AN ARCHITECTURE FOR DYNAMIC E-LEARNING ENVIRONMENTS BASED ON STUDENT ACTIVITY AND LEARNING STYLES

John A. Kaliski, Minnesota State University Mankato
Queen E. Booker, Minnesota State University Mankato
Paul L. Schumann, Minnesota State University Mankato

ABSTRACT

Using e-learning systems, computer assisted technologies, or learning management systems to supplement or replace the classroom experience is becoming more common in education. The use of these technologies generates a large volume of transactional data that record how each student progressed through the learning materials in the e-learning system. This data, which is currently underutilized, could be used to understand student learning behaviors, and to help both the instructor and the student benefit more from the course content. This paper describes an architecture using business intelligence methodology for using the data captured by e-learning systems to understand what students are doing (or not doing) in the e-learning system, and thereby to make changes that enhance student learning.

JEL: C88, I21, M00, M19, Z00

KEYWORDS: E-learning, Learning Management Systems, Business Intelligence, Business Education, Business Education Research

INTRODUCTION

Learning technologies encompass a broad range of communication, information, and related technologies that can be used to support learning, teaching, and assessment. Learning technologies are common in the education industry as parents, students, and teachers use the technologies to try to promote and improve learning. The use of learning technologies in higher education has grown significantly over the years, ostensibly because the use of learning technologies, particularly learning management systems (LMSs), provides opportunities for instructors to have a flexible learning environment for students. LMSs provide an opportunity for instructors to allow students to access course content 24-7; this content can range from quizzes and exams to online lectures to a wide variety of learning experiences limited only by the technology and the instructor's creativity.

According to U.S. Department of Education, National Center for Education Statistics 2008 (2011), 97 percent of public 2-year universities offered web-based learning programs, followed by public 4-year institutions at 89 percent. The majority of the students enrolled in online learning courses was students with some type of personal responsibility such as a spouse or child, were older, employed, or had mobile disabilities. The students were more likely to access the learning system during non-traditional hours such as after 5PM or before 8AM when people, including faculty, typically work. While the flexibility in the online access provides opportunities for students to complete a class at any time, the challenge is that the instructor is not always available to help them when they might need it most. Thus, there needs to be a support system that can be available to the student whenever they might choose to engage in the learning system.

A yet underutilized component of the LMS is the transactional data captured through the use of the system. The use of LMSs to manage course content allows for the capture of student behaviors such as accessing reading materials and teaching notes, when students start online assignments, how long students take to complete the assignments, and productivity in discussion forums. Some LMSs also capture how often and how long students log into the system. While many LMSs capture this information, little is known about how to appropriately use this information to better understand student behaviors or e-learning usage patterns, and how to use the information to create a better learning environment for the student. This paper describes an architecture for understanding students' e-learning usage pattern, and how to utilize that information to improve how LMSs can be used to improve student learning.

This paper specifically discusses an approach to using the data available through an LMS to provide students with information that can improve their performance and their learning. The paper shows that by using the data captured by an LMS, the LMS environment can be enriched, and student learning can be enhanced, by using a combination of historical data and automated learning consultants that use business intelligence methodology to help encourage students through their learning activities.

The next section of the paper provides a literature review that shows existing research on LMSs focus on the development of LMSs and how they are used. We find that the literature does not currently examine how the detailed transaction-level data about what students do in the LMS can be used to enhance learning. Then we outline our proposed architecture for monitoring student activities in a LMS. The Data and Methodology section describes the development of metrics that measure student activity in the LMS and the use of business intelligence methods to automate the provision of timely feedback to the students as they progress through the learning activities in the LMS. The Results section presents examples of the application of our proposed architecture. The last section of the paper provides some concluding remarks.

LITERATURE REVIEW

Because of the rapid increase in the use of the Internet, the delivery of computer based learning programs has rapidly shifted from local desktop to online-based applications. The online learning applications allow the instructor to make modifications to the learning content in one place at one time, and the revised content is then made available immediately to all students rather than waiting for the next time the course is offered, thus saving both time and costs (Naidu, 2006; Cavus, 2011). Online computer based learning systems, or simply learning management systems (LMSs), are software applications that comprise a suite of tools for learning and teaching online.

An advantage to online learning from the learner's perspective is that they get access to the learning content 24-7 (Cavus, 2011). Much research has been done on the convenience and flexibility, and the concomitant user satisfaction, with LMSs, especially among non-traditional students (Naidu, 2006). Research on LMSs has also examined the user adoption of the technology (Yu and Yu, 2010), and the development of the LMS applications and systems (Fontela et al, 2010).

In terms of convenience and flexibility, researchers consistently point to results that show many students prefer, and in some cases even expect, instructors to use LMSs to enhance the learning experience. Revere and Kovach (2011) state in their study, "[T]he effectiveness of course design and student engagement remains uncertain. To deliver the highest quality online education, students should be engaged in learning exercises. Appropriately integrated technology can be used to foster student engagement, build a learner-centered environment, and make course content come alive." They discuss the use of technologies such as discussion boards, chat sessions, blogs, Twitter, Skype, YouTube, and others, to provide guidance to educators interested in integrating these tools within their online learning environment. They show that instructors who effectively incorporate technology as learning tools in their online courses can expect to achieve enhanced student engagement as well as higher levels of learning and more efficient classroom management (Revere and Kovach, 2011).

A study by Yu and Yu (2010) examined the adoption of LMSs. They discuss factors related to the adoption of LMS, and determined that three independent variables associated with theories of planned behavior contributed to student use and to student attitudes toward the use of the technology: attitudes, subject norms, and perceived behavior controls (Yu and Yu, 2010). Other studies, such as Barki et al. (2007), Goodhue and Thompson (1995), McGill and Hobbs (2008), McGill and Klobas (2009), Staples and Seddon (2004), and Pagani (2006), have suggested that learner technology fit regarding the system is associated with positive attitudes toward LMSs; user satisfaction encourages use, which in turn encourages active participation in the learning environment.

An example of the development research can be found in Fontela et al. (2010). In their paper, they describe an architecture for LMSs that overcomes the problem of trying to use both a hosting system's LMS (such as Blackboard or WebCt) and a content provider's LMS (such as the publisher Wiley's online content). Using both systems typically requires the instructor to manually integrate, or basically collect student performance data in one system and then manually transfer the results to the other system, to get the full picture of the student's performance. Creating a "hard" integration of third-party tools allows the primary LMS to have tighter control over external tools and provides an opportunity for the instructor to manage different issues needed to better understand and control what the student is doing and when (e.g. events, permissions, and sessions).

LMSs can collect a wide variety of transactional data concerning how students navigate the LMS and how students use (or fail to use) the learning content in the LMS. The data can include whether a particular student accessed a particular piece of learning content, how long the student accessed the content, and the results of any assessments of student learning based on the content.

While the existing literature examines how to develop LMSs, how people adapt to them, and user performance when using them, a limitation of the literature is that it has not examined how to use the student transactional data from the LMSs. In particular, the literature has not yet described a methodology to identify particular student behaviors that successful students engage in while in the LMS, and then how to use that information to guide other students towards a better learning experience that would result in better student performance and higher student knowledge retention.

A reason for this gap in the research on LMSs is in part due to the difficulty of conducting conclusive studies suggesting that a student would not have performed as well, or as poorly, in a non-online environment as in an on-line environment because all conditions for learning would have to be exactly the same, including the student. Thus, the ability to conduct a pure impact study is limited to pre-test/post-test analyses for learning within the class, and a post-post test analysis for knowledge at an even later period.

As Van Nijlen and Janssen (2011) reported, mastery is essential when monitoring student progress and is crucial for instructional interventions to deal with learning difficulties. Even more important, instructors cannot always detect when students are having difficulty in the online environment unless the student specifically tells the instructor that such is happening. When a student is studying and the student begins to have trouble, only when the instructor is available is the student able to ask the instructor for directions for improvement. The difference in the online learning environment is that the student could have this corrective intervention without the instructor if the system is able to identify the student is having a problem based on his or her series of recent actions and activities, and if the system has the intelligence built into the system to recommend to the struggling student a corrective course of action.

A PROPOSED ARCHITECTURE

The use of e-learning technologies such as LMSs has strengths and weaknesses. One possible strength is in the ability to manage and grade large numbers of assignments, thus making larger classes a bit more manageable for instructors. Another possible strength is that LMSs centralize access to learning materials, assignments, discussion forums, and learning assessments; thus, LMSs allow for learning materials to be available to students 24/7, and for the materials to be easily updated and instantly distributed.

LMSs also can collect detailed data on how student progress through the learning content and assessments. The instructor potentially has a rich dataset of transactions that capture student behaviors in the LMS, including information about when the student logs in to the system, what the student did while logged in, and the results of assessments. Some learning platforms can tell the instructor the proximity of IP addresses of students; some can show the instructor what other websites the student accessed while the student was logged in to the LMS. For example, the instructor could determine which websites outside of the LMS that a student accessed while the student was logged onto the LMS and presumably working on the learning content: Where those websites relevant to the LMS learning activity, or did the student spend most of the time accessing websites unrelated to the learning activity (e.g., Facebook)?

The weakness of this LMS data is that an instructor might be perceived as invading a student's privacy. A student may not want the instructor to know how frequently they logged in and what they were doing while logged in. While the data generated can help the instructor better understand and identify behaviors of the student while online, students may not appreciate the instructor knowing so much about them, especially if the student does not know the instructor well and does not trust that the instructor is using the information to benefit the student.

Despite the capabilities of learning management systems to report information that could aid in the management of classes and student behavior, some LMS only offer minor reporting. As an example, a typical LMS offers a series of student activity reports that include information related to student login, checklist, content, discussion, dropbox, grades, quizzes, and surveys. The instructor can see how many times a student has accessed each area of the learning environment. These reports simply give a list of relevant activities by the student and some summary statistics. For example, the login report for a student presents a list of time stamps and IP address locations from which the student logged in.

Few of the standard reports offer more than either raw data or a superficial summary of the student activity. The reports do not indicate when the student began the assignments, how long it took for the student to complete the assignment, or the time differences between attempts. Also, there is no provision for analyzing all of the student data to find patterns, unless the instructor is willing to download assignment or discussion forum data details for additional analysis outside of the LMS. The reports and the reporting are both limited and limiting. Instructors can at best do a cursory analysis of where the problems are in the course and with the students. The data, though, could be collected and made available to the instructor by the LMS. Doing so would allow the instructor to perform a more in-depth analysis and potentially enrich the online experience for the student if there were tools available to more easily access the data and generate reports that are helpful to both the student and the instructor.

In the architecture described in this paper, several ideas and motivations were borrowed from Google Analytics (GA). GA is a free facility provided by Google to assist web developers and administrators in analyzing and understanding website traffic patterns. A partial list of the functionality included in GA include page view statistics, visitor path analysis, campaign conversion, goal tracking, time on page, location of visitor, and exit properties. The interested reader can learn more about GA at EPM1 (2010) and EPM2 (2010). GA does an excellent job presenting a large amount of data in an accessible way through a Business Intelligence (BI) based dashboard. This dashboard displays traffic information in a variety of different graphical ways: trend lines, bar charts, pie chart, maps, and so forth. A series of GA screen shots may be found at GA0 (2012), GA1 (2012), GA2 (2012), and GA3 (2012). GA allows users to investigate different aspects of web traffic patterns; each part of the dashboard is clickable, which allows users to drill-down for a series of more detail reports.

GA allows a website administrator to set up campaigns with conversion goals. These goals center on various performance measures for the website such as increased sales of a certain product, increased traffic coming for certain sources or resulting from certain keyword searches, increased time on page, and so forth. Once the goal is set, GA tracks the site's performance towards that goal. These goals become a metric of performance through which success of the website site may be measured. EPM2 (2010) provides a brief overview of this goal functionality contained in GA

We incorporate many of these ideas from GA into our architecture. We re-focus the ideas taken from GA towards a higher education student population. In the architecture highlighted in this paper, special care is spent on detailed page-to-page activity tracking, robust reporting tools, and analysis of the alignment of the learner activity and the learning goals set by the instructor.

At the heart of the architecture is the collection of detailed activity information of how the learner is doing within the LMS and how the learner is interacting with the learning content. It should be noted that the collection of the detailed activity data does in no way represent a compromise of either the LMS's or the browser's security protocols. All of the information obtained is readily available to any web based application. Most applications simply choose to ignore this source of information. The coarsest level of data retained in support of the current architecture is page level information such as which learning content is view by which student at what time for how long and from what physical location. The data is very similar to that retained by most current LMS systems as well as by GA.

At the core of the raw data generation subsystem is the event-based programming environment supported by virtually all modern web browsers. At its core, this environment allows web applications to detect and react to users as they interact with the system. For example, when a user types something into an input box on a web form, each keystroke is signaled to the browser as an event. Many applications will capture this event for helpful purposes, such as automated input completion, error checking, or interactive form generation. There are many events recorded by most browsers. A somewhat nontechnical explanation of event programming is available at Bergen (2010).

For the purposes of the architecture outlined in this paper, an extensive list of events are captured and retained. All student keyboard and mouse activity is retained. The events are associated with an element within a given piece of learning content. A careful tracking of the hyperlinks that were clicked on the page is maintained. Also retained is a characterization of the student's screen. The architecture records what portion of the learning content is viewable on the user's screen at a given time, how long did that portion stay viewable, and did the learner view the bottom of the page.

This raw data generation scheme is a general framework. Implementation details will vary based on the learning environment under consideration. Two partial examples of the potential of this framework have been implemented within the Aspen LMS. The examples were developed using data from business courses. The examples are based on approximately 200,000 transactional level data generated by approximately 400 students over a period of one month. A visual inspection of even a small subset of the data quickly overwhelms the viewer. More importantly, without a performance metric to compare the data against, deriving meaning conclusions from the data is difficult. We therefore next discuss schemes of how to generate these metrics and one set of possible analysis tools that can be employed.

DATA AND METHODOLOGY

There is typically an expected order in which students and instructors alike expect courses to be taught and hence how the student is supposed to learn. Instructors place learning activities and learning assessments into a sequence that the instructor believes will enhance learning. Instructors set specific deadlines for learning activities and assessments, and the student is assessed on the activities based on their performance at that specific time even though something later in the class may provide the student an opportunity to understand the content better. Some instructors accommodate the possibility that subsequent learning activities improve the students' understanding of previous learning activities by giving students the opportunity to complete a comprehensive exam or some other type of assessment that can improve the grade.

But there is an inherent assumption based in how courses are taught that many instructors believe that students must follow and complete course material in lock-step with the instructor's plan or the student will not do well in the class. LMSs make it possible to see if students are accessing learning materials on the specified date or just before the assignment deadline. IP addresses can indicate if the student is working alone or as part of a group. Details from the assignment, such as when it was opened by the student, when it was submitted, overall performance scores, and information from individual questions, can also indicate whether or not the assignment was rigorous enough, what material the class as a whole grasped well, and so forth.

Ascribing meaning to the raw data generated is impossible without a benchmark to measure the activity against. For example, suppose the activity data indicates that during a portion of the course student online activity has dramatically dipped. From a learning perspective, what is the meaning of this finding? It is very difficult to tell. The lack of online activity may have been caused by many reasons: the course is engaged in non-online content or activities, or the students are on a break. Student activity only has meaning when coupled with the expectations of the instructor. For example, the dip in activity is an area of concern only if the course is at a point where the instructor would expect the students to be heavily engaged with the online content.

Building these activity benchmarks are equivalent to the GA's conversion goals that we briefly described previously. Just as GA's goals become a metric of performance through which success on of the website may be measured, the activity benchmarks provide meaning to the learner activity. These benchmarks allow the instructor to communicate with the system what the instructor's expectation for the learner is. This is a key component because it allows measurements for the alignment of learner activity and instructor expectations. For example, when an instructor is working on an activity within the course, one would expect that the student traffic patterns on the supporting materials for that activity to dramatically increase. If this increase fails to occur, this would be a misalignment of the student activity with the instructor's expectations and would be an area of concern. This alignment can be measured, which provides a means of tracking the student's progress through the learning activity.

These benchmarks are particularly helpful to those instructors who regularly teach the same course. These benchmarks are maintained across terms and may be statistically correlated with other performance measures within the course. For activities that are repeated across terms, the students' activity patterns may be correlated with their final activity score recorded in the gradebook, and may be correlated with their final letter grade in the course. For large class sizes, statistical significance of these correlations tend to happen quickly. Also, the combination of the benchmark performance coupled with the student activity provides a knowledge base for a wide variety of data mining tools.

It should be noted that this benchmark-activity alignment does not guarantee any performance outcome for a specific student or for a specific course. The intention of this alignment is not to attempt to predict the future; it cannot tell an instructor what final grade a student will likely receive on an assignment or in the course. Rather this alignment provides statistical statements only, indicating the propensity of a student or class to perform in a certain manner based on facts. The alignment makes possible statistical statements similar to the following example: 75% of the students who started the assignment the night before the due date did not complete the assignment. Those that did complete the assignment spent an average of 15 hours completing it.

The benchmarks should be refined over time as the course curriculum evolves or as the instructor's expectations change. Misalignments between the benchmark and the student activity can drive a continuous improvement process for the curriculum. If disappointing alignment results are seen repeatedly across terms, this may be an indicator of a potential area for curriculum improvement.

For a concrete example, one of the authors of this paper teaches a very large (250 students) Introduction to Management Information Systems class using the Aspen LMS. The class has a series of large, complex, multi-week projects. The instructor regularly shares the alignment information in class with the students. Thus, for example, the instructor tells students before the due date how long students in previous terms took to complete the assignment and the failure rate on the assignment for students who try to do the assignment the night before the due date.

Repeatedly over the past four terms, the instructor has noticed a spike in student desirable activity immediately after the reports are shown in class. In addition, inter-term performance of the class activities continues to improve. More students appear to heed the instructor's warning to start the assignments early and to work consistently on them over time. These findings are heuristic only; other factors may be influencing these performance changes.

One of the perceived advantages of teaching in a small classroom environment is that the instructor has the opportunity to gain firsthand understanding of each student in the class. The instructor will get a sense of the student's perceived intelligence, ability to work in teams versus independently, commitment to the course, level of understanding of the material covered, and so forth. Then, based of these perceptions, instructors frequently subconsciously adjust the volume and detail of advice given to the students. In essence, the instructor builds an informal user rating metric for their students; the instructor's interaction with the students is in part determined by this metric. Over the duration of the course, this user metric can be adjusted as the instructor has further interaction with the student.

Building such a metric is feasible in a small classroom environment where an instructor may have direct, firsthand interaction with the students. Unfortunately, this does not scale to a large classroom environment where instructors typically have very little or no firsthand knowledge of their students. In such cases, customizing the delivery of the course material to meet the needs of an individual learner is a very difficult proposition.

The architecture outlined in this paper attempts to parallel the construction of the user rating metric through electronic means. Using the raw student activity data and the instructor driven benchmark, a user rating score is derived. This score is computed as an affine calculation based on a student's performance on a series of scorecard items. To be direct, these scorecard items are *not* elements from the course gradebook. Rather these items are measurable, binary performance activities that may be observed in the activity logs. These scorecard items should be easily extensible and will vary from environment to environment.

To help clarify this mechanism, we cite an example of such a user rating mechanism that has been implemented by one of the authors into a commercially available business simulation game known as Micromatic (OakTree, 2010). The goal of the user rating with respect to the simulation is to provide some measure of the student's understanding of the simulation and their ability to manage the complex interactions of the various functional areas of the simulated business environment.

The User Rating developed for the simulation contains over 50 scorecard items. One such item included in this User Rating metric is the student's consistency in researching current market conditions within the industry before recording their decisions for their simulated company. If a student frequently makes decisions "blind" to the market, it reflects poorly on their management style and their understanding of marketing. The item is set based on the percentage of times the student "blindly" makes their decision; if this percent is above a threshold level, the scorecard item is set to one.

As with the informal, intuitive metric, this formalized user rating is recalculated repeatedly over time. By retaining these values, a time series tracking of the student progress is possible. Once a reliable, formalized user rating has been developed for the content, a large number of uses become apparent. Since the user rating for the learners may be tracked overtime, at-risk students may be quickly identified if a dramatic negative change in their user rating occurs. These students can then be automatically referred to an advising function within the university or college. Based on the students User Rating score, intelligent agents may be built that will customize on a mass scale the delivery of content and advice based on the current student's needs.

If the scorecard items are categorized into sub-topic areas, a more detailed picture of student's strengths and weaknesses may be derived. This may be used to deliver more focused remediation to the student. At a macro, programmatic level, statistical profiles of all students in a program, department, or college may be derived as evidence of program outcomes and of potential weaknesses in those programs that should be addressed. This provides an automated way of generating program assessment reports and continuous improvement feedback loops that are required by many program or college accrediting organizations.

Business Intelligence (BI) is defined as "a system for analyzing collected data, with the purpose of providing a better view of an organization's operations to ultimately improve and enhance decision-making, agility and performance" (Stiffler, 2010). For faculty who teach and use LMSs, the data generated by the LMS can be extensive. For example, 400 student users of the Micromatic business simulation generate roughly 200,000 transactions in one month. The transaction level data is very detailed, consisting of a variety of information including IP addresses, login time, log out time, decisions made, and so forth. Because the data is transactional, it is associated with a particular student.

Thus, the instructor (or the system) has the ability to associate behaviors with traditional assessments (such as grades) and provide automated prompts to potentially improve student behaviors and hence learning outcomes. For example, if the data from the LMS shows that the average time it has taken students in previous terms to complete a learning activity is 15 hours and that 75% of the students who start the assignment the night before the due date fail to complete the assignment, then the BI we propose to be built into the LMS could automatically warn students who have not started the assignment by a particular time of the problem. The data in the LMS can also provide both the instructor and the students with a global view of the "average student" as well as identify outliers (such as students who have never logged in to the system or students who have already looked at most of the course materials).

RESULTS

We present early stage results from two partial implementations of the proposed student activity monitoring architecture. With both implementations, all client coding is done with code that is native to most commercially available browsers; no plug-ins or other third party software is needed on the student's computers. As mentioned previously, no modification to any security related browser settings is required. Unless the instructor chooses to inform the students of the presence of the tracking, the students are unaware of its presence.

Both implementations are incomplete in that they only employ portions of the above architecture. Future plans for both systems center around analyzing the performance of the current implementation of the student activity subsystems and then modifying and increasing the robustness of the architecture's implementation.

The learning environment for both implementations is business education. We believe that the technological architecture may be extensible to other disciplines. Producing similar systems for other disciplines is left as an area of future research. The first example implementation of our proposed architecture involves a LMS. In particular, one of the authors has implemented parts of the architecture in a LMS know as Aspen (Kaliski, 2010). For the Aspen example, the results shown in this paper originate from a large (250 student) Introduction to Management Information Systems (MIS) class at a mid-sized public university. The author has taught this course for over 15 years; for the past 4 years the author has taught this course using the large section format. The student activity monitoring has been in place for the most recent 4 semesters. The course is a project-based course only; there are no objective exams in the course. Rather the students work either individually or in teams on a series of large, complex, multi-week MIS-related projects. The vast majority of learning content is delivered through the Aspen LMS.

At the beginning of each semester, the author informs the students that the activity monitoring is in place. In this discussion, the students are told what type of activity is tracked and are shown sample reports from previous semesters. The typical initial student reaction to the monitoring is a feeling of unease; the students express concern that this activity monitoring is an invasion of their privacy. At weekly intervals during the student's work on the various projects, the author will show in class course-level activity reports for the current project. The students are reminded that their individual activity can be charted as well and that the author is willing to do so for the student during office hours.

The second example implementation of our proposed architecture involves a business simulation game. In particular, parts of the student activity monitoring architecture have also been implemented into the latest version release of Micromatic (OakTree, 2010). Micromatic is a commercially available business simulation that is in use at many colleges of business internationally. The typical courses that employ the simulation are Business Policy and Strategy and Principals of Management.

Micromatic offers a wide series of activity monitoring to the instructors. MM0 (2012), MM1 (2012), MM2 (2012) show several examples of screen shots from these reports; the examples are "live" in that they were taken from simulations played by actual students enrolled in actual courses. All identifying markers for the students have been removed for privacy reasons. These reports are intended to provide the instructors with insights of student participation in the simulation. The reports go well beyond monitoring the amount of time or effort the students are spending. The reports help the instructor to understand which parts of the simulation the students focus on, and where their strengths and weaknesses lie. Each report contains a course level section; the reports allow the instructor to drill down to an individual student.

The activity monitoring built into Micromatic also has a direct effect on the student's play of the simulation. The activity monitoring in Micromatic feeds into a business intelligence system called a Business Consultant. In particular, the Business Consultant (BC) is an intelligent agent built into the simulation that watches the student's play and then offers helpful tips or questions based on the student's play. The goal of the BC is to provide the type of guidance to a player that a well-informed instructor would provide, thereby reducing (but not eliminating) the need for human-based experts to run the simulation. At the instructor's discretion, the BC agent is available to the students at all times. Table 1 shows some examples of the kinds of advice that is automatically generated by Micromatic's BC.

Table 1: Examples of "Business Consultant" Advice from Micromatic

I am your business consultant and I am here to help you make your managerial decisions. As you proceed through the simulation, watch this area for the comments I will leave you. My comments will relate to the quarter just processed and the current quarter.
As the exercise progresses, I will withdraw my advice as the game progresses. After 8 quarters, I will withdraw from the game.
You are stocking out on some of your sales. This upsets both your customers and your salesforce. Should you raise your prices, produce more, or continue to lose your salesforce?
Your stock price is in serious shape. Should you sell off assets to get some cash to get back into the market? Your finished goods inventory appears to be unnecessarily building up. Are you overproducing? Are you overestimating your sales?
Your cash forecasting needs improvement. Be careful not to run out of cash! What happens if your sales don't meet forecast sales?
You are experiencing a significant loss from operations for the most recent two quarters. Are there operating expenses you can save to get you in a better financial position?

Table 1 shows the text of a conversation with a student playing the Micromatic business simulation game. These sample statements are automatically generated based the student's activity, perceived understanding of business in the simulation, and current performance in the simulation.

The goal of the BC agent is to assist in the student's learning. The BC's advice must be as relevant and current as possible. With that in mind, the BC heavily references the student's User Rating (described previously) to deliver advice that is accessible to the student at their current level of development. The number, detail, and level of helpfulness of the hints generated by the BC are directly impacted by the student's User Rating. This makes the BC agent adaptive on a mass scale to the status of the participants.

CONCLUDING COMMENTS

This paper outlines a potential model and use for the transactional data available in learning management systems (LMSs). Effective use of LMS transactional data can potentially increase student retention, especially when the students are bombarded with large classes in their first few years of college level enrollment. Retaining students is concomitant with retaining a loyal customer — that is one less seat that the admissions department does not have to identify a transfer student for replacement, thus saving the college or university significant dollars.

We consider the architecture outlined in this paper as being at an early stage of development and it is only one possible way the activity monitoring could be accomplished. This architecture will likely evolve over time. The implementation of the architecture will also grow as product plans come to fruition. The goal of the architecture is to find and solve problems involving student learning sooner than has been possible before and on a much larger scale. Having access of this information modifies the behavior of both the instructor and the students. Furthermore, for LMS vendors that decide to implement robust activity monitoring subsystems into their products, this subsystem will be a strategic product differentiator.

But there is a potential downside. The use of the transactional data must be carefully examined. Using agents to inform students of certain behaviors may help them do better in the class, but is it appropriate to use the data in this way? Is there an infringement on the individual's personal freedoms? Does it matter if the "system" rather than instructor is monitoring the student at the transactional level?

There are a daunting number of non-technical questions to be explored concerning this research. At this point, the societal and ethical impacts of this technology are unclear. Who should have access to this information? What are the privacy and security concerns? What are the implications to the students? What are the implications to advising? What are the implications to the instructor's evaluation and tenure? What are the implications to a program's or college's accreditation efforts? These are some important questions to be answered by future research.

REFERENCES

Barki, H., Titah, R. & Boffo, C. (2007). "Information system use -- related activity: An expanded behavioral conceptualization of individual-level information system use." *Information Systems Research*. Vol. 18. (2), p. 173-192.

Bergen, J. (2010). Event Programming Fundamentals. Accessed on December 16, 2010 at: http://csis.pace.edu/~bergin/Java/eventfundamentals.html.

Cavus, N. (2011) "The application of a multi-attribute decision-making algorithm to learning management systems evaluation." *British Journal of Educational Technology,* Vol. 42 (1, January), p.19-30

EPM1 (2010). The Basics of Google Analytics. Accessed on December 9, 2010 at: http://www.epowermark.com/basics-google-analytics-wp.php

EPM2 (2010). Setting Up Goals in Google Analytics. Accessed on December 9, 2010 at: http://www.epowermark.com/2008/03/setting-up-goals-in-google-analytics.html.

Fontenla, J., Caeiro, M., and Llamas, M. (2010). "A SOA architecture to improve the tailorability and extensibility of e-learning systems." *Latin America Transactions, IEEE,* Vol. 8 (2, April), p. 135-140.

GA0 (2012). Accessed February 5, 2012 at: http://aspen.mnsu.edu/GA0.gif.

GA1 (2012). Accessed February 5, 2012 at: http://aspen.mnsu.edu/GA1.gif.

GA2 (2012). Accessed February 5, 2012 at: http://aspen.mnsu.edu/GA2.gif.

GA3 (2012). Accessed February 5, 2012 at: http://aspen.mnsu.edu/GA3.gif.

Goodhue, D. L. & Thompson, R. L. (1995). "Task-technology fit and individual performance." *MIS Quarterly*, Vol 19 (2), p. 213-236.

Kaliski, J. (2010). Aspen,: An online classroom tool. Accessed on December 9, 2010 at http://aspen.mnsu.edu.

McGill, T. J. & Hobbs, V. J. (2008). "How students and instructors using a virtual learning environment perceive the fit between technology and task." *Journal of Computer Assisted Learning*, Vol. 24, p. 191-202.

McGill, T. J. & Klobas, J. E. (2009). "A task-technology fit view of learning management system impact." *Computers & Education*, Vol. 52. P. 496-508.

MM0 (2012). Accessed February 5, 2012 at: http://aspen.mnsu.edu/MM0.gif.

MM1 (2012). Accessed February 5, 2012 at: http://aspen.mnsu.edu/MM1.gif.

MM2 (2012). Accessed February 5, 2012 at: http://aspen.mnsu.edu/MM2.gif.

Naidu, S. (2006). "E-learning: a guidebook of principles." In *Procedures and practices* (2nd revised ed.). New Delhi, India: Commonwealth Educational Media Center for Asia (CEMCA).

OakTree Simulations (2010). Micromatic Business Simulation._Accessed on December 9, 2010 at: http://oaktreesim.com/Business-Simulation-Game-Instructors.html.

Pagani, M. (2006). "Determinants of adoption of high speed data services in the business market: evidence for a combined technology acceptance model with task technology fit model." *Information & Management*, Vol. 43, p 847-860.

Radford, A. and Weko, T. (2011) Learning at a Distance: Undergraduate Enrollment in Distance Education Courses and Degree Programs. Accessed on February 10, 2011 at http://nces.ed.gov/pubs2012/2012154.pdf.

Revere, L. and Kovach, J. (2011) "Online Technologies for Engaged Learners: a Meaningful Synthesis for Educators," *Quarterly Review of Distance Education*, Vol. 12 (2), p 113-124.

Staples, D. S. & Seddon, P. (2004), "Testing the technology-to-performance chain model." Journal of Organizational and End User Computing, 16, 4, 17-36.

Stiffler, J. Business Intelligence: Critical Insight for Private Equity. *Financial Executive*, v. 26 no. 8 (October 2010) p. 64-5.

Van Nijlen, D. and Janssen, R. (2011) "Measuring mastery across grades: An application to spelling ability." *Applied Measurement in Education,* Vol. 24 (4, October), p 367-387.

Yu, T. and Yu, T., (2010) "Modelling the factors that affect individuals' utilisation of online learning systems: An empirical study combining the task technology fit model with the theory of planned behaviour." *British Journal of Educational Technology*. Vol. 41 (6, November) p1003-1017.

BIOGRAPHY

John A. Kaliski is an Associate Professor of Management at Minnesota State University Mankato. His research appears in journals such as *Journal of Business Cases and Applications* and *Decision Sciences Journal of Innovative Education*. He can be reached at Department of Management, Minnesota State University Mankato, 150 Morris Hall, Mankato, Minnesota, 56001, john.kaliski@mnsu.edu.

Queen E. Booker is an Associate Professor of Management at Minnesota State University Mankato. Her research appears in journals such as *Journal of Information Assurance and Security*, *Journal of College Teaching and Learning*, and *International Journal of Mobile Communications*. She can be reached at Department of Management, Minnesota State University Mankato, 150 Morris Hall, Mankato, Minnesota, 56001, queen.booker@mnsu.edu.

Paul L. Schumann is a Professor of Management at Minnesota State University Mankato. His research appears in journals such as *Human Resource Management Review*, *Journal of Management Education*, *Decision Sciences Journal of Innovative Education*, and *Journal of Business Case Studies*. He can be reached at Department of Management, Minnesota State University Mankato, 150 Morris Hall, Mankato, Minnesota, 56001, paul.schumann@mnsu.edu.