Semantic Visual Analytics for Today’s Programming Courses

I-Han Hsiao  
School of Computing, Informatics & Decision Systems Engineering, Arizona State University, Tempe, AZ, USA  
Sharon.Hsiao@asu.edu

Sessa Kumar Pandhalkudi Govindarajan  
School of Computing, Informatics & Decision Systems Engineering, Arizona State University, Tempe, AZ, USA  
spgovind@asu.edu

Yi-Ling Lin  
Department of Information Management  
The National Sun Yat-Sen University  
No. 70, Lienhai Rd., Kaohsiung 80424, Taiwan  
ylin@mis.nsysu.edu.tw

ABSTRACT
We designed and studied an innovative semantic visual learning analytics for orchestrating today’s programming classes. The visual analytics integrates sources of learning activities by their content semantics. It automatically processes paper-based exams by associating sets of concepts to the exam questions. Results indicated the automatic concept extraction from exams were promising and could be a potential technological solution to address a real world issue. We also discovered that indexing effectiveness was especially prevalent for complex content by covering more comprehensive semantics. Subjective evaluation revealed that the dynamic concept indexing provided teachers with immediate feedback on producing more balanced exams.

Categories and Subject Descriptors
Computing education → Computing education Programs → Computer science education→CS1

Keywords
Visual analytics, programming, auto grading, semantic analytics, intelligent authoring, dashboard, orchestration technology.

1. INTRODUCTION
Technology has become much more affordable and made abundant learning materials available online, which provides unprecedented opportunities to integrate various learning analytics in tracking diverse learning activities. However, paper-based exams are still one of the main assessment methods in today’s majority of classrooms. On one hand, cheating remains a big issue in the exams taken online. On the other hand, making meaningful exams is very time-consuming. Paper based exams simply do not convert to online tests overnight. There exists a technology gap between real classrooms and ideal technology-enabled ones. Such gap becomes more apparent especially in blended classrooms, where lectures and exams are delivered in traditional settings, but lecture slides, study guides, assignments and other educational resources are provided electronically through online portal or course management systems.

In the field of Computer Supported Collaborative Learning (CSCL), researchers describe it as a field in transition for classroom orchestration, which defines how a teacher manages multilayered activities in real time and in a multi-constraints context [1]. Orchestration emphasizes attention to the challenges of classroom use and adoption of research-based technologies [2]. Managing physical and cyber courses together can be demanding enough, yet adding more complex tools on top may make the complexity and time demands of technology even worse. Essentially “orchestration technology” may introduce new and unnecessary complications [3]. Our goal in this work is to study a less intrusive learning analytics approach that taps into blended classrooms with minimum technology introduction. We aim to assist teachers in continuing managing blended classrooms with their own traditional instructional and assessment methods, but connect with advanced learning analytics without modifying existing course delivery process. According to the broadening participation in Computer Science Education, we currently focus on Introduction to Programming classes, which are the cornerstone courses offered across programs & majors in almost all universities.

In doing so, we have identified the good, the bad and the ugly of traditional assessments in programming classrooms. Take paper-based programming exams for instance, they are easier for instructors to proctor and to manage the exam questions and to prevent from cheating, but they are suffering from (a) challenge for teachers to give personalized feedback on each individual test; (b) grading large scale amount of paper-based exams can be very time consuming and can create inconsistencies among graders; (c) paper-based exams are harder to keep persistent traces on detailed performances (i.e. no traces on how a student received partial credits; semantic level assessments and several other learning analytics are unavailable). When a student receives marks on a paper-based exam, it is difficult to get feedback in understanding if it is a single concept mistake, careless mistake or a long term misconception. Without the persistent traces of learning analytics, it is difficult for student to manage learning. With the existing pitfalls of instrumenting paper-based exams, it may result in students focusing solely on the score they earned on the returned exams, and missing several learning opportunities, such as identification of strength and weakness, characterizing the nature of their errors or any recurring patterns if any, appropriateness of their study strategies and preparation [4]. Thus, the grand objectives are not only to be able to support advanced learning analytics by providing detailed and in-depth semantic feedback through traditional assessment methods, but also to leverage using visualizations in visual analytics to promote reflection, self-monitoring, and support planning. In this work, we focus on presenting the design and evaluating the automatic indexing...
methods to associate semantics to paper-based exam questions and establish online persistent semantic visual learning analytics.

The rest of the paper is structured with literature review, research platform (semantic visual analytics system, EduAnalysis) and study methodology with underlying assumptions. Finally we present the evaluation results and discussed implications.

2. LITERATURE REVIEW

2.1 Orchestration & Learning Analytics

In the field of Computer Supported Collaborative Learning (CSCL), researchers describe it as a field in transition for classroom orchestration, which defines how a teacher manages multilayered activities in real time and in a multi-constraints context. It discusses how and what research-based technologies have been adopted and should be done in classrooms [1]. We have begun to see more tabletops, smart classrooms or interactive tools such as Classroom Response Systems (AKA: Clickers) etc. provide dynamic feedback and integrative students knowledge updates [5-8]. One of the biggest criticisms of introducing orchestration technology in class is that it might potentially add more complexity and time demands of technology and introduce new and unnecessary complications [3].

In the inaugural LAK proceeding, researchers describe a framework, TMTA [9], in discussing the importance involving three stakeholders in learning analytics: teaching expert, visual analytics expert and design-based research expert. The focus of learning analytics has been on the integration of computational and methodological support for teachers to properly design, deploy and assess learning activities. In addition, the focus is also to immerse students in rich, personalized and varied learning activities in information ecologies and data-rich classrooms [9]. One of the pioneer systems that align with TMTA framework is eLab (exploratory Learning Analytics Toolkit). It was designed to enable teachers to explore and correlate content usage, to help teachers reflect on their teaching according to their own interests [10]. ASSISTments [11] an integrative tutoring system includes assistance and assessment components for students and teachers. The system is built on a mantra - put the teacher in charge, not the computer, which creates flexibility to allow teachers to use the tool in organizing the classroom routines.

2.2 Visual Learning Analytics & Student Modeling

Visual learning analytics, essentially, extends the scope of information visualization by using computer-supported techniques to visualize learning information in amplifying human cognition. It goes beyond the “footprints” representation of summarizing and visualizing interactions or behaviors between students and learning content. Examples like network visualizations in semantic discourse analysis [12], dashboard visualizations to provide historical data in supporting awareness, teaching practices, explore and/or identify monitor status [13, 14]. The working group, ViSual Approaches to Learning Analytics (VISLA) workshop, in the Fifth International Conference of Learning Analytics and Knowledge [15], gathered a range of visual learning analytics cases. For instance, applying sentence compression technique in analyzing short answer questions in network visualizations; utilizing predictive modeling to visualize uncertainty of academic risks; innovative visualizations for visualizing semantics in discussion forums [16] etc. Studies showed that the majority of visual learning analytics discusses visual representations or the system’s usefulness while the core should be focused on real impact to improve learning or teaching [13]. However, from student modeling literature, we found several successful examples presented interactive visualizations in supporting students’ learning. Such approach is called Open Student Modeling (OSM). It is a group of approaches that makes traditionally hidden student models available to the learner for exploration and possible editing. Representations of the student models vary from displaying high-level summaries (such as skill meters) to complex concept maps or Bayesian networks. A spectrum of OSM benefits have been reported, such as increasing the learner’s awareness of their own developing knowledge and difficulties in the learning process; as well as student engagement, motivation, and knowledge reflection [17-20]. Several other examples of OSM interfaces reported promising results too. For instance, interacting with open learner modeling engages learners in negotiating with the system during the modeling process [21]. Progressor system integrates open learning models with social visualization that can dramatically increase student motivation to work with non-mandatory educational content [22] and encourage students to start working sooner. Chen et al. [23] investigated active open learner models in order to motivate learners to improve their academic performance. Both individual and group open learner models were studied and demonstrated the increase of reflection and helpful interactions among teammates. CourseVis provides graphical visualization to teachers and learners for multiple groups of users and helps instructors to identify problems early on, and to prevent some of the common problems in distance learning [24].

3. SEMANTIC VISUAL ANALYTICS SUITE: EduAnalysis

EduAnalysis is a semantic visual analytics suite specifically designed to extract semantics from physical learning environment and map onto a virtual setup to integrate blended learning activities. We developed a web application, including three main components, frontend analytics dashboard and web services to process physical data input (such as paper exam processing service, manual concept indexing service etc.), backend consists of an ontology parser, a concept mapper that maps sources of collected data to their corresponding concepts, and an analytics framework that exposes insights from data using APIs, and output via dashboard. EduAnalysis is designed to provide semantics representation of diverse learning activities between inside and outside classrooms, in order to provide more holistic and realistic learning analytics by harnessing the learning content semantics.

Teachers can upload an exam paper with a simple one click (interface omitted) and EduAnalysis will trigger exam parsing service to perform automatic concept indexing and immediately lead teachers to an overview (Figure 1). It guides teachers to navigate the entire exam concept distribution. Teachers can opt for further editing on exam questions or provide concept emphasis configuration. Figure 3 shows a view of the authoring phase. Left panel displays each question text, which enables dynamic editing and indexing to provide teachers instant feedback of the indexing performances. The correct answer and corresponding marks can also be collected and adjusted here, for future auto-grading services and partial credit assignment based on semantics services. Middle panel shows an interactive ontology authoring circle packing visualization. Teachers can select the bubble to zoom in and out to examine the concept coverage. Teachers can select/deselect concepts for the corresponding question by clicking on the bubble. They can also adjust the slider bar to configure the concept emphasis.
4. METHODOLOGY

This project aims to study an innovative semantic visual analytics that supports sources of learning activities and how teachers would perceive of using it in programming language courses. We hypothesized that intelligent automatic semantic indexer is an effective method to collect semantic information from course content. We call the instance of automatic exam concept indexing service as ExamParser, it inherits from generic Topic Facet Model [25], which consists of natural language parser and domain specific language Parser. It recognizes exam question patterns in a document and extracts content by indexing each question to corresponding concepts (a high level concept topic and sets of facets). A typical exam question pattern includes question_text{phrased as natural language, may or may not contain domain specifics}, codes{composed as fully or partially of an entire executable program}, answer_type{ranges from multiple choices, fill-in-the-blanks, short answers, code writing etc.}. For instance, a sample exam question is presented in Figure 2. It contains mainly natural language phrased question descriptions, a piece of partial executable java codes, and multiple choices answer type for this question. ExamParser will translate this question as a set of concepts {Q1: ForStatement, VariableInitialization, ConditionalStatement, LessOperator, IncrementOperator, MethodInvocation, Arithmetics}. However, do these concepts all weigh equally in this exam question? If we purely count the concept appearances, Question 1 consists of three AssignmentOperators and one ForStatement. Does it mean that ForStatement is less important than AssignmentOperator in this question? The answer is it depends! Therefore, we design a dynamic concept indexing authoring interface in the parser (Figure 3). It labels each parsed concept with the equivalent, default quantity weights, but the weights are adjustable according to teachers’ emphasis. In the case of using Question 1 in a CS1 midterm exam, the focus should be on ForStatement, ConditionalStatement concepts; using the same question in a CS1 final exam, every concept should weigh equally proportionally. In addition, providing dynamic concept weight authoring interfaces not only allows teachers to include or exclude additional or redundant concepts to exam questions, but also enables dynamic exam content editing and corresponding concept indexing (Figure 3). Embedding such dynamic authoring mechanism along with intelligent parsing can help raise teachers’ flexibility to configure and coordinate entire exam topical emphasis, at the same time, complement to algorithmic flaws, in case of any missing concepts.

Figure 1: Exam overview on topics and concepts distribution

Figure 2: Question1: a sample exam question.

The concept indexing method enables a scalable framework in two essential educational technology aspects: (1) systematically assign partial credits, which they are traditionally provided by teachers’ experiences or generic grading rubrics (such as credits to right path toward key concepts but erroneous implementation). By
associating each programming problem to weighted concept sets facilitates an organized fashion to quantitatively distribute partial credits in semantic level; and (2) harness different levels of learning analytics on both individual and group levels, including strong and weak concept clusters, misconceptions co-occurrences, conceptual progress over time etc. Currently we focus on aggregating various levels of semantics analytics as teachers’ overviews, however, these results can serve as detailed feedback to students as well. For instance, on a returned exam to student, student will no longer just receive the grade marks, but also a detailed feedback on what kinds of errors they made on the exams. The approach provides (1) individualized detail conceptual feedback, which normally can’t be done especially on large class size; (2) analytics to keep persistent traces on students’ conceptual growth; (3) opportunities for students to engage in reflection and self-monitor their own learning (foster metacognition development).

5. EVALUATION RESULTS
To evaluate the proposed semantic visual analytics for programming courses, we performed (1) content evaluation to examine the semantic approach impacts on exam question content; and (2) subjective evaluation by collecting interview feedback from teachers to understand user experiences and system usefulness. We collected 4 programming introductory courses exams, with a total 76 exam questions in the subject of Object-Oriented Programming. Each exam was populated in EduAnalysis; each question was automatically associated with a set of concepts through Topic Facet Model algorithm [25]. In order to verify the embedded indexing algorithm effectiveness, we had also collected the ground truth of corpus concept indexing by two expert judges, who both have more than 5 years teaching experiences in the subject domain. They manually examined the entire corpus by selecting concepts from a list of JAVA ontology keywords, which the inter-rater reliability was measured using Cohen’s Kappa= 0.386.

5.1 Concept Indexing Effects on Content
We consider the following aspects to assess analytics impacts on domain content: Content Complexity & Content Knowledge Structure.

5.1.1 Content Complexity:
According to CS1 course curriculum, depending on the exam foci topics, we split each exam by three levels of complexities, easy, moderate and complex. For instance, first exam usually covers topics from variables, primitive data types, arithmetic operations, Strings, conditions etc. These topics are usually considered relatively easy in the entire CS1 curriculum. However, in order to assess students’ knowledge, an early CS1 test usually is devised with a mixture of difficulty levels questions. Thus, a question comprising of multiple topics was considered as a complex question in that exam. We have tabulated two interesting findings. Firstly, we found that human experts (experienced teachers) had no differences in indicating the mount of concepts among three complexity levels of questions. The results supported the point that teacher experts tended to point out key concepts, instead of all concepts. Secondly, ExamParser indexed significantly more concepts in complex questions than the other two categories (Table 1). This result was very encouraging. More complicated questions were usually the ones that students made mistakes, which suggested more attention was demanded. However, teachers may not necessarily have the class time to go through details on every single question. Even if they did, such as mentioned key concepts of the tougher questions, the amount of feedback may not be sufficient. This is where the ExamParser can make a difference by supplying more detail feedback.

Table 1: Average # indexed concept by content complexity & Concept coverage by knowledge type

<table>
<thead>
<tr>
<th>Concept</th>
<th>Baseline</th>
<th>ExamParser</th>
</tr>
</thead>
<tbody>
<tr>
<td>Easy</td>
<td>1.943±0.121</td>
<td>4.400±0.541</td>
</tr>
<tr>
<td>Moderate</td>
<td>2.318±0.253</td>
<td>4.455±0.443</td>
</tr>
<tr>
<td>Complex</td>
<td>2.316±0.410</td>
<td>8.158±0.763</td>
</tr>
</tbody>
</table>

5.1.2 Knowledge Structure: Procedural vs. Declarative Knowledge
In order to address the cognitive aspects of our approach’s impact on learning content, we analyzed the indexed exam questions based on their knowledge types, procedural knowledge and declarative knowledge. A coarse-grained definition on procedural knowledge explains one knows how to do something; declarative knowledge approximately defines the knowledge about something. Thus, we identified the majority of the code writing questions were to test students’ procedural knowledge, and most of the multiple choices questions were designed to assess declarative knowledge. However, there were a few exception cases did not follow such classification. For instance, in one of the code-writing questions, students were asked to write Java code to “Instantiate an ArrayList that contains decimal numbers and assign it to an appropriate variable”. The question only involved syntactical tasks of the programming language, but excluded the application of syntax to perform further problem solving tasks. Thus, even though it was a code-writing question, it was classified as declarative question. Overall, we found 55% procedural questions and 45% declarative questions in the corpus. Based on the indexed concepts both by human experts and ExamParser, we found that, both types of questions had significant higher concepts indexed by ExamParser than the experts. This was consistent with 5.1.1, where ExamParser achieved higher coverage. What was interesting to note was that there were no significant differences between declarative and procedural knowledge types of questions, no matter who and how the questions were indexed. It showed the consistency among experts and the algorithm, which indicated ExamParser’s stability. Although, we anticipated procedural type questions would have been indexed more concepts due to knowing how to do may inherently involve some declarative knowledge components in addition to apply them to solve problems. However, we did not find such pattern observed. Possible explanations could be declarative types of questions (i.e. multiple choices) tend to include a range of meaningful distractor choices to prevent from simple memorization tasks. It also explained why there were larger variances in declarative type of questions compared to procedural ones.

5.2 Subjective Evaluation
We conducted a structured interview with two programming course instructors. Both are currently using Blackboard as course management platform and both give lectures and paper-based exams. One teaches medium size of Java courses (20–50 students averagely) and one teaches large size of courses (> 100 students...
We were mostly interested in finding how do instructors analyze students’ learning activities outside classrooms if any. Both instructors provide extra online learning materials (i.e. problem-solving or the sort) for students to perform self-assessments as non-mandatory resources for their courses. They encourage students to do more work through the selected online resources and provide partial credits for formal assessment as incentive.

We then allowed both instructors to explore EduAnalysis system and solicited feedback on the usefulness and potential threats of current implementation. They were instructed to test on the concept indexing procedure for different types of questions. They tried multiple choices and open-ended questions, and both agreed that the dynamic concept indexing provided them immediate feedback on producing more balanced exams. Both instructors reported that they found it convenient to perform one-click to upload and index exam concepts. They compared the experience with Blackboard evaluation feature, which requires them to configure each question one by one. Although the indexing authoring interface is available for every question, instructors considered it as flexible to assign designated emphasis to accommodate CS1/CS2 exams, or first/final exams. There were two major criticisms from both instructors: (1) they worried the auto-indexing precision may not be stable and result in them doing more configurations; and (2) the usability was not conclusive at the moment, at least not until they adopt the tool for their courses. However, both instructors expressed the current semantic visual analytics was reasonably useful, and both indicated extreme interests in using for their own classes in the future.

6. DISCUSSIONS AND CONCLUSIONS

In this work, we designed and studied a semantic visual learning analytics, EduAnalysis. It embedded intelligent concept indexing support to assist teachers in analyzing exam semantic composition in detail. We evaluated the effectiveness of the indexing services, the indexing effects on content and investigated instructors’ experiences and perceive usefulness on the system.

We found that the proposed approach shed some lights in extracting semantic information from paper-based exams. Such findings unlock the opportunities to (1) make persistent traces of learning analytics in semantic level; (2) provide more personalized feedback for students that is normally difficult to achieve or afford in a traditional (large) classroom. In addition, we found that EduAnalysis empowered teachers to configure exam topical emphasis and the results of indexed concepts appeared to maintain coherence within exam. It suggested that the proposed ExamParser approach could potentially make it possible to assign partial credits by concepts. We also discovered that the ExamParser indexing effect was especially prevalent for complex content. The results complemented the cases when teachers could not afford a lot of class time, but were forced to discuss key concepts on the tougher problems on a returned exam. Moreover, we also found the automatic indexing method was consistent with teacher experts in indexing both procedural and declarative types of questions. Subjective evaluation revealed that dynamic concept indexing provided teachers immediate feedback on producing more balanced exams; teachers expressed strong interests in using EduAnalysis for their own classes.

There were a few limitations under current study setup. For instance, current exams selection was a sample of CS1 four exams from our home university, which can be a biased sample. We should consider a wider range of exams and questions, such as textbook sample exams etc. There were a few evaluation limitations; such as teacher experts’ Cohen Kappa only indicated moderate agreement in our baseline group. As a result, the automatic ExamParser could potentially easily outperform experts. However, we argued that one of the reasons the inter-raters’ agreement was low could be due to the nature of indexing challenges and the setup for experts to pick out concepts from a long list of ontology. In addition, teachers were used to identifying key concepts even though they were instructed to be as comprehensive as they could when indexing. Given that the ground truth was not perfectly satisfying, we did not measure indexing error rate at this moment.

In the near future, we need to address the teachers’ concerns and to improve current design and evaluation. We plan to conduct field studies to collect larger scale of actual classroom usages and evaluate the semantic learning analytics impacts on students’ learning. In the mean time, we need to establish a stronger ground truth for future evaluation validation. In the long run, we would like to implement and examine the mechanism on assigning partial credits based semantics, experiment related technology to facilitate auto-grading, investigate different visualization impacts and finally, nevertheless, integrate other learning activities for more comprehensive analysis. More exhaustive evaluation is required.

7. REFERENCES


