Visualizing Online Course Participation Using Social Graphs

Sam Chenkin, Chris Grabosky, Neal Magnotta, John Misczak, and Jamie Thomson

Abstract—Online courses do not afford awareness of student participation as easily as traditional classroom-based courses. Though course management systems collect a plethora of data that could be used to create a better sense of how a course is progressing, few viable solutions have made it into commercial products. Social network visualization is one approach that may allow instructors to examine the learning of the group as a whole and any predefined or emergent subgroups as students interact on discussion boards. In this paper, we look at approaches that have been attempted in the past and then implement a visualization solution for the open-source course management system Sakai.

1 Introduction

Education and technology have long seemed a natural pairing. They both emphasize progress, discovery, and resolutions to well known, existing problems. The application of technology to education has been, to this point, fairly tame; instead of attempting to revolutionize the process, it has been restricted to making efficiency improvements to long-standing traditions. This approach, while tolerable for a traditional classroom where a crowd of students and one instructor gather for learning, is ill-equipped to handle the growing wave of online classrooms, built and supported by course management systems.

On top of individual learning, course management systems tend to embrace a new learning paradigm – that of group cognition. Kirschner, et al [1] discuss the traditional unidirectional nature of education, where instructors broadcast knowledge to students to accomplish specific learning outcomes at the individual level. They point out that as more collaborative pedagogical methods are employed, it becomes necessary to look beyond the individual to how students are sharing and complementing each other as they communicate and work in groups.

The data to support this analysis is largely already gathered by these systems. Every course management system records the date and time of posts and the users who post them. Most systems also record interactions that do not result in artifacts – logins, views, etc. Although this data is considered a helpful by-product of a course management system, it is rarely displayed in ways that can benefit the students in the class, the instructor, or both. This paper examines a potential solution to this issue by building an information visualization component into Sakai, an open-source course management system, which allows students and instructors to explore the social network structure that develops around them as they interact in the online learning environment.

2 The Problem

Students in online courses are given access to many of the same materials that they would use in a traditional classroom setting – lecture notes, reading assignments, and even software. This virtual setting also allows for more frequent use of outside references and sources of enrichment, which can be linked to easily and viewed at the student's leisure. In place of the immediate verbal and physical interaction that takes place in the classroom, students and instructors instead make use of tools such as discussion forums and both asynchronous and synchronous tools such as e-mail and chat rooms. The goal is to mirror the traditional classroom's amenities as closely as possible while also providing some new advantages and increased flexibility for learning.

In that regard, the instructor for a particular class is assumed to construct their grading scales and plans in manners similar to those used in traditional educational settings. While assignments and assessments are due at regular intervals to be examined and graded, contribution and interaction on the part of the students is to be weighed and considered when evaluating how proficient each individual became in learning and applying the material. These metrics are likened to the attendance and participation grades that are often used to ensure students are present and paying attention in physical lectures.

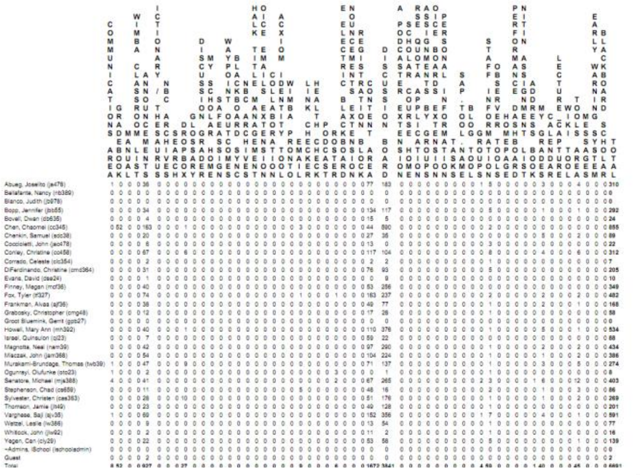
It is here that the dissimilarities between traditional and virtual classrooms begin to surface. While an instructor in a physical classroom can take roll at the beginning of class, a virtual classroom may not (and most likely, will not) be populated by all of the members of the class at the same time. This reality is actually an advantage of the virtual classroom – the fact that a student can plan learning time around the rest of his or her schedule. Even if an instructor could view how many students are connected to the classroom at any given moment for the entire term, it is rare that this would be a representative statistic, as students often download the course materials and assignments from the virtual classroom and spend a large amount of time completing work when disconnected from the classroom portal. Even discussion posts may be composed in a document on a student's computer and then copied into the online forum.

Additionally, weighing the quality of each instance of interaction by a student versus the quantity of contributions of the entire class is a significant problem in a virtual classroom scenario where the instructor is not an omnipresent facilitator. During a traditional lecture, an instructor can call on students at any time. In this way the instructor can directly draw participation out of each member of the class as well as encourage the introduction of a variety of viewpoints and opinions. The quantity of contributions to conversation is inherently restricted by the temporal limits of a face-to-face class session. With an online message board or text chat, however, each student could theoretically type out as many messages or posts as he or she wanted; simultaneously, some students who "sit quietly" in the class may not be noticed and prodded by the instructor. Setting a requirement on the number of posts each student must make is a tactic used by some instructors to ensure some level of engagement as well as provide an easy metric by which they can grade participation. However, this may lead to students posting for the sake of posting without adding much value to the discussion; in some instances, if posts veer off-topic or focus on only the most superficial issues, it could even hamstring meaningful discourse entirely.

In the instances where there are a large number of contributions from each student, the instructor must parse each one, looking for relevance to the topic and discussion at hand, originality in thought, and sound reasoning and analysis. This process can be extremely time consuming once the number of students in a class grows from just a handful to around two dozen. Additionally, depending on how the forum is structured, students may re-engage in discussions from past days and weeks, causing the instructor to have to search every area of the virtual classroom lest some significant part of a student's participation be overlooked. In many cases, the instructor will miss at least a few occurrences of interaction, which may or may not skew the opinion and eventual grade of one or more students.

Some course management systems recognize that this is a growing problem, and have implemented some measures to assist instructors in identifying key contributors to a class. Blackboard, a popular course management system used by many collegiate institutions, allows a user with the role of instructor for a class to view summary statistics of each student's participation.

Figure 1 shows the statistics table for each student, with the amount of participation in each area separated into columns labelled with the area name, such as discussion board or 'Wimba Classroom'. Although seemingly exhaustive in details provided, this function only slightly facilitates the instructor's task, as he or she still has to examine each part of the table, comparing not only students to students but also weighing the importance of some content areas against others.



	COMMUNICATIONS AREA	GRADEBOOK	MANUAL	WIMBA CLASSROOM LIST	ANNOUNCEMENTS	ICTIONARY / THESAURUS	NBC ARCHIVES	SEARCH	DROPBOX	GLOSSARY	DISCUSSION GRADER	MY SCHOLAR HOME	SAFEASSIGN	WAYPOINT OUTCOMES	BLACKBOARD SYNC	NBC ARCHIVES	HALK TITLE MANAGEMENT	OICE EMAILS EXTENSION	COLLABORATION	ACXIOM IDENTITY TOOL	WIMBA PRONTO	EMAIL	LINK CHECKER	HEALTH CHECK	PODCAST	ROSTER	E ELECTRIC BLACKBOARD	NOUNCEMENTS EXTENSION	REGISTRATION BLOCK	CONTENT AREA	DISCUSSION BOARD	A PODCASTER EXTENSION	SCHOLAR COURSE HOME	RSE QUOTA APPLICATION	ASSIGNMENTS BY GROUPS	OICE BOARDS EXTENSION	PERSONAL INFORMATION	TASKS	SAP.TOOL.NAME	BLOG TOOL	GROUPS	STAFF INFORMATION	OBSERVER TOOLS	HOMEPAGE	PERFORMANCE DASHBOARD	NITIN ASSIGNMENT LIST	ADDRESS BOOK	MY GRADES	CALENDAR	LECTURE_CAPTURE	WIKI TOOL	TOOLS AREA	MESSAGES	EARLY WARNING SYSTEM	BLACKBOARD SCHOLAR	TOTAL
Abueg, Joselito (ja476)	1	0	0	0	36	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	77	183	0	0	0	0	0	0	0	0	0	1	5	0	0	0	0	0	3	0	0	0	4	0	0	0	310
Bellafante, Nancy (nb389)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Blanco, Judith (jb978)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bopp, Jennifer (jbb55)	0	0	0	0	34	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	134	117	0	0	0	0	0	0	0	0	0	0	5	0	0	0	0	0	1	0	0	0	1	0	0	0	292
Bovell, Dwan (db635)	0	0	0	0	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	15	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	24
Chen, Chaomei (cc345)	0	52	0	0	163	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	3	0	0	0	0	0	0	0	44	590	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	855
Chenkin, Samuel (sdc38)	0	0	0	0	20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	27	35	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	5	0	0	0	2	0	0	0	89
Coccioletti, John (jac478)	0	0	0	0	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	13	0	0	0	0	0	0	0	0	0	0	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0	22
Conley, Christine (cc458)	0	0	0	0	67	0	0	0	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	117	104	0	0	0	0	0	0	0	0	0	0	8	0	0	0	0	0	4	0	0	0	6	0	0	0	312
Corrado, Celeste (clc354)	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	2	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	7
DiFerdinando, Christine (cmd364)	0	0	0	0	31	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	76	93	0	0	0	0	0	0	0	0	0	0	5	0	0	0	0	0	0	0	0	0	0	0	0	0	205
Evans, David (dse24)	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	10
Finney, Megan (mcf36)	0	0	0	0	40	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	53	256	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	349
Fox, Tyler (tf327)	0	0	0	0	74	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	163	237	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	2	0	0	0	2	0	0	0	482
Frankman, Alvas (ajf36)	0	0	0	0	38	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	49	77	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	2	0	0	0	1	0	0	0	168
Grabosky, Christopher (cmg48)	0	0	0	0	12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	17	28	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	58
Groot Bluemink, Gerrit (gpb27)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Howell, Mary Ann (mh392)	0	0	0	0	40	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	110	376	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	5	0	0	0	1	0	0	0	534
Israel, Quinsulon (qi23)	0	0	0	0	7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	59	22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	88
Magnotta, Neal (nam39)	0	0	0	0	42	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	97	290	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	2	0	0	0	2	0	0	0	434
Miaczak, John (jam368)	0	0	0	0	54	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	104	224	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	2	0	0	0	386
Murakami-Brundage, Thomas (twb39)	1	0	0	0	47	0	0	0	9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	71	137	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	3	0	0	0	5	0	0	0	274
Ogunreyi, Olufunke (oto23)	1	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	8
Senatore, Michael (mja388)	4	0	0	0	41	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0	67	265	0	0	0	0	0	0	0	0	0	2	3	0	0	0	0	1	6	0	0	0	12	0	0	0	403
Stephenson, Chad (cs659)	0	0	0	0	11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	5	0	0	0	0	0	0	0	48	16	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	1	0	0	0	3	0	0	0	86
Sylvester, Christen (cas363)	0	0	0	0	28	0	0	0	10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	51	176	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	2	0	0	0	269
Thomson, Jamie (jt49)	0	0	0	0	23	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	49	128	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	201
Varghese, Saji (sjv35)	1	0	0	0	69	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	152	356	0	0	0	0	0	0	0	0	0	1	7	0	0	0	0	0	4	0	0	0	1	0	0	0	591
Wetzel, Leslie (lw366)	0	0	0	0	9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	13	54	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	77
Whitlock, John (jlw92)	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	11	2	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	16
Yegen, Can (cly29)	0	0	0	0	22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	53	58	0	0	0	0	0	0	0	0	0	0	5	0	0	0	0	0	0	0	0	0	1	0	0	0	139
~Admins, iSchool (ischooladmin)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Guest	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2
Total	8	52	0	0	927	0	0	0	27	0	0	0	0	0	0	0	0	0	0	0	0	9	0	0	0	6	0	0	0	1672	3841	0	0	0	0	0	0	0	0	0	4	59	0	0	0	0	1	40	0	0	0	45	0	0	0	6691

Fig. 1. Blackboard participation summary statistics.

This tabular format is simply not flexible enough to provide the array of information instructors need for grading and course design. Instructors cannot drill down to relevant details about a particular student or content type. The instructor cannot filter out details deemed unimportant for the current class context. The table is calculated only at a single level of detail, a level not particularly useful for gaining a picture of the class as a whole or even determining the full breadth of a single student's contributions. The numerical nature of this presentation also makes it difficult to spot trends and exceptions within the course. A simple sort may illustrate which students have the most "points," but it fails to show more nuanced trends.

This format also makes it impossible to gain insight on the most important feature of any discussion-based classroom – student interactions. A summary table is not sensitive to the contexts of discussion topics or time. It cannot easily display which students interacted with which other students, or which time period was the most active. A system designer could make a separate table for each of these characteristics, but it would be difficult for the user to navigate and gain insight into the true workings of the class.

A different approach is clearly needed – one that allows instructors to "browse" the statistics for their class. Instructors need to be able to gain an overview of the class, filter out irrelevant details, and drill down to a particular student. Instructors also need to be able to view this information in a way that helps call out to trends and exceptions. These are all tasks well suited to the practice of information visualization.

3 BACKGROUND

Information visualization has grown dramatically as a discipline and technologically over the past few decades. Meanwhile, researchers have watched as online learning has proliferated and identified a number of important factors that relate to student success in this context. This paper focuses on the role of discussion forum participation as a major factor that can be measured, visualized, and explored to give students and instructors a better picture of what is going on in their courses.

3.1 The Necessity of Participation

It has been recognized for over a decade that "active online participation is a key factor in the success of student learning" in solely online courses [2]. Macfayden & Dawson [3] used regression analysis to look at what variables tracked by Blackboard Vista correlated with overall course performance as measured by the end grade for the class. They found that amount of total number of discussion messages posted, total number of mail messages sent, and total number of assessments completed accounted for 30% of variation in students' final grades in a particular course. Assessments in this case were optional quizzes that helped students study, and mail messages were generally used to ask the instructor questions. Macfayden & Dawson emphasize that systems using analytics to predict student performance must be customizable – as different material, teaching styles, and implementations of the LMS tools will require different key performance indicators.

3.2 Visualization of Participation

Since tools to monitor online course activity are not very robust, the most common method of measuring student participation is counting the number of posts students have contributed and other interactions they have had with the course management system. However, as previously mentioned, a tabular view of this data is not very usable for instructors or students. A few researchers have explored visualization solutions to this problem.

3.2.1 General Approaches

The most comprehensive course activity visualization project to date seems to be CourseVis [4]. This system uses six different types of visualizations to convey different facets of the course:

- *Discussion plot* – a 3-dimensional scatterplot displaying student creation of discussion threads over time. The number of responses to each thread was displayed using size and color of each sphere on the scatterplot. This visualization is interactive and engaging, but could be disorienting when the axes change as one switches angles.

- *Discussion graph* – a simple graph showing each student as a vertical line, with his or her thread creation total plotted as a star and number of replies to those threads plotted as a triangle plotted on the line. This makes it easy to see who creates the most threads and receives the most replies across the class. It also makes it easy to see who receives more responses than posts they make – which implies that they may post less often than others, but their posts generate conversation and therefore make a significant impact on the class. Another statistic that might be added to the discussion graph to reflect participation is the number of posts a student has read. An alternative method to visualize this data could be authorlines, which were used by Wesler, et al (2007) to identify social roles in Usenet discussion groups [5]. Authorlines only display an individual, whereas the discussion graph shows an entire group – but authorlines show activity over time on the X-axis, with quantity of posts plotted above and responses plotted below.

- *Cognitive matrix* – A grid displaying blocks of color representing students' performance on assessments for each concept defined by the instructor as part of the course. This visualization makes it easy to see which concepts an individual may be having difficulty with as well as what concepts the class as a whole is struggling with; this could facilitate adaptive teaching at the group and individual level, prompting an instructor to revise course materials midstream if they don't seem to be effective.

- *Cognitive graph* – A line graph showing the mean performance of the class on each topic. Less robust than the matrix, but still useful in highlighting what topics are causing students the most trouble.

- *Student accesses plot* – A scatterplot showing individual access events positioned over a histogram access totals. This shows clear trends in when students are logging into the course management system.

- *Student behavior graph* – A scatterplot of individual access and activities such as submitting assignments and quizzes.

Though an ambitious project with a number of effective visualizations, CourseVis only compares students' activities and does not consider the connections between them.

3.2.2 Social Network Analysis

Social network analysis provides a useful lens through which to observe student activity in online courses. A class can be visualized as a network of nodes (students and instructors) connected by edges (interactions, generally discussion board posts). In a communication context such as email or discussion board, replies to posts could be represented as directed edges [2].

Prior to Saltz, Hiltz & Turoff in 2004, most social network analysis conducted on online communication systems looked at the overall group without exploring an individual's social network [2]. They point out that the whole-network approach often does not scale well for larger networks, reintroducing the problem of information overload. Additionally comparing individuals' networks may lead to more insight into what the social network of a "successful" student looks like, thus helping instructors identify students who may need prompting – the digital equivalent of calling on a student who consistently does not raise his or her hand in class. The system in 2004 was Java-based and consisted of two components outside the course management system – one to gather the data and another to visualize. This is very similar to the architecture we ended up using, as will be discussed later in this paper. The resultant visualization showed an individual student's social graph, known as an ego network. The individual was represented as the largest node, with all other nodes sized as a reflection of how many messages they had exchanged. Thickness of the directional edges going to and from the individual also reflected the quantity of messages, in this case, discussion board replies. Students were represented by circular nodes, and instructors octagons.

In 2007, developing on their first project, Saltz, Hiltz, Turoff and Passerini reported on the development of iPET, an Integrated Participation Evaluation Tool that could be integrated into Blackboard's WebCT and other commercial systems [6]. This solution allowed users to define participation rules, view community activity, view participant activity, and receive automated participation reports. Community and participant activity could be viewed over time with the ability to filter and drill down in a variety of useful visualizations. This seemed like an extremely promising solution, but we were unable to ascertain its fate; if it was developed beyond 2007, the name must have been changed.

3.3 Participation Begets Participation

Though it may or may not exist anymore, the iPET solution demonstrated the power of information visualization as a motivator to increase student participation: "Students posted comments more often when they had access to iPET (between 23 and 57 percent more often). Assessment of qualitative feedback from instructors and students also shows common patterns of perceived quality increases." [6].

Quality of participation is an additional dimension that was ruled outside the scope of this project for the time but would be a fascinating area for further study. It is important to remember that correlation does not necessarily mean causation; just because a student posts frequently does not mean they will necessarily get a higher grade [3]. Still, the ability to visualize trends easily gives instructors greater insight into their students' progress, and provides students with a baseline for participation so they know approximately how much they should be doing to keep up with the rest of the class.

4 Approach

Research in this area has been conducted for nearly a decade as online learning proliferated, and yet solutions seem to have remained in the realm of research since they are not yet commonly found in commercial products such as Blackboard. While students at Drexel are most familiar with Blackboard, the closed architecture of the system has likely been a barrier to the development and widespread adoption of course activity visualization solutions.

Many other schools utilize an open-source course management system called Sakai that provides similar functionalities of Blackboard. Since the program is open-source, our team found it suitable for developing an information visualization add-on module. Sakai is bundled with a Context-aware Activity Notification System (CANS), an add-on which was developed to improve awareness of course activity and provoke higher rates of meaningful participation [7]. Thus far, CANS communicates with users via a daily email digest of class activity and a desktop notification widget. This project aims to extend CANS into the realm of social network analysis and visualization to help students and instructors measure their participation in online courses. Currently, the interaction information gathered by CANS is maintained as simple aggregate statistics and communication logs, similar to Blackboard's summary participation statistics.

To visualize this data, our team proposes the development of an add-on module that will generate visualizations of interaction data and integrate them within the Sakai interface, providing seamless transition from interacting with class participants in the system to viewing interaction visualizations. Using the Sakai Enhanced Awareness Module (SEAM), instructors will be able to identify emerging interaction groups and tailor projects or content directly towards specific groups in their classes. Also, students will be able to see their "ego network," that is, other students who often comment on their ideas in discussion forums.

5 Solution

The scope for this module is enormous. The information gathered by the CANS module can be evaluated across student, time, topic, interaction type, class, class module, class group, and many more. However, our research team elected to build a single proof of concept visualization – a social graph we call an interaction map. This map will allow instructors to see who interacts in the class and the strength of those connections. This is clearly only a tiny portion of the desired functionality, but does provide a foundation for future work. It is our hope the foundations of this implementation can be reused for more advanced functionality for this open-source project.

5.1 Application Components

SEAM will consist of three basic modules that will provide the visualization functionality: Interaction Warehouse, Interaction Visualization Generator and Sakai Integration Module. All of these modules as depicted in Figure 2 will work in unison to render social graphs.

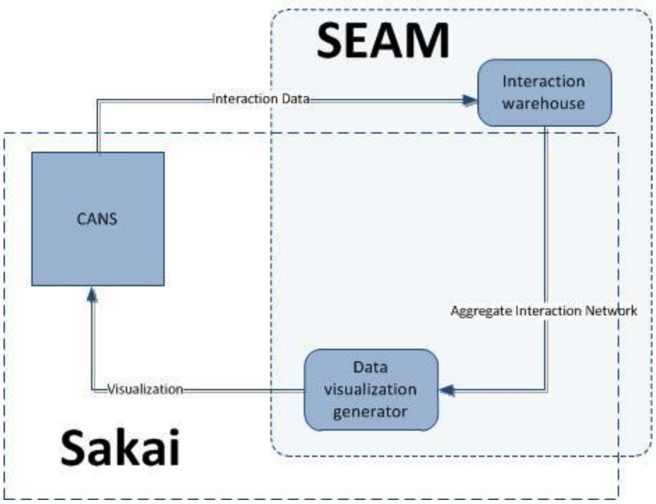


Fig. 2. SEAM architecture.

The Interaction Warehouse stores transformed interaction data extracted via CANS from a Sakai online course. This module interfaces with CANS to extract the discussion data, transforms the data into a format that more easily allows for visualization, stores this transformed information, and is used by the Interaction Visualization Generator to build the actual visualizations.

The Interaction Visualization Generator renders a pictorial image of a social graph based on data queried from the Interaction Warehouse. This module utilizes a Java backend to communicate with the Interaction Warehouse and Sakai, generates an HTML page with embedded JavaScript which is parsed by a JavaScript graphing library to generate scalable vector graphics (SVG) and ultimately show the visualization to the user.

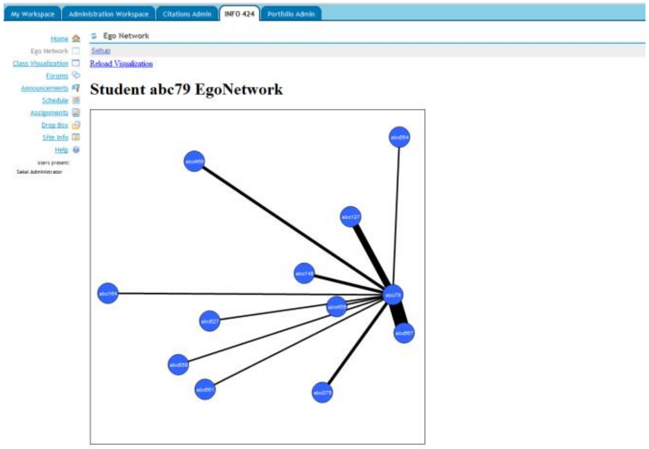
The Sakai Integration Module (SIM) connects SEAM into the Sakai user interface. The bulk of SEAM is separate from Sakai to avoid coupling between the systems such that SEAM could potentially be reused on any system that supports CANS.

The SIM module relies on minor modifications to the Sakai code base to allow users to access the newly created social graph visualization features that SEAM implements. This module enables an Instructor or Student to request social graphs.

5.2 Application Implementation

Users of SEAM are able to access the interface through the SIM module, which is hard-coded into the Sakai environment. For Students, the SIM user interface is one screen that takes in date parameters and loads a social graph visualization. For Instructors, the SIM user interface is again one screen, but takes in Class Participant(s) and type of visualization in addition to date parameters. The two main types of visualizations supported by SEAM are ego networks and class visualizations. Ego networks are a visualization based on one individual's interactions; one student is chosen to be the center point and then all other students which communicate with the selected student are displayed and connected. A class visualization extends this by having a node for every student in the class and draws every connection (representing communication) between them. The latter visualization can only be viewed by instructors.

Figure 3 depicts the visualization of a student's ego network. The user interface was implemented to hide the complex data manipulations that are required to transform the raw information in to an understandable visualization. Each blue circle, or node, is one member of the class that is labeled by their system username. Connecting the nodes are lines that represent the significance of interaction between the two students that is classified as "interaction weight". This weight correlates with the frequency and timeliness of interaction between the two students connected by the line. For example, if two students reply to each other's work and discussion posts frequently within a short period of time, the line will be extremely thick. Conversely, if two students only interact a few times throughout the class, the line will be extremely thin and almost unperceivable.



Student abc79 EgoNetwork

Fig. 3. Student ego network visualization.

6 THE DATA

As previously specified, our solution relies on data from the CANS notification system, which provides in-depth data on specific interactions within the Sakai online learning environment. This information is extracted from the CANS database and is loaded into a data warehouse (the Interaction Warehouse) to power the visualization engine. This process was originally devised by Sean Goggins of the iSchool at Drexel University (formerly at the University of Missouri-Columbia).

6.1 Data Transformation

This extraction, translating, and loading procedure is the core of SEAM. CANS is a transactional database that records a very large numbers of interactions every hour. CANS records every login, page view, document access, post view, reply, topic creation, etc. for every student in the system. To ensure a reasonable response time for each logging event the data is stored in a single table with a row for each interaction. However, this format is not well suited for visualizations. Related postings are not grouped (for example, both an original post and a reply are discrete rows in the table) and key characteristics such as the posting user and date are not easily accessible.

The ETL procedures used to load the Interaction Warehouse transforms this structure into something suitable for visualization. For our network visualizations this meant transforming an event from a discrete action into an interaction between users. Completing this process required first discarding any rows that did not represent an interaction between participants (logins, original content postings, document views, etc). Once this was completed each row was transformed to create a row in the 'Event Fact' table within the Interaction Warehouse.

Most of the information in this table is the same as the corresponding record in CANS. However, the transformation procedure created one critical new piece of information. Known as the 'Interaction Weight,' this new field provides a quantitative significance to the interaction between two individuals. This number is based on the assumption that the time elapsed between a discussion post and the next related event (either a reply or a read) indicates the strength of the interaction. This is based on the nature of CANS as a provider of real-time awareness notifications. This Interaction Weight provided the necessary information to build our visualization.

6.2 The Interaction Warehouse

The Interaction Warehouse, also originally created by Sean Goggins, was designed using a star schema to provide the maximum flexibility for analysis. At the heart of the data warehouse is the ‘Event Fact’ fact table which stores information about a specific event that occurs within Sakai. Each row of this table signifies the occurrence of an interaction between class participants, whether it is posting a new discussion post, a reply to a discussion post or simply reading a discussion post. Each row also includes the parent-post id for each reply and read action. The Interaction Warehouse does not store the post itself and is therefore not useful for content analysis.

References are included to important details about each interaction such as the institution, time, semester, and user person. This schema would provide supporting data for any number of visualizations beyond the two constructed as part of this implementation. For a visual representation, please review the database schema in Figure 4.

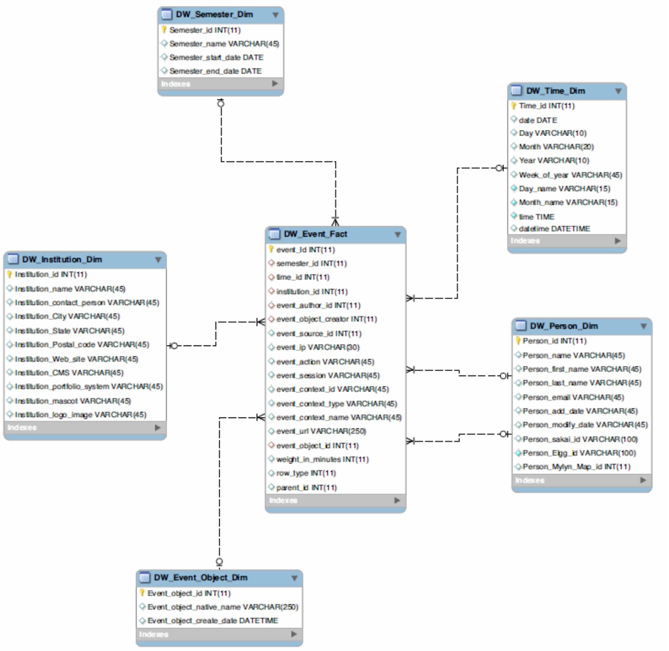

DW_Semester_Dim
Semester_id INT(11)
Semester_name VARCHAR(45)
Semester_start_date DATE
Semester_end_date DATE
Indexes
DW_Time_Dim
Time_id INT(11)
date DATE
Day VARCHAR(10)
Month VARCHAR(20)
Year VARCHAR(10)
Week_of_year VARCHAR(45)
Day_name VARCHAR(15)
Month_name VARCHAR(15)
time TIME
datetime DATETIME
Indexes
DW_Event_Fact
event_Id INT(11)
semester_id INT(11)
time_id INT(11)
institution_id INT(11)
event_author_id INT(11)
event_object_creator INT(11)
event_source_id INT(11)
event_ip VARCHAR(30)
event_action VARCHAR(45)
event_session VARCHAR(45)
event_context_id VARCHAR(45)
event_context_type VARCHAR(45)
event_context_name VARCHAR(45)
event_url VARCHAR(250)
event_object_id INT(11)
weight_in_minutes INT(11)
row_type INT(11)
parent_id INT(11)
Indexes
DW_Institution_Dim
Institution_id INT(11)
Institution_name VARCHAR(45)
Institution_contact_person VARCHAR(45)
Institution_City VARCHAR(45)
Institution_State VARCHAR(45)
Institution_Postal_code VARCHAR(45)
Institution_Web_site VARCHAR(45)
Institution_CMS VARCHAR(45)
Institution_portfolio_system VARCHAR(45)
Institution_mascot VARCHAR(45)
Institution_logo_image VARCHAR(45)
Indexes
DW_Person_Dim
Person_id INT(11)
Person_name VARCHAR(45)
Person_first_name VARCHAR(45)
Person_last_name VARCHAR(45)
Person_email VARCHAR(45)
Person_add_date VARCHAR(45)
Person_modify_date VARCHAR(45)
Person_sakai_id VARCHAR(100)
Person_Elgg_id VARCHAR(100)
Person_Mylyn_Map_id INT(11)
Indexes
DW_Event_Object_Dim
Event_object_id INT(11)
Event_object_native_name VARCHAR(250)
Event_object_create_date DATETIME
Indexes

Fig. 4. Star Schema.

The dimensions of the ‘Event Fact’ provide important context for each interaction. The person object stores the user ID, name, e-mail and other metadata about a specific participant in the online classroom. This information is used to integrate the visualization with Sakai by providing a link between users in that system with the data in the warehouse. This information is also used to display full names instead of user IDs in the generated visualizations. By providing the names of participants in visualizations, users can focus their cognitive processing on identifying trends and exploring the visualization.

The Time object stores detailed information about the occurrence of an interaction including the date, day, month, year, week of year, name of day, name of the the month and time down to milliseconds. This provides the infrastructure to enable users to filter interactions by many time characteristics in future iterations of the visualization program. This dimension in particular would be useful for more complex visualizations displaying interactions over time.

Several other information elements are stored for consistency, or for potential future use, but are not directly relevant to the visualization. Information about the institution includes standard information recorded by CANS including the name of the institution, its address, website, logo and Sakai implementation settings. Information about the semester includes the semester name, start date and end date of a specific semester. This enables multiple interactions from different institutions and semesters to exist for the same class, which may be useful for professors looking to compare and contrast the performance of different classes. Finally, the Event Object data element is used to track additional information about Event Fact Objects, such as their creation data and user-friendly names. While this information is not used in the visualization itself, it could be modified for additional features in future revisions of the application.

All of the previously described interaction data will be extracted from CANS’s proprietary database on a regularly scheduled basis. To begin with, the information will be loaded into the visualization system on a daily basis. In future revisions, the goal would be to enable real-time population of Event Facts into the visualization engine’s database.

In addition to the ETL and warehouse schema two stored procedures were created to facilitate pulling information out of the data warehouse. The data in the warehouse is not, on its own, sufficient to support the visualizations. The visualization generator expects a simple table containing a unique pairing of participants and the weight of the interactions between those two participants. To achieve this, the data in the warehouse needs to be de-duplicated and summed.

Two stored procedures, one for the class visualization and one for the ego network visualization, were created. The ego network visualization takes in a Sakai user ID and a start and end time to filter the desired interactions. The class network visualization takes in only a start and end time – this proof of concept relied on data from only a single class.

The functionality provided by these stored procedures could easily be integrated into the ETL procedures used to populate the data warehouse. However, this approach would have precluded using the underlying data for other, more complex visualizations. The stored procedures provide easy access to the necessary information for our two visualizations while maintaining flexibility for other implementations at a later date.

The ultimate goal of caching all of the previously described data is to provide a visualization of digital classroom interactions. To create this visualization, the visualizer application shall query the database to access information to create an array of information that can then be drawn. A tabular depiction of this array is displayed in Table 1.

Table 1. Interaction Data Array Example

Participant 1 ID	Participant 1 Label	Participant 2 ID	Participant 2 Label	Interaction Weight (min)
schenkin	Sam Chenkin	nmagnotta	Neal Magnotta	200
schenkin	Sam Chenkin	jmisczak	John Misczak	37
schenkin	Sam Chenkin	cgrabosky	Chris Grabosky	456

7 The Visualization

The supporting data transformations provided the necessary backend to support any number of visualizations. However, selecting visualizations to build into SEAM proved to be a challenge. The data structure supported any number of different parameters and interaction styles.

Before ultimately choosing a node or network visualization as what should be displayed, other types of visualizations were considered. For example, frequency histograms that showed communication frequencies were considered as one possible choice. Fractal diagrams that would follow a thread of communication were also considered. However the network diagram was ultimately chosen for one main reason: it represented visually how students naturally group together for communication, something otherwise hard to glean from statistics alone.

The histograms would show how much students communicated, something easy to see with the statistics that Sakai provides. Similarly, following one communication thread visually is interesting, but does not provide useful information to the instructor, especially since that can be seen in the Sakai message boards already. Instead, a node diagram would be useful to see the natural groups that form in a class – something difficult to perceive even for the students forming those groups.

Having chosen the visualization to generate, the development of the visualization generator could begin. To keep SEAM as portable as possible, one Java library was created. This would allow SEAM to be installed easily on any Sakai instance as well as easily ported to any system utilizing CANS. The SEAM package was augmented with several JSP pages which make up the Sakai Interaction Module. These pages are embedded in the Sakai interface and take in URL parameters and interact with the SEAM module, allowing users to directly communicate with SEAM through the Sakai interface.

In building SEAM, several important design decisions were made. From an end user perspective, the prevalence of portable computing platforms had an important influence on the design. While technologies like Adobe Flash, Microsoft Silverlight, and Oracle Java Applets would allow for the visualization to be rendered on the client machine, requiring a user to have the latest version of these technologies is cumbersome. Similarly, each of these technologies are not implemented fully on every mobile platform. As a result, the visualization generator would need to display dynamic visualizations in another way.

The chosen solution was Scalable Vector Graphics, or SVG. This allowed the visualization generator to issue very simple commands to draw simple objects like circles and lines. These simple objects would work together to create one larger visualization.

SVG is not implemented the same on every platform. However, this issue was resolved by using the the Raphael JavaScript library. This library handles complex multiple browser support by dynamically draw the SVG in JavaScript to ensure every platform, whether Windows, Linux, Mac, iPad, and more all see identical visualizations.

The visualization generator itself is not that complex. Using Java and JSP, it leverages the well-built interaction warehouse to get the data it needs. The data it retrieves can be seen in Table 1. This data is parsed into a format that allows for easy analysis in the Java code. The analysis process consists of several steps. First it determines the number of nodes. Next it determines the minimum, maximum, and average interaction weight calculated by the interaction warehouse. This weight is used to create a scale of line thickness to draw between nodes. The result is a dynamic line weight determined by how the students interact. Without this calculation the line weight would not vary sufficiently between connections, making it difficult to identify those that interacted the most.

The next step is to determine the X and Y coordinates of each node. This is one of the most complex parts because overlapping nodes would make it hard to discern the visualization. To solve this, nodes are placed in a circle format using calculus. Again, this is dynamically determined based on the number of nodes in the diagram.

To avoid further ambiguity as a result of line placement, the lines representing communication frequency between nodes are placed before the nodes themselves. This simplifies the edge detection of codes greatly by putting the lines below the nodes on the z-axis. Lastly, the nodes themselves are placed and labeled with the student login name.

The final visualization is the result of complex math and database work. While this speeds up with every refresh through caching at the database and application layer, it still can be somewhat time consuming. As a result, additional logic in the SIM helps to alleviate user confusion. The SIM uses AJAX to dynamically load the visualization in the background upon a user's request. Additionally, a loading progress bar is shown while the visualization generator is working. This feedback serves as a mechanism to keep the user abreast of what the system is doing and to prevent the user from getting upset, confused, or hitting the refresh button in the browser.

8 Conclusion

Course management systems and computer supported collaborative systems in general have made possible an entirely new way of learning. However, these tools still fail to provide a natural sense of awareness and intuitive feedback that is available in the physical classroom environment – it remains difficult to get a feel for overall class participation and group interactions.

Our SEAM module for Sakai is a first step in rectifying this failing in open-source course management systems. By taking advantage of the tremendous data-logging capabilities provided by CANS our system allows professors and students to gain insight into the social dynamics of class interaction. Moreover, the underlying data structure supporting the two visualizations created as part of this proof of concept could easily be extended to any number of non-network based visualizations. Future directions include improving visualization loading time, using an edge bundling algorithm to make the social graph easier to digest, providing additional visualizations to communicate other aspects of course participation, and working in other data captured by CANS. It would also be valuable to test the visualizations with students and instructors to see if they are easily understood. Still, SEAM provides a solid foundation for visual analysis of class participation in Sakai.

Acknowledgments

The authors wish to thank Professor Sean Goggins who prompted us to explore this topic and provided technical support and documentation for Sakai and CANS.

References

[1] P. Kirschner, J. Strijbos, K. Kreijns, & P.J. Beers. (2004) Designing Electronic Collaborative Learning Environments. *ETR&D, 52*(3). pp.47-66.

[2] J.S. Saltz, S.R. Hiltz, and M. Turoff. (2004) Student social graphs: Visualizing a student's online social network. *CSCW, 6*(3) pp.596-9. Chicago, Ill.

[3] L.P. Macfadyen and S. Dawson. (2009) Mining LMS data to develop an ‘‘early warning system” for educators: A proof of concept. *Computers & Education, 54*. pp.588-99.

[4] Mazza, R. and Dimitrova, V. (2007). CourseVis: A graphical student monitoring tool for supporting instructors in web-based distance courses. *International Journal of Human-Computer Studies, 65*. pp. 125-139.

[5] H.T. Wesler, E. Gleave, D. Fisher, and M. Smith. (2007) Visualizing the Signatures of Social Roles in Online Discussion Groups. *Journal of Social Structure, 8* (2). pp. 1-32.

[6] J.S. Saltz, S.R. Hiltz, M. Turoff, and K. Passerini. (2007) Increasing participation in distance learning courses. *IEEE Internet Computing, 11*(3). pp. 36-44.

[7] Laffey, J., Amelung, C. & Goggins, S. (2009). A Context Awareness System for Online Learning: Design Based Research. International Journal on E-Learning, 8(3), 313-330. Chesapeake, VA: AACE.