Bridging Cyber and Physical Programming Classes: An Application of Semantic Visual Analytics for Programming Exams

by

Sesha Kumar Pandhalkudi Govindarajan

A Thesis Presented in Partial Fulfillment
of the Requirements for the Degree
Master of Science

Approved April 2016 by the
Graduate Supervisory Committee:

I-Han Hsiao, Chair
Brian Nelson
Erin Walker

ARIZONA STATE UNIVERSITY

May 2016

ABSTRACT

With the advent of Massive Open Online Courses (MOOCs) educators have the opportunity to collect data from students and use it to derive insightful information about the students. Specifically, for programming based courses the ability to identify the specific areas or topics that need more attention from the students can be of immense help. But the majority of traditional, non-virtual classes lack the ability to uncover such information that can serve as a feedback to the effectiveness of teaching. In majority of the schools paper exams and assignments provide the only form of assessment to measure the success of the students in achieving the course objectives. The overall grade obtained in paper exams and assignments need not present a complete picture of a student's strengths and weaknesses. In part, this can be addressed by incorporating research-based technology into the classrooms to obtain real-time updates on students' progress. But introducing technology to provide real-time, class-wide engagement involves a considerable investment both academically and financially. This prevents the adoption of such technology thereby preventing the ideal, technology-enabled classrooms. With increasing class sizes, it is becoming impossible for teachers to keep a persistent track of their students progress and to provide personalized feedback. What if we can we provide technology support without adding more burden to the existing pedagogical approach? How can we enable semantic enrichment of exams that can translate to students' understanding of the topics taught in the class? Can we provide feedback to students that goes beyond only numbers and reveal areas that need their focus. In this research I focus on bringing the capability of conducting insightful analysis to paper exams with a less intrusive learning analytics approach that taps into the generic classrooms

with minimum technology introduction. Specifically, the work focuses on automatic indexing of programming exam questions with ontological semantics. The thesis also focuses on designing and evaluating a novel semantic visual analytics suite for in-depth course monitoring. By visualizing the semantic information to illustrate the areas that need a student's focus and enable teachers to visualize class level progress, the system provides a richer feedback to both sides for improvement.

DEDICATION

To my loving mother - Mala G.

ACKNOWLEDGMENTS

I take this opportunity to extend my heartfelt gratitude to my advisor, Dr. Sharon Hsiao. It would not have been possible for me to reach this stage without her continued support and invaluable feedback during the course of the degree. Her immense trust in my capabilities and encouragement kept me motivated. I am grateful to Dr. Brian Nelson and Dr. Erin Walker for lending their invaluable time to be a part of my committee.

I am thankful to all the team members of the CSI Lab who helped me get all the necessary resources for my research and made the journey easier.

I am where I am today because of my mother - the bravest superhero in my life. Her faith in me and her support in every decision I made helped me move forward without doubting myself. All my achievements are motivated by you and dedicated to you. Thank you mom.

TABLE OF CONTENTS

CHAPTER Page

LIST OF TABLES

LIST OF FIGURES

Chapter 1

INTRODUCTION

Our education system from school to college is focused on teaching and obtaining feedback on how well the teaching has been received by the learners. Exams and assignments have been the major source of this feedback aimed at improving the teaching and the learning together. But this feedback is not complete and to a less extent considers the diverse set of subjects (students and teachers) in any classroom. Online exams can extract metrics from students and analyze their performance from many perspectives to provide personalized feedback based on the collected data. But cheating, safe access and proctoring remain a bigger issue in online exams. Paper-based (or paper) exams are still widely carried out in many schools. In majority of the schools paper exams and assignments provide the only form of assessment to measure the success of the students in achieving the course objectives. Though the paper exams are comparatively easier to proctor they suffer from:

- the challenge for teachers to give personalized feedback on each individual test, and
- the lack of (or harder) mechanisms to keep persistent, detailed traces on students' performances (i.e. no trace on why a student received partial credits; semantic level assessments are unavailable).

Given that the right feedback can help direct the focus of each individual student and

the teachers on the right areas that need their attention this gap has to be filled to improve the quality of assessment that in turn ensures improved quality in education.

1.1 Motivation

Online classes provide a unique opportunity to the educators to collect data on students and use them to drive insightful actions which were non-existent otherwise. But the traditional, non-virtual classes lack the ability to find deeper insights into the capabilities of students individually. In part this can be addressed by incorporating research-based technology into the classroom environment. Active learning as described in [9] and [10] introduce Tablet PCs into the classroom and engage students in real-time. This brings in real-time interaction in the classroom and provides students with concrete examples to reinforce their understanding. Yet this is not feasible in every classroom setup. There exists a technology gap between real classrooms and the ideal, technology-enabled ones. This gap becomes more apparent especially in blended classrooms, where lectures and exams are delivered in a traditional setting, but lecture slides, study guides, assignments and other educational resources are provided electronically through online portal or course management systems. Managing physical and cyber materials can be demanding enough; introduction of more complex tools on top may increase the complexity [1]. Assignments and exams serve as a reflection of students' reception to what is being taught in the class and as a feedback to teachers to align their course towards the course objectives. In traditional setting students get feedback only in the form of their overall marks. The final marks do not indicate whether the student's mistakes resulted from not understanding a particular

topic/concept, a careless mistake or a long term misconception. Also, the teachers in current scenario have to meet with every student if they want to give a detailed feedback, which is impossible with a bigger class size. Without the persistent traces of learning analytics, the existing pitfalls of instrumenting paper exams may result in students' focus directed solely toward the score they earned on the exams, and missing several learning opportunities, such as identification of their strength and weakness, characterizing the nature of their errors or recurring patterns if any, appropriateness of their study strategies and preparation [2].

Furthermore, making meaningful exams is very time-consuming. The teachers have to manually make sure they cover each concept specific to a particular exam to test the understanding of students on each of those concepts. Once the exams are evaluated the teachers do not have a holistic picture of how well the class has understood the concepts that were taught in the class. There is no easier way to evaluate how the students have responded to each concept tested. The loss of such information might prevent teachers and students from reaching the course objectives. If we can collected semantic information from paper exams and online assignments then we can provide detailed feedback in place of only an overall score. Providing every student with a detailed (visual) report on their academic standing might lead to a better understanding of their status in the class and a more focused approach toward achieving the course objectives. Thus, the focus of this research work is to bring the capability of conducting insightful analysis to paper-based exams with a less intrusive learning analytics approach that taps into blended classrooms with minimum technology introduction. This work specifically focuses on Java based courses. To that extent we designed and studied a semantic visual learning analytics suite to

provide intelligent support for today's majority modern classrooms without tampering teachers' current instruction pedagogy.

1.2 Research Questions

Exams are conducted based on the teachings in the class to measure the progress of the class toward achieving the course objectives. But, from the evaluation of a paper exam, it is difficult to get feedback on the student's understanding of the concepts tested in the exam. In order to bridge the gap in providing important feedback and understand the effectiveness of these analysis on a class, we aim to answer the following questions:

- how can we provide technology support without adding more burden to the existing pedagogical approach?
- how can we enable teachers to create an exam enriched with semantic information that can translate to students' understanding of the concepts taught in the class?
- how can we provide feedback to students that goes beyond just a cumulative sum of marks to answers and reveal what concepts a student really lacks in understanding?
- how can we provide intelligent assistance in pushing a class toward achieving the course objectives?

The rest of the thesis is structured in the following manner. Chapter 2 presents the literature review done as part of this research explaining the current state of

technology's part in classrooms. Following that in chapter 3 we discuss in detail the design rationales and functionalities of the EduAnalysis system. Chapter 4 discusses the evaluations performed on the system's automatic indexing component and presents the results. Chapter 5 presents the conclusion and future work of this research.

Chapter 2

LITERATURE REVIEW

This section covers the background literature and related research work. The need to investigate an intelligent system that supports semantic enrichment of programming exams, and analyze students' performances to provide feedback, is explored from three different topics of research. Firstly, I present an overview of the Computer Supported Collaborative Learning, the transition towards orchestration in technology-enabled classrooms and the use of learning analytics in providing insights in these classrooms. Secondly, I review the tools built around visual learning analytics and student modeling with a specific focus on open student modeling (OSM). Lastly, I review the research advancement towards automatic grading and providing automated student feedback with the use of technology at large scale.

2.1 Orchestration and Learning Analytics

The field of Computer Supported Collaborative Learning (CSCL) deals with creating a learner-centric environment with an emphasis on social interaction in the form collaboration between peers/students. This involves exploration and experimentation of introducing technology in the classes to aid day to day activities focused on the collaborative aspect of learning. The use of technology and collaboration is also aimed at improving the instructional scenarios in different areas of education thus enabling teachers to control the attention of their students. With the introduction of more

technology into the classrooms the need to ensure the effectiveness of collaboration between students and the effect of enhanced learning on individual students becomes increasingly important. But the pedagogical approach of CSCL does not fully address the combination of physical and virtual dynamics in a classroom thus leading to a transition to classroom orchestration, which defines how a teacher manages multi-layered activities in real time, with multiple tools, and in a multi-constrained context [9]. It empowers the teachers to drive the classroom activities. The literature in [23] discusses how and what research based technologies have been adopted and should be done in classrooms enabling orchestration. 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 student knowledge updates [19] [27] [20] [24]. As much power they bring to the classroom they also increase the components that a teacher has to manage additionally through different mediums. One of the biggest criticisms of introducing orchestration technology in a class is the possibility of potentially increasing the complexity and time demands of technology and introduce new and unnecessary complications [25]. Thus there is a need for providing tools to teachers that enable orchestration along with near real time insights into the effectiveness of the activities with the use of learning analytics.

In the inaugural LAK proceeding, researchers describe a framework, TMTA [28], 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. The information thus provided should help in the decision making process of the teachers to introduce

or restructure the program according to the current state of the class. In addition, the focus is also to immerse students in rich, personalized and varied learning activities in the information ecology and data rich classrooms [28]. 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 [12]. ASSISTments [15], 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 and Open 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 [7], dashboard visualizations to provide historical data in supporting awareness, teaching practices, explore and/or identify monitor status [29] [8]. The working group, VISual Approaches to Learning Analytics (VISLA) workshop, in the Fifth International Conference of Learning Analytics and Knowledge [11], 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 [2] 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 [29]. However, from student modeling literature, we found several successful examples presenting 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 [4] [5]. 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 [10]. Progressor system integrates open learning models with social visualization that can dramatically increase student motivation to work with non mandatory educational content [17] and encourage students to start working sooner. Chen et al. [6] 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 [21]. The information thus visually provided to the teachers can

bring to surface the otherwise latent signals on the effectiveness of the instructional approach carried out in the class. This can augment the details of the class from other collaborative technologies thus providing a deeper insight to aid the orchestration process.

2.3 Technology Support for Programming

The introduction of orchestration has led to more moving components to be addressed by the teachers thereby increasing the need to find methods to reduce mundane tasks carried out by teachers while ensuring better feedback in the system for both students and teachers. Paper exams still being the primary method to assess the learning of students in college curricula the information on a student's performance is not virtually available. Owing to the advantages in terms of ease of administering paper exams compared to online exams and the challenges in ensuring a secure and fool-proof online exams the paper exams are still preferred. But to enable teachers to focus on better orchestration there have been many different tools built to reduce the burden of grading on teachers. The research here has focused on the interpretation of the exams [1], plagiarism detection [17][6], to grading systems [14]. The system in [3] uses tablet PCs to grade paper based by converting the paper based exams to digital version that can then be loaded and corrected from within their system. This system ensures a faster turnaround time for the graded exams along with comments and feedback on the virtual copy of the exam. Other systems like [26] try to solve the grading problems at large scale considering the MOOCs that are rising in popularity. This system tries to reduce the time in grading short answers

by analyzing the answers of different students and clustering similar looking answers thus reducing the number of answers to be read and corrected to one representative answer from the clusters. One of the main disadvantage of the paper exams is that all the information of students' understanding is summarized by different scores for each question. This information could otherwise be important in helping the teachers obtain insights that can aid in effective orchestration and also help students work on the right areas that need focus. But providing feedback to each and every student in a paper exam is harder and time consuming. Thus developing systems that can provide proper feedback to students has been an active ongoing research. The system in [22] is specifically built to analyze answers for programming questions and provide feedback in terms of code corrections and a measure of incorrectness of a particular solution. Providing the same feedback would be a tedious task for evaluators and the feedback could be more broader without providing exact places of errors. The system thus helps the students learn their mistakes with a direct, personal feedback which is harder to obtain otherwise. Similar to [26] Overcode is a system built to cluster similar answers and provide accurate feedback to students on a large scale.

Majority of the above systems for student feedback require a major deviation from the traditional method of evaluation and consumes more time in terms of scanning the answer papers along with greater space requirements in computers to store the large images created by scanning. The systems above also mostly concentrate on online tests where data is abundantly available to tap into. These systems mostly miss out on paper-based assessments which form one of the major form of assessment in many schools and colleges. Also, these systems do not identify or utilize the semantic information present in the questions. The students still do not get feedback that

specifically points to the gaps in their understanding of the concepts taught and lack in directing their focus. The teachers on the other hand cannot identify how the class is progressing toward achieving the course objectives by looking only at the marks and the comments. If the semantic information from the questions can be mapped to their corresponding concepts then the answers provided by students can be mapped to the understanding of those concepts which can then be aggregated to provide a holistic view of a class's understanding of the curricula. Our system differs from the above systems primarily in utilizing this rich information to provide feedback to the students. In addition to providing comments from the evaluators we also provide visual indicators of the students' performances that can help the students understand their position in the class and be provided with a direction to work toward improving the areas that need their attention more.

Chapter 3

EDUANALYSIS: SEMANTIC VISUAL ANALYTICS SUITE

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. To this extent we developed a web application, consisting of three main components viz.

- 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 based 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 semantic 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. It is designed to support data inputs from varied resources to accommodate different mediums of evaluating students in enhanced learning environments. This particular version of the system is built to support Java beginner level programming courses and is tested on the exam papers from those classes. The following sections cover the architecture and the main components of the web application.

3.1 System Architecture

The system has been built as a web application. The Figure 1 depicts the architecture of EduAnalysis system with three main components viz. an ontology based parser, a concept mapper that uses natural language processing techniques to map questions to their corresponding concepts, and an analytics framework that exposes insights from data using APIs. The core part of the system is in the "domain & student model." MongoDB acts as the primary datastore for the application. Ontologies are separately stored in simple file formats that can be read and parsed to load according to the domain in study. Concept mapper algorithm takes as input the hierarchical data from the ontology and the question content and maps the concepts and stores the final mappings into MongoDB. The auto-grading algorithm updates the score automatically according to the evaluated answers. Student model is updated with every exams and evaluations. This historical trace of a student's progress helps in personalized prescription of learning materials matching the student's weaknesses.

The core system works on files containing pre-defined patterns for parsing the questions and answers in their exams. The question content and the answer content are required to have a recognizable pattern (e.g. Q1, a. etc) from a list of patterns the application is programmed to recognize. It does not depend on any specific form of input media till the input to the system can be provided with the necessary patterns to build upon. Here the *analytics web server* acts as an interface between the outside world and the EduAnalysis system. As in the figure, the exam documents and/or the Blackboard evaluations can be uploaded to the system through the analytics web

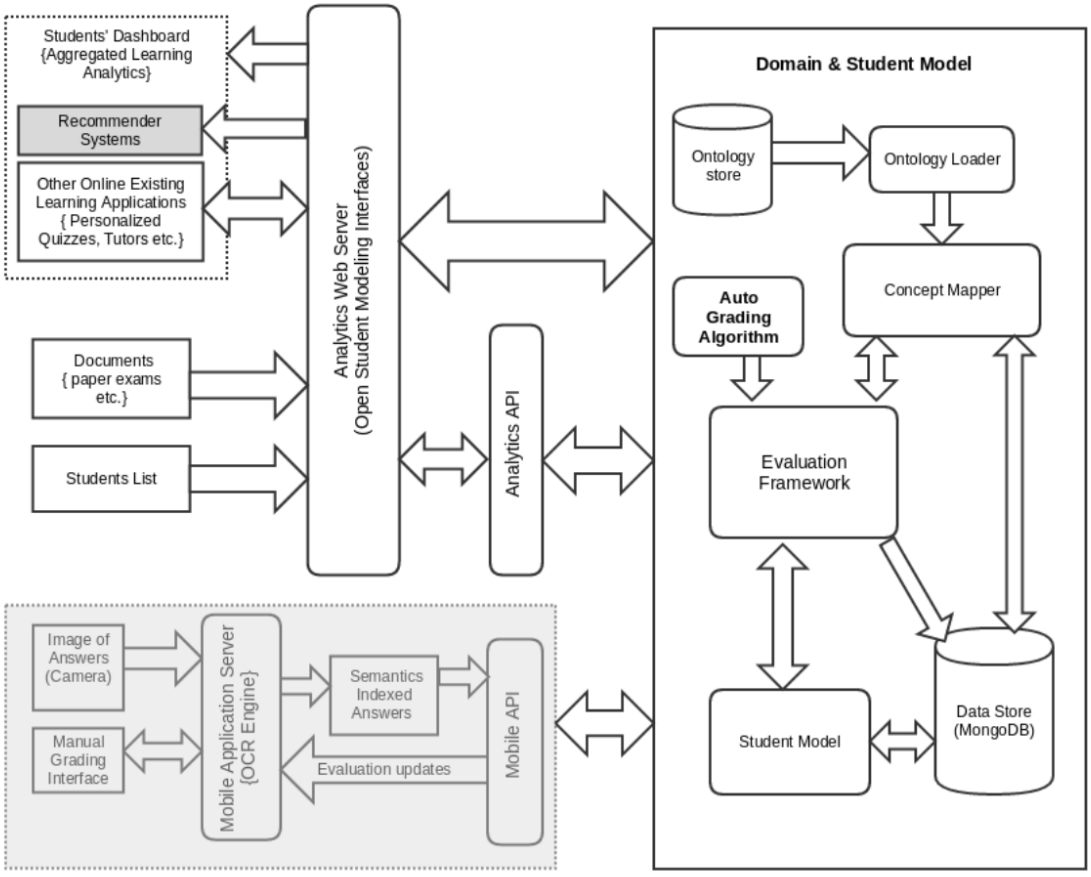

Figure 1. EduAnalysis system architecture.

server. The system then parses and stores the enriched question dataset for further use. The components shown in the dotted boxes are not part of the implementation. EduAnalysis is built in such a way that these components can be supported out of the box. The mobile component can be built to utilize grading on handheld devices and it can interact with EduAnalysis to get the concept mappings, store the evaluations and to retrieve analytics and feedback to students. The data collected on the students' understanding can be utilized to make recommendations of resources to read from to

strengthen their knowledge on the concepts that need their focus. Also we can inject data from Blackboard and other similar online components till we can parse the data to an acceptable format for parsing.

The system is built using MongoDB as (NoSQL) database, Python Flask as back-end web server and AngularJS for front-end. The application is deployed on an Amazon EC2 instance that can be accessed at [13].

3.2 System Components

The following subsections explain in detail the main components of the system.

3.2.1 Authoring Interface

The main purpose of the EduAnalysis application is to enable efficient and faster grading of the exams for teachers and simultaneously utilize the semantic information in the questions to evaluate the understanding of the students on topics of interest. This is achieved by providing an interface to the teachers to control the direction of the course based on each exam, with an overview of the class's performance through semesters and different courses and configure the semantics of each individual exams to test only the needed topics for understanding. Teachers (or administrators) author the flow of their courses through this interface. The system allows the teachers (or administrators) to construct hierarchies of a course setup. The teachers can create a semester, classes within a semester and different exams within each classes. The interface to create a semester is shown in Figure 2 wherein the teacher/administrator

has to enter the semester name and the year in which the semester falls. This is the first or top level in this system's hierarchy.

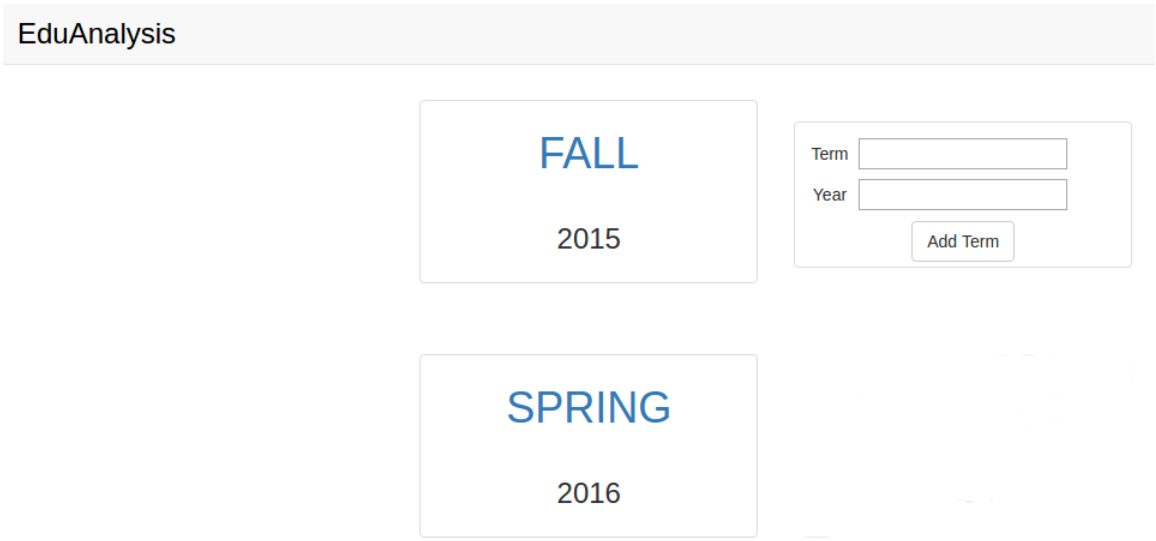

Figure 2. Add semesters to the system

Once a semester is created, any teacher can create a class within that semester by entering details as in Figure 3. The classes take inputs for the class code, the name of the staff who is in charge of the class and a file containing the list of students along with their id who have taken this particular class in this semester. Once a class is created the teacher can then open the class hierarchy and start adding exams with the minimal detail of the exam's name as in Figure 4. This hierarchy allows the system to be a repository of student information providing their progress across semesters and courses. Thus the system acts as a central store where all the information about

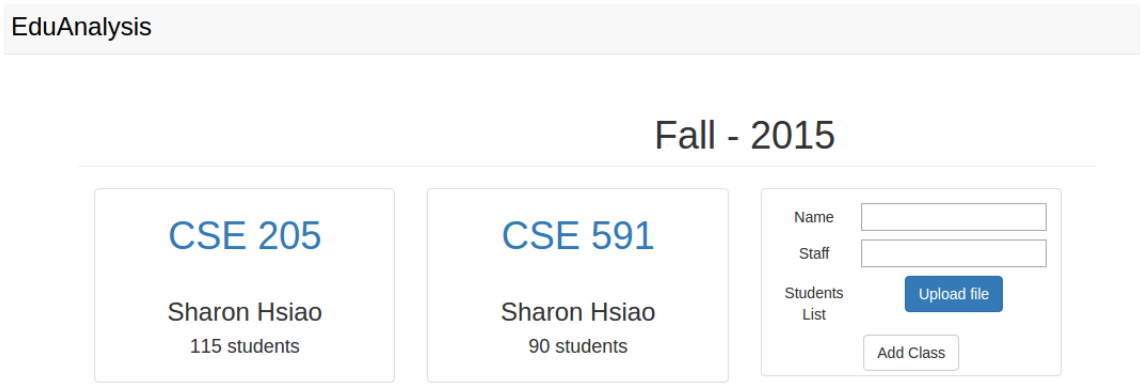

Figure 3. Add classes to the semesters

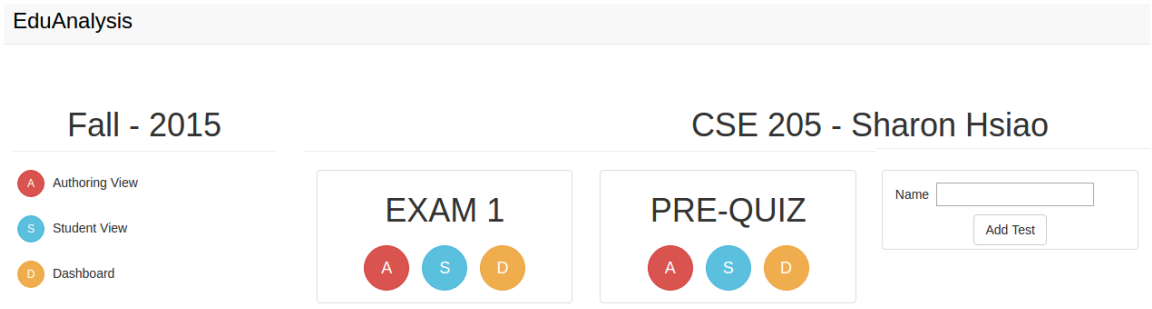

Figure 4. Add exams to classes

a course's evolution as well as a student's evolution can be obtained at any point in time. This will also be helpful in the orchestration of the class with real-time and historical track of students' information.

After setting up the structure for exams the teachers can upload the master exam paper in one of the doc, pdf or text file formats using a simple file upload interface. The exam paper should contain the questions along with the answers to those questions (for multiple choices) in a pre-defined pattern recognizable by the system. The encoding

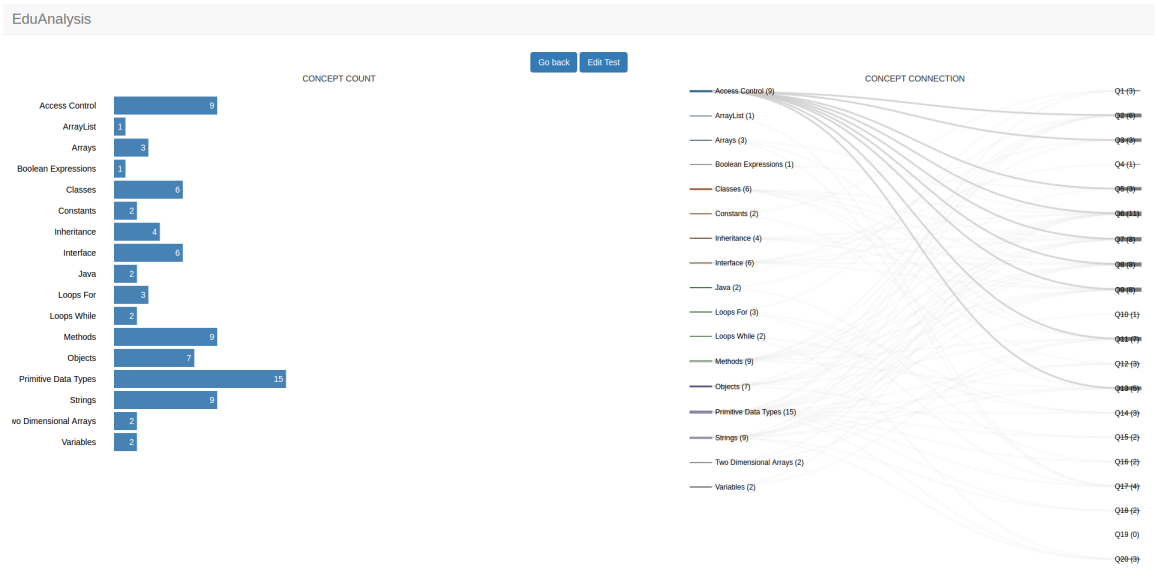

Figure 5. Overview of concept mapping for Exam1 questions.

of the exam paper to match a required format is important to parse and separate the questions from the answers. One of the simple format specification could be enforcing the addition of patterns like 'Q' and 'A' in front of questions and answers respectively. As soon as the paper is uploaded, EduAnalysis will trigger the exam parsing service to perform automatic concept indexing and immediately lead teachers to an overview as in Figure 5. It guides teachers to navigate the entire exam concept distribution.

The overview can be used to find different questions that share a common concept between them to know the variations in terms of questions and the concept distribution. The system that parses and automatically indexes the concepts for the questions will be discussed in section 1.2. The concepts are based on a specific ontology that can be built and plugged in according to the domain of the course. The assigned concepts

are provided a weight/emphasis based on each question's content. This elicits those concepts that are more importantly tested in a given question.

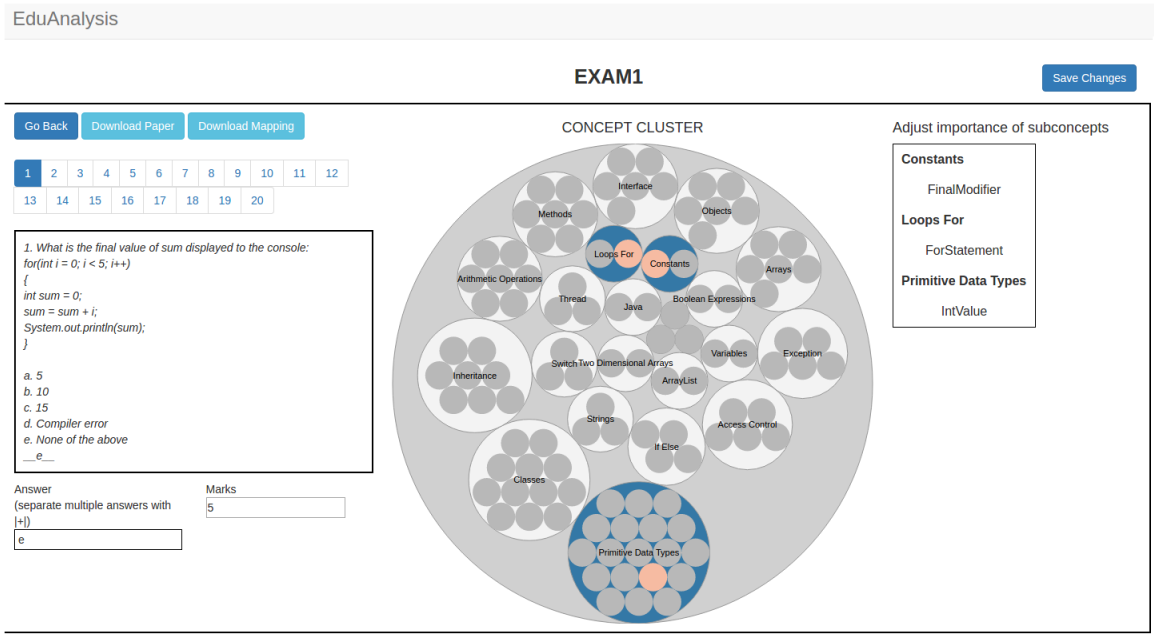

Figure 6. Authoring mode interface

Teachers can opt for further editing on exam questions or edit the concept emphasis configuration. Figure 6 shows a view of the authoring phase. Left panel displays the question text and provides a set of navigation buttons to move through all the questions in a given exam paper (Figure 7). This enables dynamic editing of each question and automatically updates indexing of concepts with question updates to provide teachers instant feedback on the indexing performances. The correct answer and corresponding marks can also be collected and adjusted here, for future auto grading services and also for assigning partial credits (for a partially correct answer)

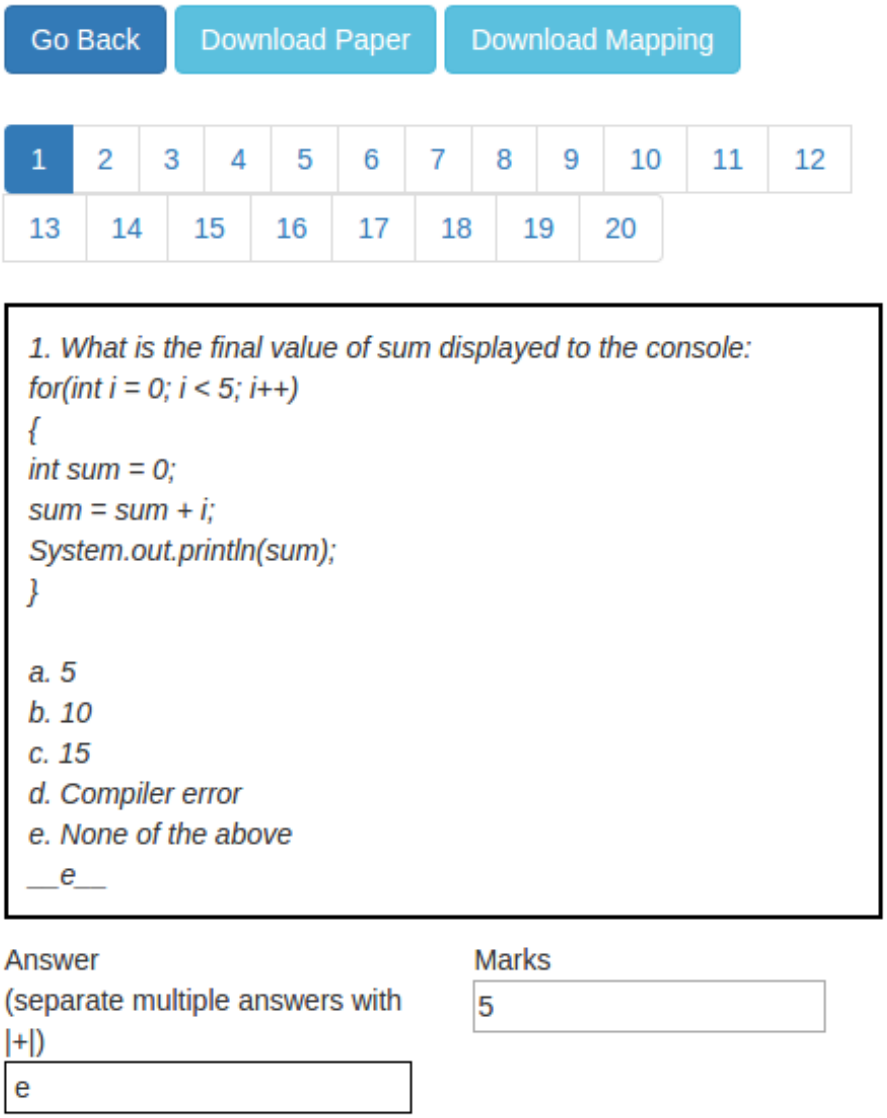

Figure 7. Authoring mode interface - question features

based on semantics services. Middle panel shows an interactive ontology authoring circle packing visualization. For each question the matched concepts are highlighted in the pack as in Figure 8. Each concept has sub-concepts that are directly represented in the questions. The right panel displays a summary of all the concepts that are

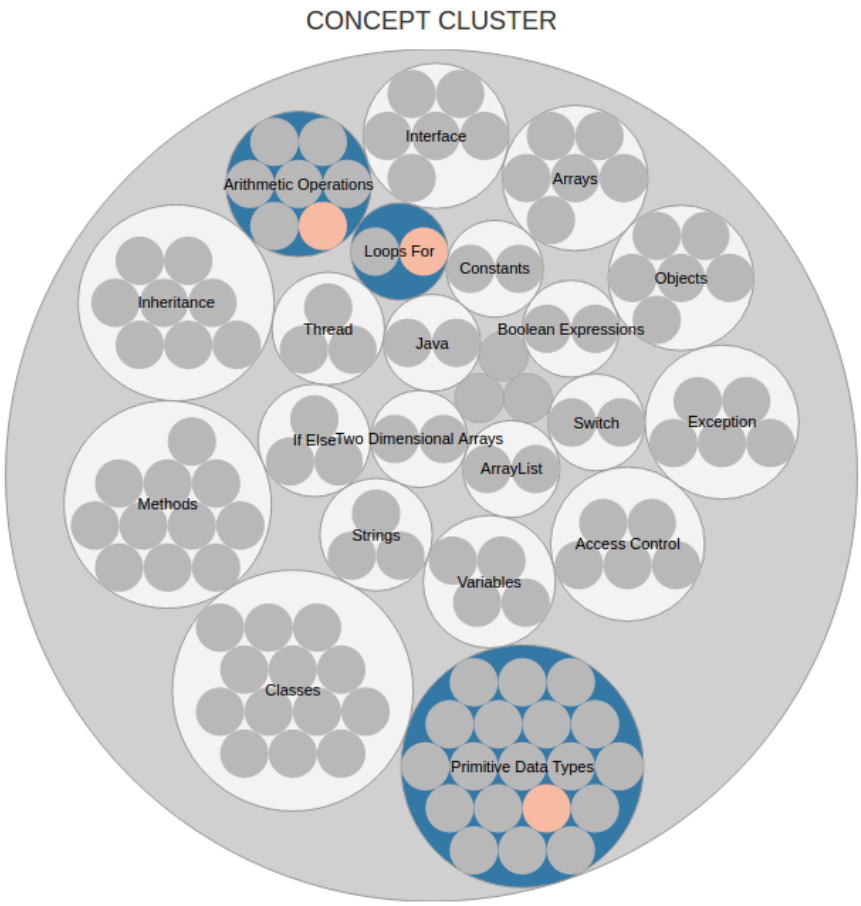

Figure 8. Authoring mode interface - concept pack

associated with the question. Teachers can select the bubbles in the circle pack to zoom in and out to examine the concept coverage. As soon as a concept is selected the right panel changes to show the subconcepts along with their corresponding weights for this particular question. These weights are represented in a slide bar that can be used to update the weights to emphasize or de-emphasize a particular sub-concept by adjusting the slider bar. Also, a concept can be excluded or included with the options in the right panel. The intensity of the colors of each bubble represents the

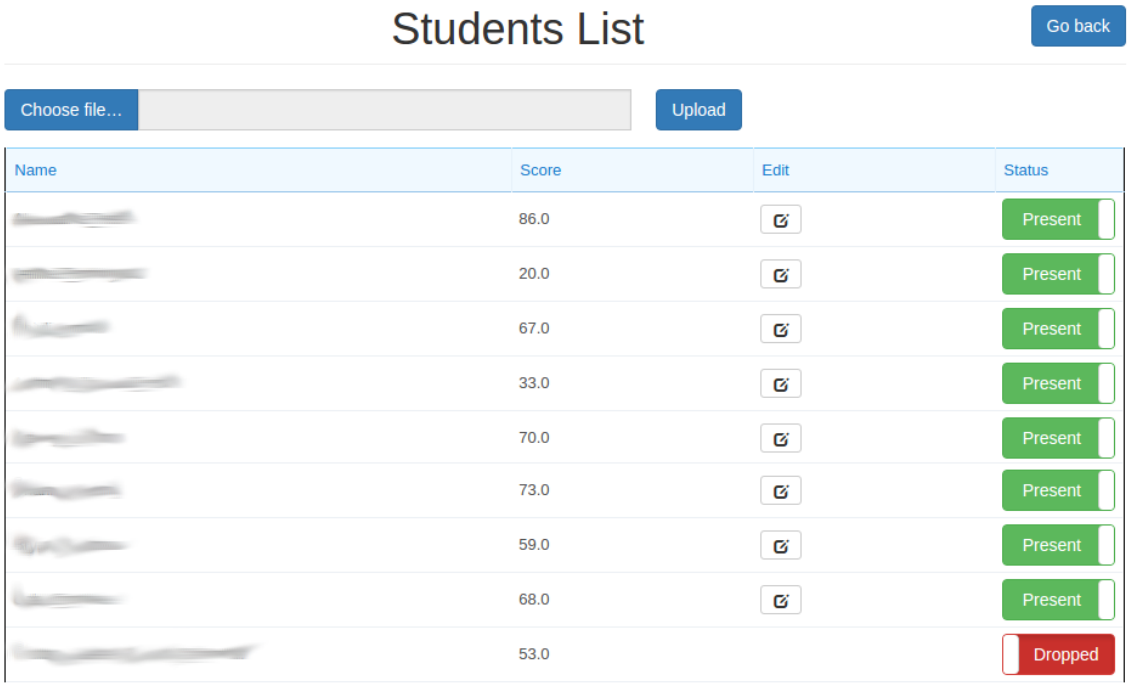

Figure 9. Evaluation framework

emphasis of that concept corresponding to the question. The darker the color the more emphasis given to the concept. This visual cue helps in clearly understanding the representative concepts for each question. The changes thus made can then be saved to the back-end. Also, the concept-mapping and the exam paper can be downloaded from the system for further redistribution or for recording purposes.

The semantic-enrichment of the questions results in the emphasis of different topics/concepts that are representative of the question itself. These concepts can be used to reflect a student's understanding once their answers are evaluated. The evaluation process for the paper exam has to be moved to a virtual setup to take advantage of the EduAnalysis system. To that extent EduAnalysis has a evaluation

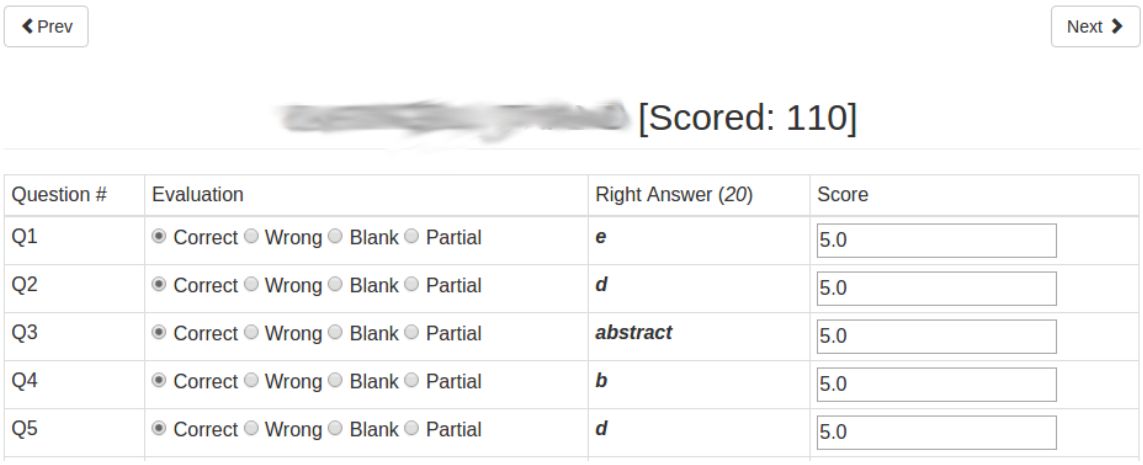

Figure 10. Student evaluation interface

interface corresponding to each class with a list of all the students of that class. The interface as shown in Figure 9 (names blurred) provides two different options in the current system to evaluate the students viz., a) manual online grading and b) upload evaluation from Blackboard like systems.

In each case the questions can be evaluated to be one of the four options as shown in Figure 10 (name blurred) namely: correct, incorrect, blank and partial. The first three options represent the obvious while the "partial" option is chosen when an answer is given partial credits, for example, in case of a short or long answer type questions. The teachers can also update the scores for each correct and partially correct answers to reflect the student's understanding and provide comments for each question. These options enrich the information that are latent in the answers by providing an overview of the entire class's understanding of the tested concepts. For example, based on the number of students who left certain questions without any answer (blank) we can estimate the understanding of the concepts corresponding to those questions.

This information is harder to collect when the papers are evaluated in the traditional manner unless the teacher or the grader keeps track of it separately. These information are recorded once for every student, for every exam and for every semester thus keeping a historical track of the student's record to analyze the student's performance over any time period. The interface also provides an option to select the student's status in the class as either continuing or dropped enabling updated information in terms of class statistics (e.g. mean, median etc.) The evaluation system in this particular version of the software works with the final evaluation reports from Blackboard systems for any exams. The software parses the formatted Blackboard file to obtain the correct answers for each question by the students without any manual intervention. This helps in keeping all the information for each class updated in EduAnalysis application acting as the single point of query.

3.2.2 Ontology Parser Backend

The concept mapping forms the core of the application. All the analysis visualizing the progress of the students at individual and class level are based on the concept mapping for the questions in an exam. The concepts are derived from an ontology that describes the hierarchical nature of the topics/concepts used in the application. According to [3] *"an ontology is a formal specification of a shared conceptualization."* Ontologies in information sciences are used to represent entities that exist in a particular domain along with their relationship with each other. This serves as a formal specification that can be used as the underlying dictionary for representing the different connections between entities of interest in a given formal language. The

ontology used in this version of EduAnalysis is based on [18] Java ontology, trimmed down for Java beginners course topics. The ontology in the system is made up of two parts: mapping of language specific keywords to concepts and mapping of concepts to the main concepts or topics that encompass them. For example, the following could be a two-part ontology as stored in the system:

keyword to concept: {"} do": ["DoStatement", "WhileStatement"]}
concept to topic: {"WhileStatement": ["LoopWhile"], "DoStatement": ["LoopDoWhile"]}

This forms the basis for the semantic parsing of the questions. The semantic indexing of the questions depend on this ontology to identify and match the valid ontology entries for a given domain. Replacing this ontology with another ontology corresponding to a different domain of interest will adapt the system to parse the semantic information from that specific domain. This core part is also the replaceable part of the system to adapt the entire system to whichever domain it is intended to work on.

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. The ExamParser module inherits from a generic Topic Facet Model [16], which consists of a natural language parser and a domain specific language parser. It recognizes exam questions and answers in a document based on a predefined set of patterns and extracts the relevant content which will be used for further indexing stages. The content of each question is then parsed and augmented by indexing each question to its corresponding concepts (a high level concept topic and sets of facets). A typical exam question pattern includes

question_text (phrased as natural language text, may or may not contain domain specific terms or concepts), codes (composed from full or partial code of an entire executable program), and answer_type (ranges from multiple choices, fill-in-the-blanks, short answers, code writing etc.) For instance, an exam question looks like the below sample.

Q. What is the final value of sum displayed to the console:

```
for(int i = 0; i < 5; i++) {
  int sum = 0;
  sum = sum + i;
  System.out.println(sum);
}
```

a. 5

b. 10

c. 15

d. Compiler error

e. None of the above

It consists mainly of natural language phrased question description, a piece of partial, executable java code, and multiple choice answer type for this question. ExamParser will take the textual content of the question as the input and translate this text into a set of weighted concepts. For example, the question in the above code yields the following concepts as output from the ExamParser:

ForStatement:1, VariableInitialization:1, IncrementOperator:1, MethodInvocation:1, AssignmentOperator:3

The weights are provided based on the count of appearances of each of these concepts. 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 subjective and depends on the teacher and the progression of the class. Therefore, we designed a dynamic concept indexing authoring interface as described in the previous section. It labels each parsed concept with an equivalent, default quantity weights, but the weights are adjustable according to the teacher's emphasis. This puts the teachers in control over deciding the importance of the identified concepts. The automatic indexing identifies concepts better than the teachers, as explained later, and lets the teachers configure the concept emphasis as they see fit. In the case of Question 1 in a CS1 midterm exam, the focus could be on *ForStatement* and *ConditionalStatement* concepts; using the same question in a CS1 final exam, every concept could potentially weigh equally proportionally. In addition, providing dynamic concept weight authoring interfaces not only allow teachers to include or exclude additional or redundant concepts to exam questions, but also enables dynamic exam content editing and corresponding concept indexing (Figure 11).

That is the teachers can include new concepts or remove existing concepts from a question or totally replace the question text with another question and get the updated concept indexing dynamically. This dynamic authoring mechanism along with intelligent parsing is designed to help improve teachers' flexibility to configure

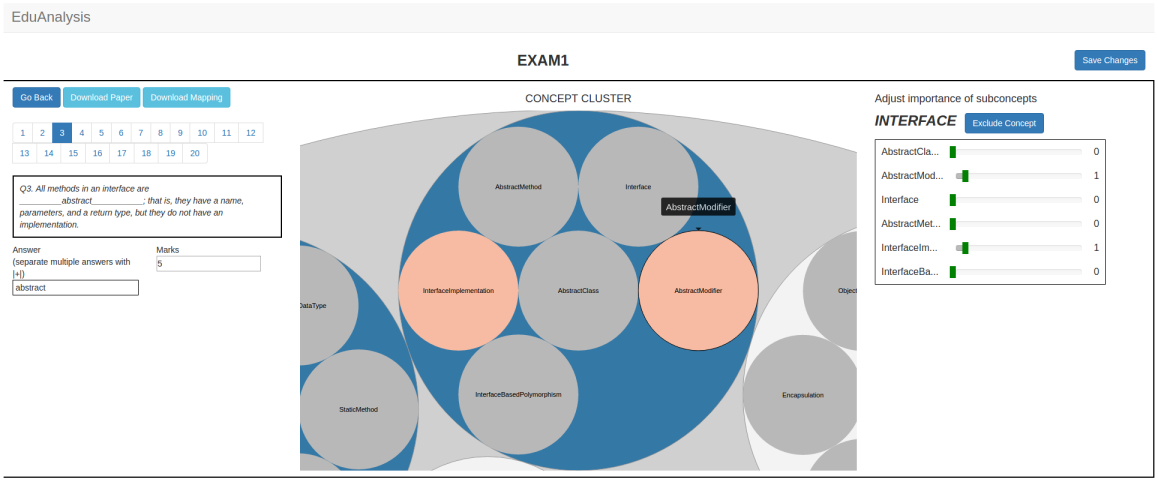

Figure 11. Configuring concepts indexed using the sliders

and coordinate entire exam's topical emphasis, at the same time, complement to the algorithmic flaws, in case of any missing or addition of lesser relevant concepts.

The concept indexing method enables an extensible framework in two essential educational technology aspects: (1) systematically assign partial credits, which are traditionally provided by teachers' or graders' experiences or generic grading rubrics (such as credits to right path toward key concepts but erroneous implementation). By associating each programming problem to a set of weighted concepts we can facilitate quantitatively distributing partial credits at the semantic level in an organized fashion. (2) harness different levels of learning analytics on both individual and group levels, including strong and weak concept clusters, co-occurrences of misconceptions, conceptual progress over time etc. Currently we focus on aggregating various levels of semantics analytics that are provided as overviews to teachers. But these results can serve as detailed feedback to students as well. For instance, on turning back the graded exam papers to students if we also include a detailed feedback on what

kinds of errors they made on the exams, the students will no longer just receive the grades but also a better understanding of the specific areas that they need their focus and attention. The approach provides (1) individualized detail conceptual feedback, which normally cannot be done especially on large class size; (2) analytics to keep persistent traces on students' conceptual growth; (3) opportunities for students to engage in self-reflection and self-monitoring their own learning (foster meta-cognitive development).

3.2.3 Analytics Dashboard

One of the main purpose of this application is to eventually provide analytical overview of the progress of a class and the individual students in an enhanced classroom environment. Also, these information on the individual and group standings can be fed back to the students to provide a targeted direction of focus toward improving their performance in the course. In a traditional paper exam based course evaluating all the papers takes significant amount of time but the result at the end of it is reflected only by the grades scored and gives less to no feedback on what went wrong. All the efforts put into evaluation is not presented to the students and giving each student individual care and feedback is tougher with growing class sizes. Even the basic calculations of median and average of a class for a given exam takes significant effort and harder to keep updated. To tackle this lack of feedback in the system and to provide deeper insights into the collective progress of the students we have designed the analytics dashboard for students and teachers alike.

The feedback here is focused on letting the students know of the particular concepts

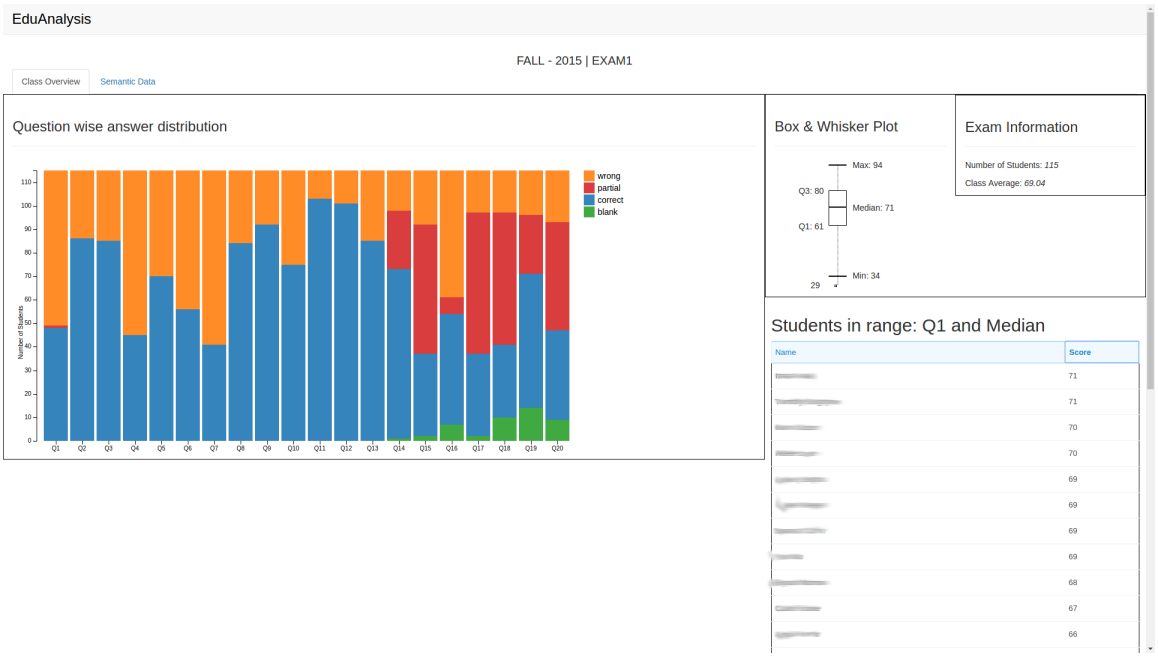

Figure 12. Student dashboard visualizing basic analytics on score

they missed more compared to the other concepts and their peers. This feedback can also help the teachers assess the overall performance of students on different exams throughout the progression of the course. Figure 12 presents the interface for the analytics dashboard built into EduAnalysis.

There are two tabs with different analytical visualizations representing different insights on the exam. The first of it presents an overview of the entire class's performance on all the questions in that exam. The right panel provides the basic summary information of the class's performance in that exam. The box plot provides the distribution of students across different quartile ranges of the marks. The outliers appear beyond the "max" and "min" representation in the plot. The students constituting these different ranges will be displayed along with their marks when the specific

quartile points are clicked. The table in Figure 12 shows the list of students in the quartile range *Q1* upto and including *median*. This provides a better understanding of the groups of students who need more attention thereby focusing on providing more feedback and personalized tests to these set of students. This features comes out of the box with EduAnalysis while it is a tedious task to obtain the same level of insight manually. The left panel in the overview page provides the distribution of answers across all the questions. Each bar represents a question, the colors represent one of the four answer types provided earlier viz. correct, incorrect, blank and partial. This answer type association now helps in determining how the students perceived each questions and find out the questions that were left blank by many or got partial credits. These questions might have been complex or tricky to the students or the concept was not properly understood by the majority of the classroom. That indicates a need to reinforce some of the concepts back in the classrooms to students. This information is also provided by the letters on top of each bar indicating whether the question was *easy* (E), *moderate* (M) or *complex* (C). Teachers can also look at the content of the questions at the same view without having to go back to find the question's content. The rich set of information in the form of the overview helps the teachers understand why students are not scoring in certain areas and who are the constant defaulters.

The second tab provides the semantic analysis of the overall class performance with each individual student's data as shown in Figure 13. Figure 14 provides a closer view of the student data presented in this tab. The panel displays a large heatmap with each rows of it representing an individual student in the class.

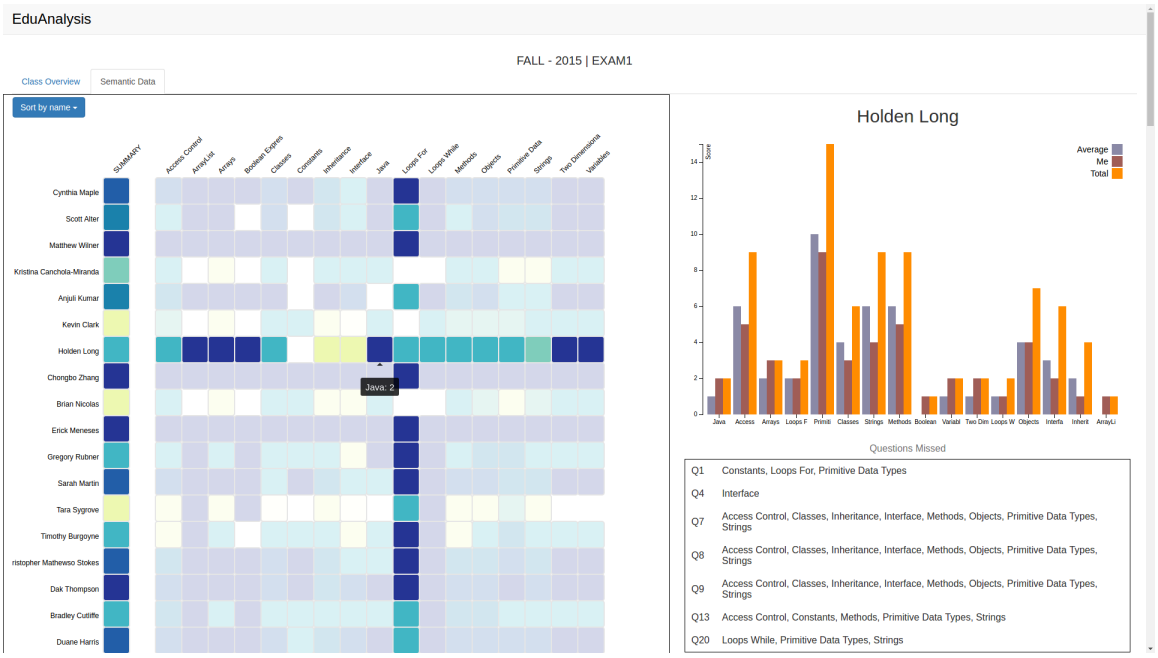

Figure 13. Insights into student's understanding of concepts

The students' performance is measured against the individual semantic concepts identified from the exam paper by the ExamParser. Each cube is color coded and the intensity of color goes from white for zero, through green to navy blue in color. The color intensity of the cube represents the student's score on the corresponding concept in the heatmap. This is based on the percentage of the count of that particular concept the student obtained by answering questions correctly over the total number of times this particular concept has occurred throughout the exam paper. The data comes from the automatic semantic indexing as described in the previous sections. Each correctly answered or partially credited questions add the weight/count of their corresponding concepts to the student's model. This is based on the assumption that the correctness of an answer ensures that the student has understood all the primary concepts for that question and hence the student shall be assigned with the weights of

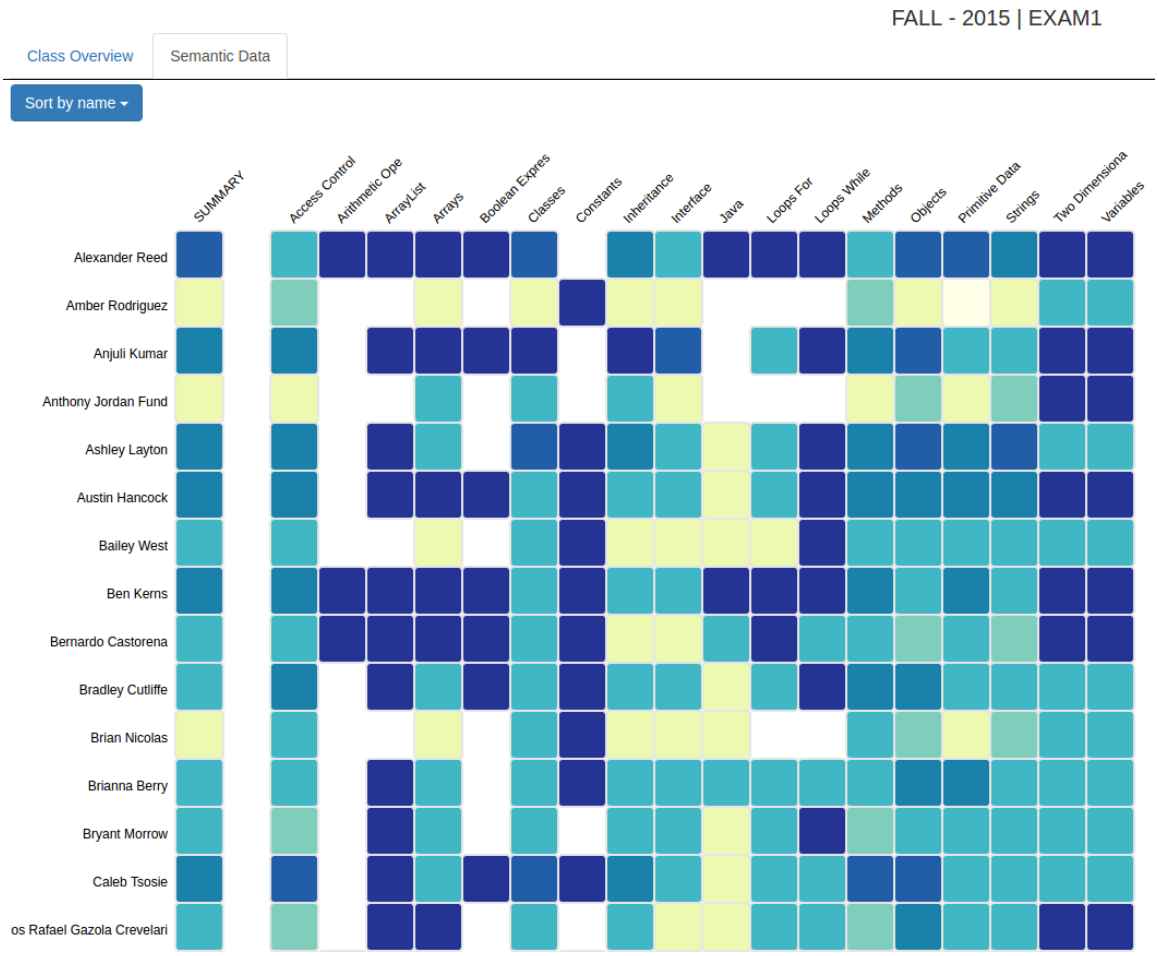

Figure 14. Student to concept heatmap

these concepts to reinforce their understanding. These scores are then summarized into one single cube termed "Summary." The summary provides an overall picture how different students have performed in the exam without looking at every individual block of different students. The student list in the heatmap can also be sorted by name or summary score for convenient view of the class performance in general.

For each student the teacher can also drill down into each student by clicking

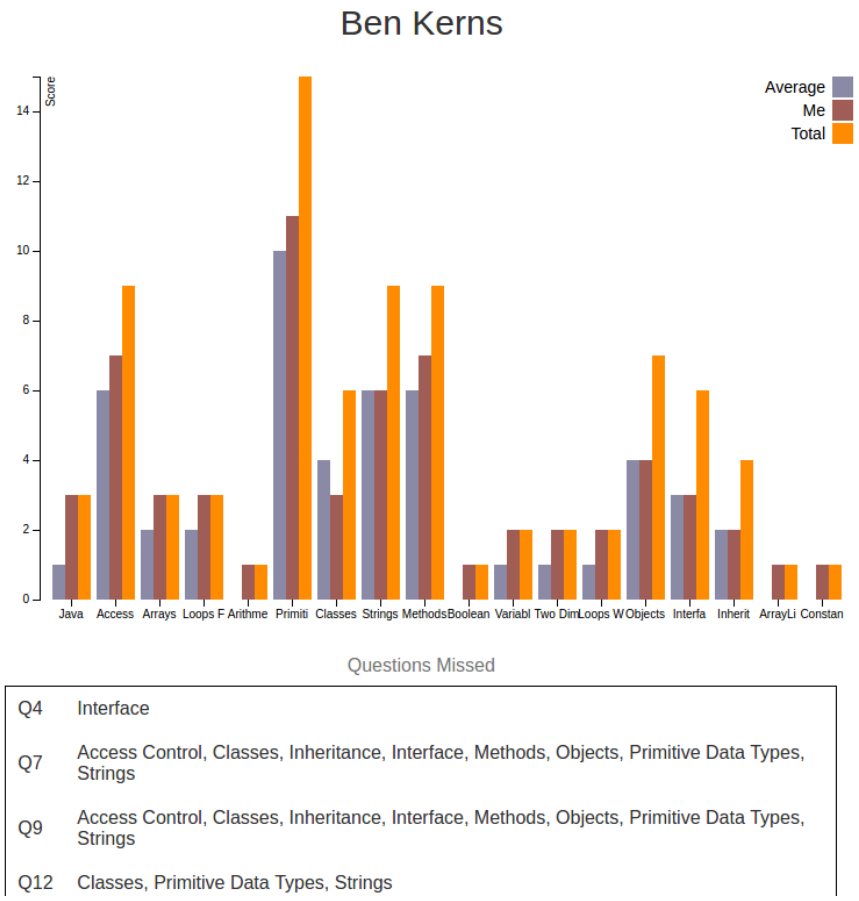

Figure 15. Detailed insights on student's exam

on any of the tiles corresponding to the student. This provides a visualization on the right panel, as in Figure 15, comparing the student's performance in each of the concepts compared to the class average and the maximum available. It also displays the questions the student missed answering or answered incorrectly along with their corresponding concepts that the student missed out thereby helping them know where they exactly lost their marks. The information provided from the analytics dashboard can also be helpful to students if they can directly look at their performance and

compare it to the class performance to know where they stand. The data used for these analytics and visualization are useful to both the teacher and student with some restrictions on the student access owing to other student names. The similarity in the need for viewing this as a feedback resulted in designing the whole analytics system build upon APIs. For both the semantic visualization and the overview of the class' performance the data comes from the API. It has been designed this way keeping in mind that these representations of students' understanding of the concepts as an individual and as a group can be of use to other entities and not just the teacher. The most important entity here is the student to whom this can be given as a feedback. Also, this information can be used to make personalized recommendations of academic resources to the students based on their strengths in the understanding of different concepts. For the students view owing to privacy we cannot show the names of other students in the heatmap view. With the data being provided by an API we designed a separate application for students with authorization to access these analysis and look at the analysis tab alone. This application can be further developed to incorporate different constraints for the students (or any other stakeholder) without changing anything for the teachers.

Chapter 4

EVALUATION AND RESULTS

To evaluate the proposed semantic visual analytics for programming courses, we performed (1) content evaluation to examine the impacts of the semantic approach on exam question content; and (2) subjective evaluation by collecting interview feedback from teachers to understand user experiences and system usefulness.

4.1 Data Collection

For the evaluation of the system we collected exam questions from 4 different exams of an introductory programming course, with a total 76 exam questions in the subject of Object-Oriented Programming. Each exam was uploaded into the EduAnalysis system; each question was automatically associated with a set of concepts through the ExamParser algorithm. In order to verify the embedded indexing algorithm's effectiveness, we collected the ground truth of corpus' concept indexing from 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, with an inter-rater reliability measure of 0.386 calculated using Cohen's Kappa.

4.2 Evaluation Metrics

We assumed an effective algorithm for the concept indexing would be able to index more relevant concepts or at least as many as the experts did. The relevancy of a concept to a question is determined based on the experts' indexing. If an expert associates a concept to a question then that concept is assumed to be relevant and a representative concept for the question. In addition to finding these key concepts, a good quality algorithm should identify peripheral concepts as well. Due to the reliance of student modelling on the concept indexing, the more concepts indexed the better the model of a student becomes. To verify our assumptions, we considered the following measures: Concept Coverage, Concept Relevance and Concept Configuration.

4.2.1 Concept Coverage

We defined the number of concepts indexed for each question as the concept coverage statistics.

$$coverage = \frac{\#indexed\_concepts}{\#questions}$$

4.2.2 Concept Relevance

To gauge concept-indexing quality on exam questions based on the semantic topic facets generated by the algorithm, we assume that a meaningful question not only has to achieve high concept indexing coverage, but also has to encompass as many essential concepts as possible. For this purpose, we consider a super-set of all the

concepts associated with a question by the experts as the most relevant, essential concepts to be indexed. The reasoning here is that the experts are expected to know the most relevant, key concepts that represent a question and hence, the concepts they provide are considered the absolute concepts that a question is associated with. We measure the concept relevance score as a percentage using the following equation:

$$concept\_relevance = \frac{\#(\{indexed\_concepts\} \cap \{baseline\_concepts\}) * 100}{\#baseline\_concepts}$$

The equation effectively calculates the ratio of the count of intersecting concepts between the identified concepts and the super-set of concepts associated to a question by the experts, over the number of concepts associated by the experts for that question and presents it in percentage.

4.2.3 Concept Configuration

The ExamParser algorithm indexes essential as well as peripheral concepts to a question. The importance of these concepts can be adjusted in terms of weight attached to each question. Also, these concepts can either be deleted from the association or be associated newly to a question. As mentioned before indexing more concepts is not necessarily a good measure of algorithm's effectiveness. By allowing teachers to add or delete concepts we give them the control to rectify the flaws of the algorithm. The lesser the need to change the concepts the better the algorithm's indexing. Thus we measure the number of additions and deletions needed for the concepts based on the experts' associated concepts to each question. Here any concept that does not match

the super-set of concepts associated with a question are considered for deletion though it will not be the case in real scenario. For example, consider the following question:

Q. Write the following for loop as a while loop:

```
for(int i = 0; i <= 10; i++) {
    System.out.println(i);
}
```

The following concepts are associated with this question where $C_{Baseline}$ is the set of concepts indexed by the experts and $C_{ExamParser}$ is the set of concepts automatically indexed by ExamParser:

$C_{Baseline}$: ForStatement, WhileStatement, IntValue

$C_{ExamParser}$: ForStatement, WhileStatement, IncrementDecrementOperator, AssignmentExpression

From the above concept sets we see that the ExamParser has indexed more concepts than the experts. Here *IncrementDecrementOperator* and *AssignmentExpression* are not present in the baseline group and hence are considered for deletion. Similarly *IntValue* in baseline is not in ExamParser's group and hence is considered for inclusion. But we can see that *IncrementDecrementOperator* is a peripheral concept which confirms that the student knows how the increments happen in a *for* loop. Most of the peripheral concepts could aid in better student modeling. Owing to the lack of tight baseline on peripheral concepts we consider their deletion as the ideal scenario in this evaluation.

Metric	ExamParser	Expert 1	Expert 2
Concept coverage	4.93 ± 0.41	2.41 ± 0.15	2.5 ± 0.1
Concept relevance	49%(152)	59% (183)	61%(190)

Table 1. Concept coverage and relevance

4.3 Evaluation Results

4.3.1 Results for Concept Coverage and Concept Relevance

ExamParser indexed 4.93 concepts per question on average, from a total of 375 concepts indexed, which is significantly higher than the baseline indexing of 312 concepts (super-set), $t(75) = 6.4408$, $p < 0.0001$ (Table 1).

This is partly attributed to the association of peripheral concepts which are not essential but still capture the trivial knowledge of students. These were usually ignored by the experts. This shows that the ExamParser could extract more concepts in general. However, as discussed earlier more concepts does not always represent a better coverage. If there were a lot of repetitions or shallow concepts, the quantity does not add more value at the semantic level. Hence, we measured the relevancy of the indexed concepts based on the baseline set of concepts. The results as in Table 1 shows that ExamParser was as close to the experts in finding the relevant concepts if not better than them. This demonstrated that the ExamParser not only extracted more concepts, but also extracted more relevant concepts. This was an encouraging note to the traditional paper based exam courses due to the fact that it has always

been challenging for the teachers to spend significant time to discuss every single detail of each exam question. It's common for teachers to focus on selected concepts (more relevant like the experts) instead of all concepts. However, ExamParser would be able to index more comprehensive set of concepts for each question and create a detailed feedback via analytics. This can potentially extend learning opportunities by complementing the brief feedback from teachers. In addition, this also provides persistent traces for engaging students to reflect and self monitor the results of their learning.

4.3.2 Results on Interaction and Usability

Interaction	ExamParser	Expert 1	Expert 2
Addition	2.13 ± 0.14	1.69 ± 0.10	1.61 ± 0.12
Deletion	2.89 ± 0.35	-	-
Concept indexing avg	4.93 ± 0.41	2.41 ± 0.15	2.5 ± 0.1

Table 2. Concept configuration

The purpose of the ExamParser algorithm is to semantically enrich questions with as many essential and relevant concepts as a human expert would do and index even more concepts. But the usability and performance of the system will be defined by how efficiently it reduces the teacher's time in authoring the question semantics and help them focus on completing the course objectives. In our case we measured this as

the amount of interaction the teacher has to make with the system to configure the concepts. The lesser the addition and deletion of concepts per question the better for the teachers to complete the semantic enrichment sooner. The values in Table 2 show the average number of concept additions and deletions required to represent all the baseline concepts associated with the question. The average addition value for ExamParser is not very different far from those of the experts. That is if the semantic information has to be completely represented then teachers have to add around 2.1 concepts on average to those automatically indexed by the ExamParser while the same for the set from experts goes upto 1.6. This shows that ExamParser finds most of the relevant concepts for each question given the average of experts' indexing is in the range 2.41 to 2.5 concepts per question. The deletion does not have an equivalent baseline since the experts constitute the baseline concepts and they have to add to their indexes those concepts they missed but do not have to delete any. Concept deletion with an average of 2.89 could seem higher at the first sight. But considering that we delete even the peripheral concepts owing to the lack of baseline which otherwise might be retained for detailed insight, this number might go down further.

4.4 Concept Indexing Effects on Content

We consider the following aspects to assess analytical impacts on domain content: Content Complexity & Content Knowledge Structure.

4.4.1 Content Complexity

Complexity	ExamParser	Expert 1	Expert 2
Easy	3.83 ± 0.46	2.17 ± 0.19	2.31 ± 0.16
Medium	4.37 ± 0.62	2.11 ± 0.21	2.47 ± 0.16
Complex	7.21 ± 1.12	2.74 ± 0.39	2.68 ± 0.19

Table 3. Average concept count by complexity

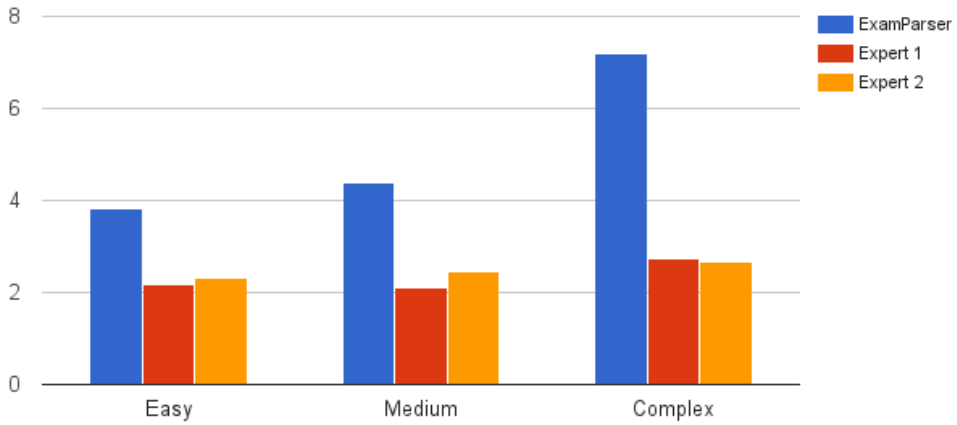

Figure 16. Concept count by complexity

According to CS1 course curriculum, depending on the exam's focus topics, we split each exam by three levels of complexities viz., 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 is usually devised with a mixture of difficulty level questions. Thus, a question comprising of multiple topics was considered as a complex question in that exam. We have tabulated two interesting findings in Table 3. Firstly, we found that human experts (experienced teachers) had no significant differences in the number of concepts they indexed among the three levels of complexities. The results supported the point that teacher experts tended to point out only key concepts instead of all concepts. Secondly, ExamParser indexed significantly more concepts in complex questions than the other two categories as displayed in Figure 16. More complicated questions were usually the ones where students made mistakes, which suggested more feedback can help there. However, teachers may not necessarily have the time to go through details on every single question. Even if they gave feedback 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 detailed feedback.

4.4.2 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 viz., 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. For example, the following multiple choice question is considered as **declarative**

The process of hiding object data and providing public methods to access the data is called:

a. instantiation

b. encapsulation

c. attribution

d. construction

The following is an example of a **procedural** question

Write a for loop that calculates aNumber! (i.e. 5!) where aNumber is a positive integer. You may assume aNumber has been initialized. Display the final result to the console.

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 their declarative knowledge. However, there were a few exception cases that 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 significantly higher concepts indexed by ExamParser than the experts as shown in Table 4. This was consistent with previous findings, where ExamParser achieved higher coverage. What was interesting to note was that there was no significant difference between declarative and procedural knowledge types of questions, no matter who indexed the questions. It showed the consistency among experts and also indicated the ExamParser's algorithm stability as it mimicked the experts in indexing quantity. As we anticipated procedural type questions indexed slightly more concepts compared to declarative types. This can be attributed to the fact that knowing how to do a coding problem may inherently involve some declarative knowledge components in addition to applying them to solve the problem. But this difference was not significant to further analyze on its impact.

Question type	ExamParser	Expert 1	Expert 2
Declarative	4.20 ± 0.49	2.35 ± 0.24	2.44 ± 0.16
Procedural	5.52 ± 0.62	2.45 ± 0.21	2.55 ± 0.14

Table 4. Average concept count by knowledge type

4.5 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 averagely). We were mostly interested in finding how instructors analyze students' learning activities outside classrooms if any. Both instructors provide extra online learning materials (i.e. problem-solving etc.) to 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 limitation of the 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 were worried that the auto-indexing precision may not be good enough and might 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.

Chapter 5

DISCUSSIONS AND CONCLUSION

5.1 Summary

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

We found that the proposed approach shed light on extracting semantic information from paper exams. These 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 topical emphasis in their exams and the results of indexed concepts appeared to maintain relevance within exam. It suggested that the proposed ExamParser approach could potentially make it possible to assign partial credits by concepts. The system being a web application provided a non-intrusive technology support that was one of objectives of this research. 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 time, but were forced to discuss key concepts on the tougher problems on finished exams. Moreover, we also found the automatic

indexing method was consistent with teacher experts in indexing both procedural and declarative types of questions. This also proved that we were able to enhance the exam questions with semantic information in a better and more encompassing manner. 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.

In summary, we tested a semantic visual learning analytics approach of orchestrating today's programming classes, by integrating physical classroom learning assessment (paper exams) onto online visual learning analytics without tampering teachers' instruction pedagogy. With a whole suite of services to enable creating exams, evaluating students and giving constructive, insightful and in-depth feedback to both students and the teachers EduAnalysis serves as a central system to maintain all the information about a class and the students. Results indicate that the automatic concept extraction from exams is promising and could be a potential technological solution to address a real world issue. There were a few limitations under current study setup, discussions were noted in the following section.

5.2 Limitations

There were a few limitations under current study setup. For instance, current exams selection was a sample of CS1's four exams from Arizona State 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's 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.

5.3 Future Work

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. This is currently ongoing in the form of a massive online survey collecting indexes for questions from various qualified professionals. 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, integrate other learning activities and recommendations for more comprehensive analysis.

REFERENCES

[1] Susan A Ambrose et al. *How learning works: Seven research-based principles for smart teaching*. John Wiley & Sons, 2010.

[2] Piyush Awasthi and I-Han Hsaio. "INSIGHT: a Semantic Visual Analytics for Programming Discussion Forums". In: *First International workshop on Visual Approaches to Learning Analytics, in conjunction with Fifth International Learning Analytics and Knowledge conference*. ACM. 2015.

[3] Aaron Bloomfield. "Evolution of a digital paper exam grading system". In: *Frontiers in Education Conference (FIE), 2010 IEEE*. IEEE. 2010, T1G–1.

[4] Susan Bull. "Supporting learning with open learner models". In: *Planning* 29.14 (2004), p. 1.

[5] Susan Bull and Mark Britland. "Group interaction prompted by a simple assessed open learner model that can be optionally released to peers". In: *Proceedings of Workshop on Personalisation in E-Learning Environments at Individual and Group Level (PING), User Modeling*. Vol. 2007. 2007.

[6] Zhi-Hong Chen et al. "Active open learner models as animal companions: Motivating children to learn through interacting with My-Pet and Our-Pet". In: *International Journal of Artificial Intelligence in Education* 17.2 (2007), pp. 145–167.

[7] Anna De Liddo et al. "Discourse-centric learning analytics". In: *Proceedings of the 1st International Conference on Learning Analytics and Knowledge*. ACM. 2011, pp. 23–33.

[8] Carrie Demmans Epp and Susan Bull. "Uncertainty Representation in Visualizations of Learning Analytics for Learners: Current Approaches and Opportunities". In: *Learning Technologies, IEEE Transactions on* 8.3 (2015), pp. 242–260.

[9] Pierre Dillenbourg, Sanna Järvelä, and Frank Fischer. "The evolution of research on computer-supported collaborative learning". In: *Technology-enhanced learning*. Springer, 2009, pp. 3–19.

[10] Vania Dimitrova, John Self, and Paul Brna. *Applying interactive open learner models to learning technical terminology*. Springer, 2001.

[11] Erik Duval et al. "VISLA: visual aspects of learning analytics". In: *Proceedings of the Fifth International Conference on Learning Analytics And Knowledge*. ACM. 2015, pp. 394–395.

[12] Anna Lea Dyckhoff et al. "Design and Implementation of a Learning Analytics Toolkit for Teachers." In: *Educational Technology & Society* 15.3 (2012), pp. 58–76.

[13] *EduAnalysis System*. 2015. URL: ec2-54-69-239-219.us-west-2.compute.amazonaws.com:5000.

[14] Mohammad Hassan Falakmasir et al. "The impact of social performance visualization on students". In: *Advanced Learning Technologies (ICALT), 2012 IEEE 12th International Conference on*. IEEE. 2012, pp. 565–569.

[15] Neil T Heffernan and Cristina Lindquist Heffernan. "The ASSISTments Ecosystem: Building a platform that brings scientists and teachers together for minimally invasive research on human learning and teaching". In: *International Journal of Artificial Intelligence in Education* 24.4 (2014), pp. 470–497.

[16] IH Hsiao and P Awasthi. "Topic Facet Modeling: Visual Analytics for Online Discussion Forums". In: *The 5th international Learning Analytics & Knowledge Conference*. 2015.

[17] I-Han Hsiao et al. "Progressor: social navigation support through open social student modeling". In: *New Review of Hypermedia and Multimedia* 19.2 (2013), pp. 112–131.

[18] PAWs Lab. *Java ontology v. 0.8.1*. 2009. URL: http://www.pitt.edu/~paws//ont/java.owl.

[19] Roberto Martinez Maldonado. "Analysing, visualising and supporting collaborative learning using interactive tabletops". PhD thesis. The University of Sydney, Australia, 2014.

[20] Roberto Martinez-Maldonado et al. "Capturing and analyzing verbal and physical collaborative learning interactions at an enriched interactive tabletop". In:

International Journal of Computer-Supported Collaborative Learning 8.4 (2013), pp. 455–485.

[21] Riccardo Mazza and Vania Dimitrova. "CourseVis: A graphical student monitoring tool for supporting instructors in web-based distance courses". In: *International Journal of Human-Computer Studies* 65.2 (2007), pp. 125–139.

[22] Andy Nguyen et al. "Codewebs: scalable homework search for massive open online programming courses". In: *Proceedings of the 23rd international conference on World wide web*. ACM. 2014, pp. 491–502.

[23] Jeremy Roschelle, Yannis Dimitriadis, and Ulrich Hoppe. "Classroom orchestration: synthesis". In: *Computers & Education* 69 (2013), pp. 523–526.

[24] Jeremy Roschelle, William R Penuel, and Louis Abrahamson. "Classroom response and communication systems: Research review and theory". In: *Annual Meeting of the American Educational Research Association, San Diego, CA*. 2004, pp. 1–8.

[25] Mike Sharples. "Shared orchestration within and beyond the classroom". In: *Computers & Education* 69 (2013), pp. 504–506.

[26] Rishabh Singh, Sumit Gulwani, and Armando Solar-Lezama. "Automated feedback generation for introductory programming assignments". In: *ACM SIGPLAN Notices*. Vol. 48. 6. ACM. 2013, pp. 15–26.

[27] James D Slotta, Mike Tissenbaum, and Michelle Lui. "Orchestrating of complex inquiry: three roles for learning analytics in a smart classroom infrastructure". In: *Proceedings of the Third International Conference on Learning Analytics and Knowledge*. ACM. 2013, pp. 270–274.

[28] Ravi Vatrapu et al. "Towards visual analytics for teachers' dynamic diagnostic pedagogical decision-making". In: *Proceedings of the 1st International Conference on Learning Analytics and Knowledge*. ACM. 2011, pp. 93–98.

[29] Katrien Verbert et al. "Learning analytics dashboard applications". In: *American Behavioral Scientist* (2013), p. 0002764213479363.