed as a resource to offer students to encapsulate all the related activities to a learning experience with no restrictions on the tools they use as long as they are installed in the appliance. There are several advantages derived from this approach. All students are guaranteed to work with identical copies of the machine. Additionally, installation of the required tools and configuration issues are solved once when the appliance is created. Also, the appliance (once the virtual machine monitor has been installed) appears to students as a regular desktop application that can be installed in their personal computers and they can remove at will.

The main idea in this work is to use these appliances to obtain a detailed recording of the events taking place in a learning experience. By offering the appliance fully configured for the course activities, students have a self-contained environment, yet highly versatile (it is a fully equipped computer) to work in the course. But at the same time, within the bounds of the virtual appliance, a multi-level comprehensive monitoring scheme can be deployed. In the following sections the process of creating, configuring and deploying a virtual appliance with this structure is described. The appliance was used by students during a semester capturing their activities. The collected information is shown to correlate significantly with student academic achievement.

2. Approach

The adopted approach consisted on distributing a previously configured virtual appliance in a traditional face to face semester course at the University Carlos III of Madrid and monitor the student activities related to this course. The two research questions tackled with this study are:

- Is it feasible to monitor the learning activities occurring in a personal student workspace?
- Can these data be used to predict student achievement in a course?

2.1. Scenario

The study was deployed in a second year engineering course at the University Carlos III of Madrid during the Fall semester of 2010. The course follows an active learning strategy. Students meet twice a week with teaching staff. For each session, a set of previous activities are defined. Students are supposed to work in these activities before attending the session in which this material is reviewed and expanded.

Among the learning outcomes expected from the course, one of them mentions explicitly the proficient use of certain tools. As a consequence, the course implies a significant load of practical work which students must carry out in their personal study time.

The environment in which the activities are conceived requires several tools and operating system especially configured for the course. For this reason, and to lower the entry barrier and focus solely on course activities, a virtual machine was created and configured specifically for the course. The appliance is configured with all the tools required for the course activities. The advantages of using virtual appliances in learning environments were discussed in Section 1.3. For the course under study, with a large number of activities that require the use of special tools, this solution is ideal because it allows students to start working on the course material with a minimum configuration phase. One of the first activities is a step-by-step tutorial to install the virtualization platform and deploy the virtual appliance.

The course material can be divided into two type of resources: course notes, and practical material. Course notes include the auxiliary documentation, handouts and supporting material used by the teaching staff. This material is contained in a virtual community hosted in an LMS. This community offers also support for course announcements and a discussion forum. Practical material includes handouts and auxiliary files. These last resources are managed through a version control system. At the beginning of the course each student is given a folder hosted in a remote repository that can be accessed using Subversion (Collins-Sussman, Fitzpatrick, & Pilato, 2004). The resources needed for the practical activities are made available in these shared folders by the teaching staff. Students download the initial version and regularly submit modified versions of these files. This folder is a de facto workspace because it contains all the practical material of the course already organized in different folders.

Version control systems provide an agile document exchange platform among students when working on a collaborative environment, and between students and teaching staff (Pardo & Delgado Kloos, 2011). The virtual machine has the client application to access the shared repository properly configured. Students only need a personalized URL denoting the location of the shared folder. All version control systems provide monitoring capabilities. Each operation over the repository is annotated with user name, time stamp and other additional information. There are tools that read these annotations and provide visual renderization of this information. But the objective in this approach was to gain insight not only in the operations to update resources in this workspace but on how these resources are manipulated when students work in an activity.

2.2. Data collection process

The data collection process was embedded in the virtual machine available to the students. The implementation consisted of two complementary modules: monitoring the activity of some selected tools, and relaying the observations to a central server.

2.2.1. Monitoring the use of certain tools

When designing the monitoring module the first design decision was to select only a subset of the tools available in the virtual machine to be monitored. An exhaustive monitorization of all the activity in the machine, although feasible (see (Laadan et al., 2007) for an example), it would make the collection of all the observations in a central server impossible. Instead, only the tools most commonly used by students to work in the course activities were selected. This collection included a browser, a text editor, a command interpreter and several additional programs related to the course topic.

The instrumentalization was implemented on a per-application basis, using two techniques. For those programs that record their activity in a known location, and these recordings are available, the instrumentalization was simple. At certain regular intervals, these recordings were copied into a specific area and prepared to be relayed to the server. This technique was applied, for example, to monitor the activity in the browser. The virtual machine included the Firefox browser that stores the history of URLs visited in an internal database in user space. At certain intervals the content of that database is read and those records that were added from the last check are stored in a folder in user space to be relayed to the server.

For those applications that do not record any of the internal events while they are being used, their behavior was modified to record the time in which they start and finish execution. The process required the creation of one “wrapper program” for each of these applications. This wrapper program has the same name as the application to be instrumentalized but executes the following tasks in this order:

- Records the date and time when the real application started.
- Invokes the real application and captures all the output messages produced (if any).
- Stores these output messages.
- Records the date and time when the real application finished.

The information was recorded in plain text files stored in a folder as in the case of the previous tools. From the point of view of the user interface, the behavior of the original application and the wrapper program are identical. For the applications considered in this study, the wrapper program posed no significant decrease in performance.

Once the wrapper programs were created, they were installed in a special folder in the user space. Finally, the environment variable controlling the locations in which the system searches for executable is modified to include this location before any other location. With this change, the execution of any of these tools will in fact start the wrapper program instead.

2.2.2. Collecting observations in a central server

The second module of the data collection procedure is in charge of sending the monitored events to a pre-defined server. There were important issues to be solved with the implementation of this module. In the context of the described study, data collection had a very specific time frame outside of which, no event should be recorded nor relayed. This requirement was adopted due to two reasons. First, the server was guaranteed to be on-line only during the duration of the course. Second, students were informed that the virtual appliance was used only for the course activities, and therefore, at the end of the course no more information was recorded nor sent to the server (a more detailed description of the adopted policy is included in Section 2.3). Additionally, the module could not assume that the virtual appliance had an Internet connection. Students may use the machine in their personal computers with no Internet access.

Instead of developing a special communication channel to relay the data to the server, the solution adopted was to instrumentalize an application already used regularly by the students. The version control system Subversion was installed and used regularly by students to

exchange resources required to work in the course activities. All the operations are executed with a client program called “svn”. This program was modified by creating a “wrapper program” similar to those described in Section 2.2.1. Instead of recording the execution times, this wrapper program detects when the users are sending a new version of their personal files and expands this transaction with the central server to include the events collected from the last operation.

Another issue that required to be solved is that all students use an identical copy of the same virtual machine. As a consequence, events from different users would be impossible to distinguish. This information with differentiation among users could be useful for inferring usage patterns for the entire student population. But the objective is to see if these events can be used to predict student achievement in the course. Thus the events need to be kept differentiated for each user. This means that the server needs to receive the data together with a user identification. This problem is solved by using Subversion as the communication channel. Each transaction with the central repository requires the user to introduce some credentials that are reflected in the corresponding log stored in the server.

2.3. Compliance with data privacy regulations

The described data collection procedure must be in compliance with current data privacy legislation. In the case of the European Union this legislation follows Council Directive 1995/46/EC (European Community, 1995) on the protection of individuals with regard to the processing of personal data and on the free movement of such data. Member States have applied national provisions to comply with this directive. In Spain, the monitoring process must comply with the Organic Act 15/1999 on Personal Data Protection (Boletín Oficial del Estado, 1999; Vilasau, 2003) (or its Spanish acronym LOPD). The most relevant aspect of this law that apply to this study is that students must be informed explicitly, precisely and unequivocally of the following:

- The existence of a file of personal data, the purpose of collecting the data, and the recipients of the information.
- The voluntary nature of the use of the monitoring mechanism.
- The consequences of obtaining the data or of refusing to provide them.
- The possibility of exercising rights of access, rectification, erasure and objection.
- The identity and address of the controller or of a representative, if any.

Before downloading the virtual machine for the course, a web page is shown to the students. The document explains the details of the monitoring mechanism, the events that are recorded, the use of the collected data and the procedure to disable it should a student choose to do so. The machine is configured so that it shows this same document in the browser upon start-up as shown in Fig. 1. Students must agree with these conditions before downloading the machine. Alternatively, a machine with no monitoring mechanism is also available.

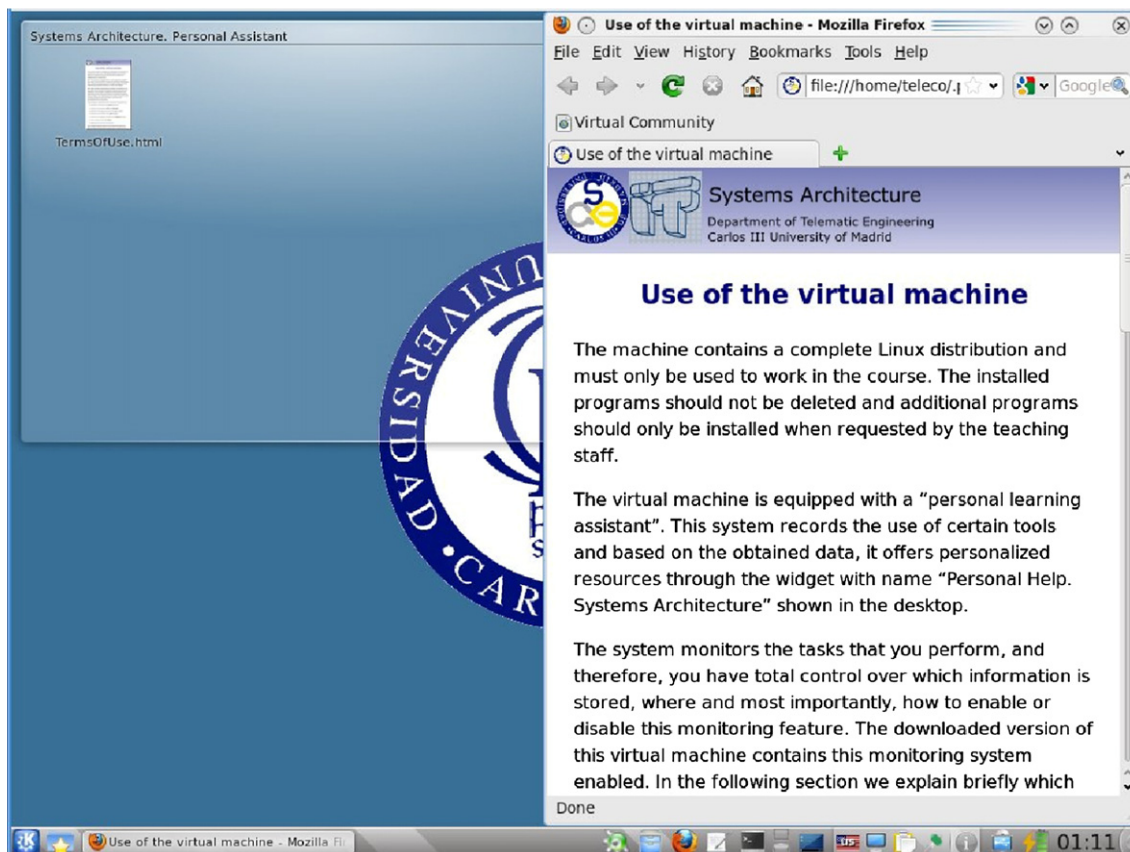


Fig. 1. Initial screen of the virtual machine.

A copy of all the recorded events is kept in plain text format inside the student folder managed with the version control system. With this approach, should the need arise, students may exercise the right to review, amend or delete any of the collected events by modifying or deleting the corresponding files. The compliance with current regulation on personal data translates basically into giving students total control over the monitoring mechanism. Although this policy may have a significant impact on the amount of data collected, it was considered part of the study to answer the first stated research question about the feasibility of the monitoring approach.

3. Results and discussion

3.1. Recorded events

The monitoring mechanism was deployed during a 15 week period from September to December 2010. Table 1 shows a summary of student participation in the study. From a total of 248 students enrolled in the course, events were recorded for 172 of them, a 69.35%. However, of the 76 students from which no event was recorded, 30 participated in the course activities while the rest dropped the course, thus the true participation percentage is 85.15%.

Students work regularly in homework assignments that require the use of three special tools: a compiler, a debugger and a memory profiler (to detect misuse of dynamically allocated memory). Additionally, they need to execute basic administrative tasks using a command interpreter. During the activities, they are required to check various information sources. The monitoring mechanism was tailored to precisely record the use of these tools. Additionally, a post-processing step was conducted on the collected events to derive the time in minutes that students were working with the virtual machine (event intervals larger than 15 min were ignored).

Table 2 shows the summary of the collected information. The row "WorkTime" shows the time in minutes using the virtual machine. The "Command" row shows the events recorded every time a student writes a command in the interpreter. Events of type "Visit URL" are recorded every time a web page is opened with the browser. Events of type "Editor" and "IDE" are recorded when students open a file with an editor. Finally, events of type "Compile", "Profiler" and "Debugger" are recorded when students invoke the C compiler, a memory profiler, and the C debugger respectively. The second and third columns show the number of events recorded and the percentage of all the collected events. The four right-most columns contain a summary of the distribution of each factor.

The first observation about these data is that, on average, 521.27 events were recorder for each student. However, as the value of the standard deviation suggest, this data has a high spread, which suggests an uneven distribution of these recordings. In other words, the monitoring scheme offers a "coverage" that varies from student to student. The distributions in each row confirm this observation. Still, events of type "Compile" and "Editor" have a more compact distribution. The conclusion is that although a large set of events were recorded, there might be differences among event types in the percentage of events truly recorded for each student.

Thus, regarding the first research question posed in the paper, we conclude that monitoring the learning activities occurring in the student workspace is feasible as long as a measure of the reliability of this monitoring is obtained. Such measure can be computed using the information about how the events were received in the server. As explained in Section 2.2.2, events are sent to the server whenever students upload a new copy of any file in their shared folder. Each update operation is in fact comprised of two sub-operations: update the files requested by the user, and upload the files containing the recorded events. These operations are executed and logged separately in the server. Thus, the coverage of the monitoring for a student s , denoted by cov_s is defined as

$$cov_s = 1 - \frac{UP_f}{UP_e + UP_f}$$

where UP_e is the number of event upload operations and UP_f is the number of file update operations. These numbers are obtained from the logs registered by the version control system. cov_s has a value between 0 and 1. A student who disabled the event monitoring mechanism for the entire course will have $UP_e = 0$ and thus $cov_s = 0$. However, when working normally, the monitoring process generates an event upload operation for every file upload executed by the user. As a consequence, a coverage value of 0.5 denotes a fully monitored user environment. Values larger than 0.5 are also possible because of update operations that send recorded events but no changes in the user files. Fig. 2 shows the distribution of cov_s only for students with non-zero coverage.

The distribution shows how a significant number of students have a coverage value below 0.5. As it will be shown in the next section, this figure of merit is useful to study the correlation between the recorded events and student achievement.

3.2. Correlation with student academic achievement

The correlation between the observed events and the academic achievement was performed based on the number of observed events and their type. For this reason, not all students in the course can be taken into account. Those with a low coverage value will have a low count on the number of events and the connection time while having an active participation in the course. For this reason, the following analysis was performed considering only those students with $cov_s \geq 0.40$ or higher (a total of 79 students).

Table 1
Student participation in the study.

Students enrolled in the course	248
Students downloaded the virtual machine	220
Students participated in the activities	202
Students with recorded events	172

Table 2
Summary of the recorded events.

Event type	Count	Percentage	Per student			
			Min.	Mean	Max.	SD
WorkTime			0	380.10	2008.57	409.46
Command	29,452	32.85%	0	171.23	1003	205.34
Visit URL	27,076	30.20%	0	157.42	946	200.03
Compile	22,826	25.46%	0	132.71	870	154.20
Editor	7449	8.31%	0	43.31	300	44.60
Profiler	2466	2.75%	0	14.34	142	24.99
IDE	238	0.27%	0	1.38	42	3.87
Debugger	152	0.17%	0	0.88	26	2.79
Total	89,659	100.00%	9	521.27	2822	520.71

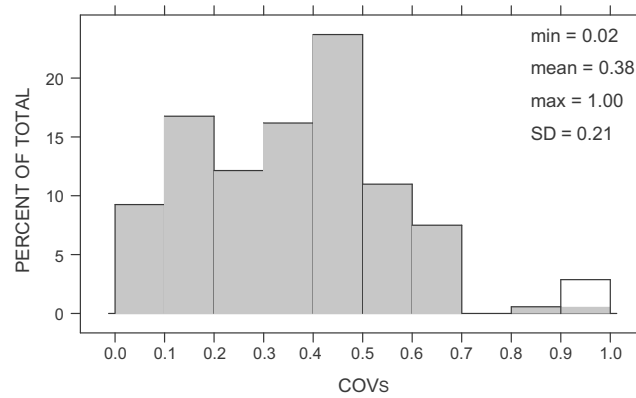


Fig. 2. Distribution of cov_s for students with any event recorded.

The first step was to analyze the significance of the observed categories as indicators of the student performance. Table 3 shows the simple bi-variate correlations between each of the collected events and the final course grade, the coefficient of determination, and the significance of this correlation.

Six of the analyzed factors show a positive and significant correlation with the student grade ($p < 0.05$). This data is consistent with the nature of the course. For instance, events of type “Command” are not directly related to tasks required to complete the activities but a high number of these events will likely derive from an intense student involvement in the activities. The analogous situation occurs with the events of type “WorkTime”, “Compile”, “Editor”, “Profiler”, and “IDE”. These events are recorded when a students executes the application which are needed to work in the course activities, thus, the relation with course activity is direct. The events of type “URL” (a page is visited with the browser), on the other hand, do not reflect so reliably this involvement. Students may (and in fact they do) use the browser to check web pages that are not related to the course material. A high number of these events does not necessarily mean that a student is working in the course activities. The last factor, “Debugger”, reflects an interesting situation. Although students are encouraged to use this tool for convenience, the low event count suggests that the advice is being systematically ignored.

3.3. Multiple regression analysis

Once the correlation of some of the recorded factors has been established, a finer grain prediction mechanism is desirable. A multiple regression analysis was applied to obtain the equation to predict the final grade. All factors were considered as potential factors to enter the predictive model. A forward and backward stepwise regression procedure was computed. In both cases, the linear model shown in Table 4 was obtained.

Of all the recorded factors, only two of them, “WorkTime” and “Profiler” were selected for the linear model. The linear coefficient for the “WorkTime” shows a significance under 0.05 and the coefficient for the “Profiler” slightly over 0.05 but still under 0.1. The multiple squared

Table 3
Simple correlation of observed factors and final grades ($N_{\text{students}} = 79$).

Event Type	r_s	r^2	p
WorkTime	0.43	0.18	0.00
Command	0.27	0.07	0.01
Visit URL	0.17	0.03	0.13
Compile	0.40	0.16	0.00
Editor	0.37	0.14	0.00
Profiler	0.37	0.14	0.00
IDE	0.26	0.07	0.02
Debugger	0.13	0.02	0.24

Table 4
Multiple regression analysis summary ($N_{\text{students}} = 79$).

Variable	Estimate	SE	t value	Pr (> t)
(Intercept)	42.2785	2.5112	16.84	0.0000
WorkTime	8.0717	2.8625	2.82	0.0061
Profiler	5.3866	2.8625	1.88	0.0637

$r = 0.4665$, $r^2 = 0.2176$, $p < 0.01$.

Table 5
Linear regression analysis of forum posts ($N_{\text{students}} = 79$).

Variable	Coefficient	SE	t value	Pr (> t)
(Intercept)	39.9186	3.0731	12.990	<2e-16
Number of posts	1.4017	0.7962	1.761	0.0823

$r = 0.1967$, $r^2 = 0.0387$, $p = 0.08$.

correlation coefficient for the model is 0.22 which means that the model accounts for more than 20% of the variability of student final grades. These results answer the second research question posed in this paper. The data obtained by the monitoring procedure can be used to derive a linear model with significant coefficients to predict the final course grade.

In order to compare the effectiveness of this approach, the obtained model was compared with one based on observations derived from the LMS. In a face to face environment, the LMS does not collect as much information as in a distant education course. For example, email exchange among students or between students and instructors usually takes place outside of the LMS. Furthermore, all submissions in this course are done using the shared folder. Thus, the activity in the LMS is reduced to a discussion forum. Table 5 shows the linear model computed to predict the final grade taking into account the number of posts in such forum.

As it can be seen, the model based in the number of posts has the significance of its linear coefficient with a value above 0.01. With this comparison we conclude that, for face to face learning scenarios, monitoring activities in the student workspace offers a better estimation of their academic achievement.

The derived linear model can be used to implement an early warning mechanism in the spirit of the one proposed by Campbell and colleagues (Campbell, DeBlois, & Oblinger, 2007). The data collected by the system combined with the obtained model are combined to detect students that would potentially fail the course, or “at risk” students. The fact that observations are recorded and collected in the central server with certain frequency allows this system to be running during the course.

4. Summary and future work

This paper has presented the use of virtual appliances to offer students a reasonable trade-off between installing the tools required for a course in their personal computers and using special equipment that is not always available. Virtual machines can be created to contain a fully configured course environment containing all the required tools. Students install these machines in their personal computers and perform all course activities with them. Such self-contained environment is ideal to observe how students carry out the course activities. This scenario offers an ideal platform to deploy a monitoring scheme. The set of applications required to work on the course can be instrumentalized to record when are they used and (for some of them) the type of events occurring while in use. These events can then be collected in a centralized server through a communication channel embedded in the machine.

The use of this approach has been deployed in a second year engineering course during the Fall 2010 semester. The course required students to complete numerous practical activities using some special tools. The type and number of recorded events confirmed that the monitoring scheme was effective. Almost 90,000 events were recorded for a total of 172 students. Thus, we conclude that the approach can be successfully deployed in a real course and students embrace it with no problems.

Data protection regulations imposed certain requirements on how the mechanism was deployed leaving the students the decision to disable the monitoring at any point in time. And the different level of engagement of students in the course made that the number of events recorded per student had large variations. A coverage measure was proposed to quantify the extent of the monitoring process for each student. With this figure of merit, the students for which the monitoring process was more extensive were identified and selected to derive a predictive model.

Out of the seven types of events recorded, three of them had a significant correlation with the course final grade. A multiple regression analysis was performed and the resulting model considered only two of these factors. These events correspond with the use of two of the specific tools required to work in the course activities. The obtained model accounts for more than 20% of the variability of the final grade. Additionally, the model was compared with a linear regression using the number of messages posted by students in the course forum. The model obtained with the observations collected in the virtual machine was more robust. The second conclusion is then that the detailed observation of the student activities in their course workspace offers a reliable framework to predict their academic achievement.

There is some further analysis that can be drawn from these results. As described in Section 2.1, the course requires students to achieve a proficient use of certain tools. Two of them are the profiler and the debugger. The obtained results clearly show that the use of these tools by the students is very different. The use of the profiler is actually one of the variables in the derived linear model. However, the debugger is barely used throughout the course. The monitoring approach has two positive effects. First, instructors have a reliable mechanism to observe how frequently a tool is used. Second, if the results prompt adjustments on the activities or the evaluation scheme (for example, a test on debugger functionality), the impact of these changes can be quantified with the monitoring scheme.

4.1. Limitations of the study

The example used to illustrate the described approach was deployed in a concrete scenario, thus introducing certain limiting aspects. The course is part of an engineering degree. Engineering courses typically have a high amount of practical work for which certain specific applications are required. These two factors mean that students need to work on practical assignments frequently, and that they need specifically configured tools to do so. This advantage though, does not disappear completely in a generic course. It is very likely that courses contain activities, and the probability of requiring a computer to work on them is increasing.

Compliance with current data protection legislation imposes important boundary conditions. Students must be able to disable the monitoring mechanism at any point, use a similar platform and not the virtual appliance, or request their data to be erased. These aspects have been taken into account by measuring the coverage of the monitoring procedure.

Another additional limitation of the approach is that in order to record meaningful events, the tools need to be instrumentalized one by one. If the set of tools changes from one course edition to another, the monitoring module needs to be modified as well. This is a technological limitation that could be overcome if the recording is performed at a deeper level in the appliance. More precisely, the operating system could offer the required functionality to record any event for any application. With this approach, changes in the type of tools used would not affect the monitoring scheme.

4.2. Future work

The data collected with the described approach has been analyzed only from the quantitative standpoint (event counts), but there are some interesting venues to explore when considering a qualitative analysis. The events recorded with the browser offer a glimpse of the type of documents that students browse while working in the activities. A significant proportion of these documents will be part of the official course material, but there will be additional documents not part of the course but related to the topic (Romero-Zaldivar, Crespo García, Burgos, Delgado Kloos, Pardo, 2011). The collected events could be used in a crowd-sourcing approach to detect resources that are not included in the course material but are relevant to the topic.

The collected events as a whole could also offer some insight on the interaction among students, especially when activities are carried out in teams. Events may be used to detect patterns that suggest dysfunctional teams.

Acknowledgments

Work partially funded by the Learn3 project ("Plan Nacional de I+D+I" TIN2008-05163/TSI), TELMA Project ("Plan Avanza", TSI-020110-2009-85), and the "Emadrid: Investigación y desarrollo de tecnologías para el e-learning en la Comunidad de Madrid" project (S2009/TIC-1650).

References

- Anderson, T. (2003). Getting the mix right again: an updated and theoretical rationale for interaction equivalency of interaction. *The International Review of Research in Open and Distance Learning*, 4(2).
- Anderson, T. D., & Garrison, D. R. (1998). Learning in a networked world: new roles and responsibilities. In C. Gibson (Ed.), *Distance learners in higher education* (pp. 97–112). Atwood Publishing, Ch. 6.
- Arnold, K. (2010). Signals: applying academic analytics. *EDUCAUSE Quarterly*, 33(1), 10.
- Baepler, P., & Murdoch, C. J. (2010). Academic analytics and data mining in higher education. *International Journal for the Scholarship of Teaching and Learning*, 4(2).
- Baker, R., & Yacef, K. (2009). The state of educational data mining in 2009: a review and future visions. *Journal of Educational Data Mining*, 1(1), 3–17.
- Blikstein, P. (2011). Using learning analytics to assess students' behavior in open-ended programming tasks. In *Proceedings of the 1st international conference on learning analytics and knowledge*.
- Boletín Oficial del Estado. (1999). *Ley Orgánica 15/1999, de 13 de diciembre, de Protección de Datos de Carácter Personal*.
- Bullers, W., Jr., Burd, S., & Seazzu, A. (2006). Virtual machines—an idea whose time has returned: application to network, security, and database courses. *ACM SIGCSE Bulletin*, 38(1), 102–106.
- Campbell, J., DeBlois, P., & Oblinger, D. (2007). Academic analytics: a new tool for a new era. *Educause Review*, 42(4), 40–57.
- Collins-Sussman, B., Fitzpatrick, B., & Pilato, C. (2004). *Version control with subversion*. Cambridge, MA, USA: O'Reilly Media.
- Dawson, S., McWilliam, E., & Tan, J. (2008). Teaching smarter: How mining ICT data can inform and improve learning and teaching practice. In *Proceedings of the Australasian society for computers in learning in tertiary education* (pp. 221–230).
- Elias, T. (2011). *Learning analytics: Definitions, processes and potential*. <http://learninganalytics.net/LearningAnalyticsDefinitionsProcessesPotential.pdf> Accessed: August 2011.
- European Community. (1995). *Directive 95/46/EC of the European Parliament and of the Council*.
- Fayyad, U., Piatesky-Shapiro, G., & Smyth, P. (1996). From data mining to knowledge discovery: an overview. *AI Magazine*, 37–54.
- Goldberg, R. (1974). Survey of virtual machine research. *IEEE Computer*, 7(6), 34–45.
- Goldstein, P., & Katz, R. (2005). *Academic analytics: The uses of management information and technology in higher education*. Tech. Rep. December. EDUCAUSE Center for Applied Research.
- Johnson, L., Smith, R., Willis, H., Levine, A., & Haywood, K. (2011). *The horizon report*. Tech. rep. The New Media Consortium.
- King, S. T., Dunlap, G. W., & Chen, P. M. (2005). Debugging operating systems with time-traveling virtual machines. In *Proceedings of the annual conference on USENIX annual technical conference* (pp. 1).
- Koedinger, K., Cunningham, K., Skogsholm, A., & Leber, B. (2008). An open repository and analysis tools for fine-grained, longitudinal learner data. In *International conference on educational data mining* (pp. 157–166).
- Laadan, O., Baratto, R. A., Phung, D. B., Potter, S., & Nieh, J. (Oct. 2007). DejaView: a personal virtual computer recorder. *ACM SIGOPS Operating Systems Review*, 41(6), 279.
- Laadan, O., Nieh, J., & Viennot, N. (2010). Teaching operating systems using virtual appliances and distributed version control. In *Proceedings of the 41st ACM technical symposium on Computer science education – SIGCSE '10* (pp. 480).
- Lin, F., Hsieh, L., & Chuang, F. (Feb. 2009). Discovering genres of online discussion threads via text mining. *Computers & Education*, 52(2), 481–495.
- Macfadyen, L. P., & Dawson, S. (Feb. 2010). Mining LMS data to develop an "early warning system" for educators: a proof of concept. *Computers & Education*, 54(2), 588–599.
- Mazza, R., & Dimitrova, V. (Feb. 2007). CourseVis: a graphical student monitoring tool for supporting instructors in web-based distance courses. *International Journal of Human-Computer Studies*, 65(2), 125–139.
- Miyazoe, T., & Anderson, T. D. (2010). The interaction equivalency theorem. *Journal of Interactive Online Learning*, 9(2), 94–104.
- Pardo, A., & Delgado Kloos, C. (2011). SubCollaboration: large scale group management in collaborative learning. *Software: Practice and Experience*, 41(4), 339–465.
- Romero, C., & Ventura, S. (2010). Educational data mining: a review of the state of the art. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, 40(6), 601–618.
- Romero, C., Ventura, S., & Garcia, E. (Aug. 2008). Data mining in course management systems: Moodle case study and tutorial. *Computers & Education*, 51(1), 368–384.

- Romero, C., Ventura, S., Zafra, A., & Bra, P. D. (Nov. 2009). Applying Web usage mining for personalizing hyperlinks in Web-based adaptive educational systems. *Computers & Education*, 53(3), 828–840.
- Romero-Zaldivar, V. A., Crespo García, R. M., Burgos, D., Delgado Kloos, C., & Pardo, A. (2011). Automatic discovery of complementary learning resources. In: *Proceedings of the European conference on technology enhanced learning* (pp. 327–340).
- Rosenblum, M. (Jul. 2004). The reincarnation of virtual machines. *Queue*, 2(5), 34.
- Rus, V. (2009). Automatic detection of student mental models during prior knowledge activation in MetaTutor. In *Proceedings of the 2nd international conference on educational data mining* (pp. 1–10).
- Siemens, G. (2011). *Learning and academic analytics*. <http://www.learninganalytics.net/?p=131> Accessed: August 2011.
- Tanes, Z., Arnold, K. E., King, A. S., & Remnet, M. A. (Dec. 2011). Using signals for appropriate feedback: perceptions and practices. *Computers & Education*, 57(4), 2414–2422.
- Uden, L., Wangsa, I., & Damiani, E. (2007). The future of E-learning: E-learning ecosystem. In *Proceedings of the digital ecosystems and technologies conference* (pp. 113–117). IEEE.
- Vilasau, M. (2003). The right to privacy and to personal data protection in Spanish legislation. *Computer and Telecommunications Law Review*, 9(7), 196–201.