2018 ASEE ANNUAL CONFERENCE & EXPOSITION
JUNE 24–27, 2018
SALT PALACE CONVENTION CENTER
SALT LAKE CITY, UT

Paper ID #23464

Innovative Approach to Online Argumentation in Computing and Engineering Courses

Dr. Swaroop Joshi, Ohio State University

Swaroop Joshi is a Senior Lecturer in Computer Science and Engineering at Ohio State University. He is interested in a range of topics in Education Technology and Software Engineering, including but not limited to Computer-Supported Collaborative Learning, Game-Based Learning, Programming Languages, Compiler Construction and Optimization.

Dr. Neelam Soundarajan, Ohio State University

Dr. Neelam Soundarajan is an Associate Professor in the Computer Science and Engineering Department at Ohio State University. His interests include software engineering as well as innovative approaches to engineering education.

Dr. Jeremy Morris, Ohio State University

Jeremy Morris has been an Assistant Professor of Practice at The Ohio State University since 2015. He completed his PhD at The Ohio State University and his research interests lie in both artificial intelligence and Computer Science education.

Innovative Approach to Online Argumentation in Computing and Engineering Courses

1 Introduction

Many researchers (e.g., [8, 16, 18, 24, 25, 26] and others) have stressed the importance of *argumentation* in science education to help students develop deep understanding. This work has mostly been at the K-12 level; but argumentation is even more important for undergraduates in engineering and computing (and other STEM fields). Not only will argumentation help engineering students master concepts, it will also better prepare them for their professional careers where they can expect to engage in vigorous arguments about trade-offs in various approaches to addressing problems in their design/implementation projects.

Prior research has shown that some key requirements must be met to ensure that argumentation is most productive: The argumentation must be in small groups of 4–5 students each; each group must include students with *different* approaches to the topic; and the instructor should *not* participate in the discussion. The last requirement may seem surprising but it is critical since, otherwise, the students will simply accept whatever the instructor says and the goal of helping them achieve *deep* understanding will be compromised. Even if we succeed in meeting these requirements, there are a number of challenging issues that must be addressed if argumentation is to be widely used in computing and engineering courses. First, how would faculty find time in their already packed courses to accommodate small-group argumentation to any serious extent? Second, wouldn't the most vocal students dominate such discussions while others, possibly more knowledgeable ones, stay in the background? Third, wouldn't stereotypical biases based on race, gender and the like, that some students may harbor concerning the abilities of other students seriously affect the discussions? Etc.

We have developed a highly innovative approach and *online* system, *CONSIDER*, to address these and other problems. A CONSIDER discussion starts with the instructor posting, on the CONSIDER *web app*, a suitable problem. Each student then submits her individual answer by a specified deadline. Next, the instructor uses the system to form groups consisting, typically, of 4 students each based on these submissions, with each group including students with *conflicting* ideas about how the posted problem may be tackled; and the discussion begins. The discussion may be specified to be *anonymous* with students in each group being labeled S1, S2, S3, S4 or they may know each other's identities; the discussion may be organized in a series of *rounds* with each student making one submission in each round and the other students not seeing the submission until the start of the next round or it may be organized in a more *forum-like* manner with each submission becoming available to the entire group as soon as it is made; etc. In each case, the student is required to explicitly specify whether she agrees or disagrees with the positions of each of the students in the group.

As we will see, the current version of the CONSIDER system, following the approach recommended by the *design-based research* (DBR) paradigm [2, 11, 23], has evolved through a series of versions, each revision being informed by instructor and student experience. The name CONSIDER is an acronym for *con*flicting *s*tudent *i*deas *d*iscussed, *e*valuated, and *r*esolved (or *r*efuted!). As the acronym suggests, the central goal is to ensure that each student in a group carefully considers conflicting conceptions held by the other students in the group and, as appropriate, refine/-correct their own conception of the topic in question.

We have used CONSIDER in a number of junior-level computing courses. The results were quite positive as were student reactions. We present the results, summarize the lessons learned, and ideas for improvements. In Section 2, we summarize the theoretical framework underlying the approach and other related work. In Section 3, we describe the CONSIDER system; as we noted, the system evolved through a series of versions and we describe some of these steps as well. In Section 4, we present our research questions, the research design, and the results obtained. In Section 5, we consider the lessons learned thus far and our plans for further revision and refinement of the system based on these results.

2 Theoretical Framework

Socio-cognitive conflict, a key concept underlying the CONSIDER system, originates in Piaget's classic work [19] on children's learning. The idea is that socio-cognitive conflict, i.e., disagreements with *other learners'* conception of the same problem or topic and interaction with peers to resolve the disagreements is fundamental to the learner's grasp of new concept since it highlights alternatives to the learner's own conception. In resolving the conflict, the learner is forced to consider and evaluate these alternatives on equal terms. Although Piaget was concerned mainly with the intellectual growth of children, his ideas are very relevant for adult learners as well, including undergraduate engineering and, more generally, STEM students. Indeed, the approach should be more effective for these students than for the young children since these students should be more capable than young children of analyzing and evaluating others' ideas that might conflict with their own. And given the serious problem of *misconceptions* harbored by students in different STEM disciplines that researchers have investigated [22], an approach which holds the promise of addressing such misconceptions is clearly worth pursuing.

A different approach, one that has been commonly used, to trigger cognitive conflict is for the *instructor* to present *anomalous data*.The expectation is that the conflict between the presented information and the student's prior conception will trigger "disequilibrium" in the student's thinking and cause the student to revise her conception. But Chinn and Brewer's work [4] showed that this approach failed to trigger conceptual change in a majority of college students in STEM courses. Given the authority of the teacher, many students seem to simply accept whatever the teacher says without much analysis. To put it differently, in cases where a student's understanding conflicts with the explanation provided by the instructor, the student simply accepts the explanation without critical evaluation. By contrast, when the (cognitive) conflict is between a given student's conceptualization of the topic and those of her *peers*, the student is forced to evaluate the alternatives critically and pick one after careful deliberation since she, rather than the peer, may be the one whose explanation is correct[1]!

Cognitive conflict is also the primary driving force behind the (in-class) *peer instruction* (PI) technique developed by Mazur [7] and Dufresne *et al.* [9] and others. In Mazur's approach, first, each student answers a conceptual multiple choice question submitting the answer via a *clicker* or other similar device; then the students turn to their neighbors and, in groups of 3 or 4, discuss the question; after a few minutes of discussion, each student again answers the same question. During the discussion time, the instructor may walk around the room but deliberately does *not* participate. Mazur reports that the percentage of students who, following discussion with their peers, change their answer from a wrong choice to the correct one far exceeds the percentage who change from the correct choice to a wrong one. However, there are a number of limitations with this and other similar approaches, mostly related to the fact that it is a classroom technique with the activity being interspersed with regular lectures by the instructor. First, since the topic in question was just discussed in the lecture, students are not likely to have had time to think about the ideas underlying the question. Second, a group may or may not include individuals who picked different possible answers since the grouping is based essentially on where students are seated. In addition, some students, not necessarily the ones with the most developed understanding of the topic, may dominate their groups; and any stereotypical biases that students may harbor, perhaps subconsciously, may compromise the discussion. Further, the amount of time spent in the discussion is, naturally, limited; hence, students who are not quick to speak and take time to formulate precise and deliberate arguments may not contribute effectively. In our approach, as we will see, all these problems are addressed effectively by appropriately utilizing specific affordances of web technologies. Thus, e.g., the discussion in a group may be organized in a series of *rounds*, thereby ensuring that no one student or set of students dominates the discussion. As another example, students in a group may be labeled, say, *S1, S2, S3, S4*, each student in the group knowing the others in the group only by their respective labels, thereby reducing or eliminating the impact of any biases.

Turning to some of the existing work related to *on-line* approaches, one of the earliest was CSILE [21] which allowed students to add, modify, comment on, or delete content to *build knowledge* collaboratively. More recent systems along these lines use *wikis* since wikis can support a variety and range of learning activities and types of interactions among students [14]. Unfortunately, many of these efforts have not been effective in ensuring individual learning even if wiki-based knowledge-building efforts outside the classroom have been quite successful, the best example being Wikipedia. Thus Cole's [6] course on *information systems* with 75 students in it was organized so that lectures were in alternate weeks, the other weeks being intended for students to discover new material and post to the class wiki. Students were told that fully one quarter of the questions on the final exam would be from the material they posted. But halfway through the course there had been *no* posts to the wiki! Leung and Chu [15] in a course on *knowledge management* and Judd *et al.* [13] in a large course on psychology report equally poor results of the use of a wiki. Rick and Guzdial [20] report that although they obtained positive results using wikis in architecture and

[1]More commonly, the student will revise her original conception incorporating ideas from other students' conceptions rather than simply abandoning her original conception and picking one of the others.

english composition classes, the results in STEM classes were "overwhelmingly disappointing". For example, they report that fully 40% of math students settled for a zero on an assignment rather than engage in collaborative learning!

In one respect, the work on collaborative learning is not directly relevant to our approach since the kinds of activities considered in much of that work, e.g., team projects in capstone design courses as well as in several of the systems listed above do not, for the most part, involve students in a team trying to resolve cognitive conflicts. Indeed, students in such teams often go out of their way to not criticize the ideas offered by other members of the team for fear of offending them. More relevant for us is the work on the role of *argumentation* in learning. But, as Driver *et al.* [8] put it, "[although] argument is a central feature of the resolution of scientific controversies, science teaching has paid ... little attention to [this] practice ... It is our contention that ... if science education is to help young people engage with the claims produced by science[or engineering, we might add]-in-the-making, science education must give access to these forms of argument through promoting appropriate ... activities" Given that CONSIDER, as we will see in the next section, requires students to offer arguments defending their positions, it has the potential to help students develop strong argumentation skills. But we should note that our *primary* goal is to help students to develop deep conceptual understanding; the fact that, in the process, they will develop strong argumentation skills is an added bonus not the main goal. In Nussbaum's terms [17], our interest is in having students "arguing to learn" rather than "learning to argue".

3 The CONSIDER System and Design-Based Research

Over the last twenty five years or so, the concept of *design-based research* (DBR), see, e.g., [2, 3, 5, 11, 23], has become widespread. The main idea of the DBR approach is to use feedback based from actual users to revise and iteratively improve the tools we are designing. The CONSIDER approach and system has, as we will see, evolved over many iterations. The key foundation for the CONSIDER approach is, as described in the last section, provided by the central notion that socio-cognitive conflict among learners how its resolution can drive the development of deep understanding among them. We built a prototype implementation and over several semesters of use in CSE courses, on the basis of feedback from instructors and students, revised and refined it in an iterative cycle. In this section, we will focus on the current version of the system but, along the way, will also mention some of the refinements the system has gone through.

Figure 1: Structure of CONSIDER approach

Figure 1 depicts the overall structure of the CONSIDER approach. The details of the box labeled "*Peer Discussions*" will be tailored to effect key variations in how the students in a group engage in the discussion as we will see below. *Class Lecture* is simply a standard lecture or lectures on a specific topic of the course. The boxes labeled *Pre-Test*, *Peer Discussions*, and *Post-Test* will take place on the CONSIDER system.

Let us now consider some details of the approach. We will use an example from our Software Engineering (SE) course. This is a fairly typical SE course, taken by juniors/seniors majoring in CSE. A main goal of the course is to help students recognize the importance of a systematic approach to understanding the overall *domain* in which the software system to be built is intended to operate, understand the *problem* in that domain that the software system is meant to help address, and the *solution approach* to be adopted in the software system. Quite often, however, students want to jump straight into designing and coding the software system without going through a careful analysis of the domain, the problem in the context of the domain, etc. Indeed, frequently there is confusion between the domain problem and specific algorithmic or data-structure related problems that might be encountered when developing the software system. The assignment below may be intended to help correct such misunderstandings.

> **Homework:** Your team has been asked to build a campus wayfinding system to help visually impaired students. Identify the category of analysis –that is, domain, problem, or solution– for each of the following elements, identified during analysis. Briefly explain why.
>
> 1. A catalog of the various types of building on a college campus;
> 2. The list of hard-to-find buildings on campus;
> 3. The range of visual and cognitive impairments that people suffer from;
> 4. Strategies by which people find their way in an unknown area – such as asking passers-by or by identifying major streets.

Item (3) is especially interesting. Many students think it falls under the *problem* category. In fact, however, it is part of the *domain* as it provides information about the range of impairments people suffer from. The software system, after all, is *not* intended to solve the problem of visual impairments (as for example would be the case if we were designing software for an artificial eye to help the person see). Different students come up with different answers and with different justifications. The standard approach would be to have a discussion on the question in class, typically in the same class period as the one in which the graded homeworks are returned to the students. The class discussion helps some students, but others remain unclear about the distinctions between the notions of domain, problem in the domain context, and solution. The key question that motivated our work was, how do we help students overcome such misconceptions and develop deep understanding?

In the CONSIDER approach, following the lecture(s) on the topic, the instructor would post the homework on the CONSIDER system. The instructor will also specify a deadline by which each student will be required to submit her answer. The homework may be similar to the one above but, for this discussion, we assume there is only one question, item (3) from the example. Once the instructor has posted the homework, each student will receive an email from the app asking her to log into the system and answer the question by a deadline, typically 24-36 hours away from the time the homework is posted, with the deadline also being listed in the email. The app will require the student to make a specific choice –such as "domain" or "problem" or "solution", and to include a brief *justification* as part of her answer. We will refer to this as the student's *initial submission*. Note, these initial submissions are made by *individual* students and each reflects the particular student's (initial/current) conception of the problem. Also, a student can log back in any time before the deadline and modify her answer if she wants to. Figure 2 shows the initial submission made by one of the students. This student indeed has a misconception and thinks the specified item belongs to the *Problem* category; and provides her rationale. Once the deadline for the initial submission expires, the system will (try to) automatically form groups of 4 or 5 students each with each group containing students who chose different answers. If most students picked the same choice, the instructor will have to form the groups based on differences in the students' justifications[2]

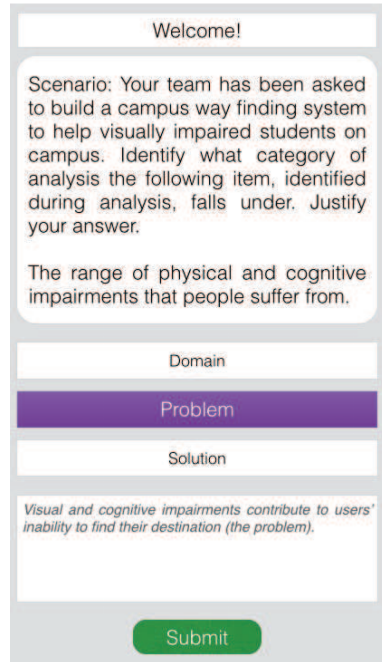

Figure 2: S2's Initial Submission

Before considering how the discussion takes place, we should note that when the instructor initially creates the homework assignment on the app (using an instructor interface which we will not detail), she will also specify various important aspects of the discussion such as whether the discussion will be *forum-based* or *round-based* (see below); whether it will be anonymous, so that students in each group will know each other as *S1, S2* etc. or students will see the identities of the other students in the group; the deadline for each phase of the discussion; in the case of round-based discussions, this will include specifying the number of rounds and the deadline for each round and it will include specifying the deadline for the final submission phase. For our current example, let us assume that the instructor has chosen the discussion to be anonymous and round-based. In a round-based discussion, each round will be of a fixed duration (specified by the instructor when she created the assignment). During each round, each student is required to make one post by the deadline for the round; but the student may log in as many times as she chooses before the deadline and edit her post as she chooses. As in the case of the initial submission, only the most recent version of the post will be saved. In our experience, a duration of 24 hours per round seems ideal. It allows students time to correct any mistakes they might make when making an early submission during a given round, and accounts for the varying time schedules of students. Also in our experience, the appropriate number of rounds for typical homeworks in courses in computing at this level seems to be two. In any case, the system allows the instructor to tweak these parameters to suit the particular course, the nature of the homework and the instructor's and students' preferences.

Suppose the student whose initial submission is shown in Fig. 2 has been assigned to a group, call it G, that has three other students. Since this is an anonymous discussion, the system will assign the labels S1, S2, S3, S4 to the four students. Let us assume that the system has assigned the label S2 to our particular student. When S2 logs in, once the first discussion round begins, she will be presented with the initial answers submitted by each student in (including herself), Figure 3.

[2]Clearly, such an approach would be impractical for large classes. We will return to this issue later in the paper and consider possible changes to the approach to allow automatic group formation.

Only the initial answers of S1 and S2 (the current student) appear in the screenshot due to lack of space in the window (but see later); S2 will be able to see the answers submitted by S3 and S4 by scrolling down as needed in the central window. Note that *S1* has submitted the correct answer. The hope is that when S2 reads this correct answer, she will resolve the resulting conflict by correcting her own conception.

At the bottom of the screen is a window where S2 is expected to type in her post for the current round, i.e., $Round_1$. Unfortunately, our hope was not realized. S2 did *not* understand the rationale behind S1's answer and, therefore, in her post for the current round which she has typed into the window at the bottom tries to justify her incorrect conception. Before we continue to the next round, a couple of points should be made. The other students in the group, S1, S3, S4 may also be logged in at the same time as S2 and working on their posts for this discussion round; some may have already made their posts; some may not yet have logged in. In none of these cases will any of the students see the posts that any of the others in the group may have made for the *current* round. This allows students to work at their own pace and, since a student can log in again and edit her post for the current round, students who may be quick with their answers or deliberate and thoughtful or anything in-between, to all work equally effectively. This is the key motivation for the round-based approach.

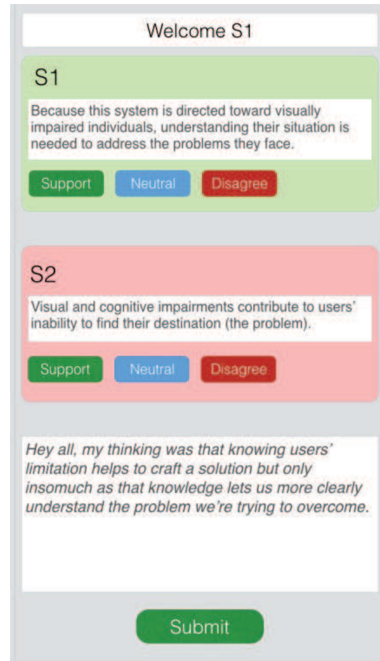

Figure 3: S2's Round 1 Sub.

We made an important revision to the system based on user feedback. Our original intent was to have students use the system on smart-phones since most of them carry a smart-phone at all times and check it frequently. After a couple of semesters of use, though, the feedback from students strongly suggested that while they appreciated the ability to access the assignment on their phones and while the initial submission, especially the part that required them to indicate a particular choice such as "Domain" or "Problem", could indeed be completed effectively on the phone, the discussion phase that follows the initial submission was much more challenging. It required, as we will see below, reading and thinking about the submissions of the other students in the group, possibly comparing them against each other and against the student's own original submission, etc., which was not convenient on a small screen. Even the rationale that the student submits as part of the initial submission could be challenging to enter –without unintended mistakes introduced by auto-correct software!– on a small-format smart-phone. Hence, we abandoned the original smart-phone implementation and moved to a web-app intended to be used on a laptop or desktop; this new interface is used in Fig. 4. It shows S2's Round 2 post. At this point, she has understood the key rationale for S1's position and recognized the correctness of that position.

To summarize, S2 is required to consider posts made by each student in her group in the previous round and analyze its relation to her current position. S2 has to do this for her *own* post as well from that round. This is important because S2 may find the post(s) of one (or more) of the other students from the previous round so compelling that she changes her mind and no longer agrees with what she herself said in the previous round! She would do this by clicking the "Disagree" button against her own previous-round post; her current-round post which she types into the window at the bottom should provide an explanation for why she no longer agrees with that position. In a real sense, *this* is the essence of *conflict-driven* collaborative learning: S2 had, in a previous round, a specific conception of the topic in question and posted that conception; similarly, other students in the group posted their (then-current) conceptions – without having access, at that point, to S2's conception; and, in the next round, when S2 reads all the posts from the previous round, the ideas expressed in those posts force to reconsider her own conception. The same may happen to the other students in the group.

Figure 4: S2's Round 2 Submission

The system also enables the student to navigate to earlier rounds (to *read* the posts in those rounds and) return when done. This is useful since, occasionally, a student will want to go back and see exactly how the discussion had gotten to the current stage. The discussion will end after the specified number of rounds at which point the *final submission round* will begin. In this round, each student in G will be required to, *individually*, submit a *summary* of the discussion in G and her own *final answer* to the question. The student's grade for the activity will depend *only* on this final submission. This means two things; first, there is *no penalty* for changing one's answer since the grade depends on the correctness of the student's final answer, not on what her initial answer was; in fact, there is a penalty for *not* changing the initial answer if it was, in fact, incorrect. Second, she had to be fully engaged in the group discussion since, otherwise, even if her final answer is correct, her discussion-summary would be of poor quality and her grade would suffer.

4 Experimental Results

The discussion in Section 3 focused on the *rounds-based* approach with the students in each group being anonymous, referring to each other only via the system-assigned labels, S1, S2, etc. CONSIDER also allows the instructor to choose the *forum-based* approach for the discussion. A forum-based discussion on CONSIDER is similar, in many respects, to forum-based discussion on tools such as *Piazza* (piazza.com). One important difference is that, in CONSIDER, it is still a discussion within a small group of 4–5 students; Piazza also allows the instructor to create similar small groups but the default that most instructors who use Piazza adopt is a forum that includes *all* students in the class. Perhaps even more importantly, in CONSIDER, the *initial submission* and the *final submission* phases are exactly the same whether the discussion is to be a rounds-based one or a forum-based one; only the discussion phase changes. This allows the instructor to form groups that include students with differing conceptions of the problem which is, of course, a key requirement of the approach. In a forum-based (henceforth FB) discussion, there is no notion of a "round", unlike the rounds-based (henceforth RB) discussion approach, elaborated in the last section. Instead, once the groups have been formed, each group engages in a discussion with each student posting as frequently or infrequently as she chooses; and the other students in the group see each post as soon as it is made.

A key question then is, which of these approaches, rounds-based or forum-based, is more effective in helping students develop deep understanding? In order to answer this question, we conducted an experiment in our junior-level course on programming language (PL) concepts. This course is similar to ones offered at other universities for these students. 27 of the 40 students enrolled in the class consented to participate in the research. One of those students did not participate in one of the two activities, so he was excluded from the analysis. The 26 students (participants) were CSE majors in their junior or senior year, in the age range 21–24. About 90% of them were white, others identified as Asians or Pacific Islander; some of the participants were mixed race. About 90% of the participants were males.

Type inferencing is a major part of Scala. This is a 2-part question.

1. Consider the following: "In some situations, the Scala programmer can omit *all* type information; in others, he/she has to provide *some* but not all; in yet others, complete type information (as you might in an equivalent Java program) has to be provided." True or false?
    - If true, provide examples of each of the three cases.
    - If false, explain what part of the statement is false and why.
2. Do languages such as Ruby (or Python) use type inferencing?
    - If yes, explain how they do it along with some simple examples.
    - If not, explain why they do not do so.

Figure 5: Type Inferencing Question for the FB activity

The experiment consisted of two different topics of similar complexity. The forum-based (FB) activity (Figure 5) concerns what is called *type inferencing* in which the compiler for the language automatically deduces some or all of the information about the *type* of a given variable in the program. By contrast, in Java, this information has to be explicitly specified by the programmer. Type inferencing is expected to make programming faster – once the programmer is comfortable with how it works. This activity tries to get students to think about what exactly happens during type inferencing in the Scala system. For comparison, the activity also asks about Ruby and Python which are two languages that are quite different from Java/Scala; Ruby and Python are somewhat similar to each other and most (or all) students are reasonably familiar with at least one of them. The goal of including this part of the activity is to get students to go beyond what was explicitly presented in the class discussion. The correct answers to the two questions in this activity are that, indeed, depending on the situation, a Scala programmer may omit some, all, or none of the type information. And, for the second part, that Ruby and Python don't use type inferencing in the same manner as Scala but there is something vaguely analogous that happens when the Ruby/Python program is executed (while type inferencing in Scala happens *before* the program is executed).

Some people carry polymorphism too far! Specifically, in the example of the Core interpreter, you might have a base class called Node that corresponds to all kinds of nodes that might appear in the abstract parse tree and have classes such as `Prog`, `Decl`, `Stmt`, etc. be derived classes of `Node`. `Node` will contain three abstract methods, `Parse`, `Print`, `Execute`. This is a bad idea because it can lead to some errors not being caught.
Choose one of the listed options and explain your choice.

1. I disagree! OO-polymorphism is a great idea and should be used whenever possible.
2. It is neither good nor bad; it is simply an alternative approach.
3. I agree this is a bad idea; for the precise reasons why, see my explanation.
4. I don't know ... I really don't understand this entire topic. Help!

Figure 6: Polymorphism question for the RB condition

In the rounds-based (RB) condition, a question on polymorphism in object oriented languages was given (Figure 6). *OO polymorphism* is a key feature of object-oriented languages such as *Java*. Polymorphism is indeed a powerful technique and, when used in the appropriate context, it allows us to build very flexible systems while, at the same time, eliminating the possibility of certain kinds of errors caused by mismatched *types* from occurring during system execution; without polymorphism, the programmer is responsible for including detailed code that explicitly checks for these errors during execution and, if found, take suitable corrective action. At the same time, some developers go overboard and apply polymorphism in situations where its use is *inappropriate* with the result that the system becomes, in a sense, *too flexible*; i.e., it allows for the creation of components with structures that do not make sense for the given application. The question is posed in the context of a project the students have worked on throughout the semester, and hence are aware of the design issues.

Both activities were conducted as graded homeworks as part of the regular course and were worth equal points. In Phase-3 for both conditions, students were asked to submit their final answers to the same questions they discussed through the CONSIDER app. For both conditions (FB and RB), the initial and final answers were evaluated on a 4-point scale, and the difference in the two scores was measured. The difference between each student's score for the final answer (post-test) from that for the initial answer (pre-test) gives a measure of improvement in learning. The adjoining table summarizes the pre- and post-test data from this study. A Gain Score Analysis was conducted on the pre- and post-test performance (Gain = Final Submission Score − Initial Submission Score). Shapiro-Wilk normality test on the two variables indicates that the data is not normally distributed. Therefore, a Wilcoxon signed-rank test, a non-parametric equivalent of the one-tailed t-test for within-subject data, is used. The analysis shows that the improvement in learning was significantly higher in the RB activity ($M = 0.62, Mdn = 0.25$) compared to the improvement in learning in the FB activity ($M = 0.27, Mdn = 0$), $r = -.38, p < .05$.

	FB Gain	RB Gain
Mean	0.27	0.62
SD	0.41	0.84
Median	0	0.25
Min	0	−0.5
Max	1	2
Range	1	2.5

5 Conclusions and Future Work

The central theoretical idea underlying our approach is that socio-cognitive conflict between small groups of computing/engineering students in standard undergraduate courses and the attempt to resolve the conflict via discussions among the students (*without* participation by the instructor) can be very effective in helping the students develop deep understanding of the topic. We have developed a scalable, platform-independent web app that implements this approach which we used in a junior level computer science and engineering (CSE) course. We compared our approach against the prevalent forum-based discussions and found a statistically significant difference in students' understanding of the topic discussed using our approach as compared to the prevalent approach.

While approaches based on the theoretical idea mentioned above have been investigated by many researchers starting with the classic work of Piaget on young children to the more recent work by Mazur and others in the development of the *peer instruction* (PI) approach in undergraduate STEM classrooms, our work, as detailed in the paper, extends that work in important ways by exploiting the facilities of online systems. While one advantage of the approach is that reduces or eliminates the need for devoting precious lecture time to the discussions, far more important is that it enables the discussion to be organized in ways that are simply *not* possible in classroom-based approaches such as PI; e.g., the ability to have students participate in the discussions *anonymously*, or the ability to allow a student to *edit* and improve a post she may have previously made before the other students in the group have access to it, etc. are possible only because of our use of appropriate online technologies. At the same time, our work also points in a number of important directions that can make the CONSIDER approach even more effective. In this section, we consider some of these questions and how we plan to address them in our future work.

A natural question that one might ask, especially given the increasing prevalence of online courses, is the possibility of applying the approach to such courses, possibly even to MOOCs. One central difficulty is that, clearly, the approach of forming the small discussion groups "by hand" simply cannot scale to handle such situations. So how do we ensure that groups can always be formed automatically? Consider again the software engineering example from Section 3. The version we used in the detailed description of the CONSIDER system required the student to address only the item related to the "range of visual and cognitive impairments that people suffer from" and specify whether it fell under the category of "domain", "problem" or "solution". In this case, if most of the students in the course chose the same answer as may indeed happen, it is not possible for the system to form groups automatically and the instructor is forced to intervene and form them by-hand. But what if the student were required to address *all* four of the items listed in the original "Homework"; in other words, for each of the four items, the student would have to specify whether the particular item fell under the category of "domain", "problem" or "solution". In this situation, the number of possible answer *combinations* is very large and the chances that most students will pick a particular one of those combinations is very low indeed; it can still happen if the homework is poorly designed, but otherwise it is an unlikely occurrence. So a natural extension of the CONSIDER approach would be to have the students indicate a choice for each of those items[3]. In our future studies, we would like to explore this option which will let the system automatically create groups of students with conflicting answers based on a combination of choices students make on related topics at the beginning of the activity, and allow the system to scale to large classrooms as well.

In conclusion, several of the online argumentation systems developed over past couple of decades (see [1, 12, 10]) are part of larger systems intended to help students, for example in high-school chemistry classes, to engage in collaborative knowledge *construction*, following principles of constructionism. As such, they often include elaborate graphical (and, often, video) facilities to enable students to engage in the necessary experimentation, literature search, etc. The entire course is often designed around the system in question. By contrast, CONSIDER is intended for use in standard undergraduate engineering courses to help students develop deep(er) conceptual understanding of the concepts and topics presented in lectures in the course in the standard fashion. Thus an engineering course that uses CONSIDER will look much like its non-CONSIDER counterpart; the only differences will be that the instructor for the CONSIDER course will post (some of) the "homework assignments" on the CONSIDER web-app; and the students in that course will complete those assignments also on the web-app. Thus, the approach should find acceptance among faculty teaching CSE or other engineering/STEM courses.

References

[1] J Andriessen, M Baker, and D Suthers. *Arguing to learn: Confronting cognitions in "CSCL" environments*. Kluwer, 2003.

[2] I Boticki, L Wong, and C Looi. Designing technology for content-independent collaborative mobile learning. *IEEE Trans. on Learning Technologies*, 6(1):14–24, 2013.

[3] A Brown. Design experiments: Theoretical and methodological challenges in creating complex interventions in classroom settings. *Journal of Learning Sciences*, 2:141–178, 1992.

[4] C Chinn and W Brewer. An empirical test of a taxonomy of responses to anomalous data in science. *Journal of Research in Science Teaching*, 35(6):623–654, 1998.

[5] P Cobb, J Confrey, A deSessa, R Lehrer, and L Schauble. Design experiments in educational research. *Educational Researcher*, 32(1):9–13, 2003.

[6] M Cole. Using wiki technology to support student engagement: Lessons from the trenches. *Computers & Education*, 52:141–146, 2009.

[7] C Crouch and E Mazur. Peer instruction: Ten years of experience and results. *American Journal of Physics*, 69(9):970–977, 2001.

[8] R Driver, P Newton, J Osborne, et al. Establishing the norms of scientific argumentation in classrooms. *Science education*, 84(3):287–312, 2000.

[3] This does raise a concern though. Students' time (and willingness to work on homeworks!) is limited so if we required them to indicate these choices for each item and provide a rationale, etc., they may rebel! One possible way to address this might be to have all four items as part of the initial submission requirement but not for later discussion rounds.

[9] R Dufresne, W Gerace, W Leonard, J Mestre, and L Wenk. Classtalk: A classroom communication system for active learning. *Journal of computing in higher education*, 7(2):3–47, 1996.

[10] K Hew and W Cheung. *Student Participation in Online Discussions*. Springer, 2012.

[11] C Hoadley. Methodological alignment in design-based research. *Educational Psychologist*, 39(4):203–212, 2004.

[12] C Howe and A Tolmie. Productive interaction in the context of computer-supported collaborative learning in science. In *Learning with computers*, pages 24–46. Routledge, 1999.

[13] D Jonassen and S Land. *Theoretical foundations of learning environments*. Routledge, 2012.

[14] T Judd, G Kennedy, and S Cropper. Using wikis for collaborative learning: Assessing collaboration through contribution. *Australasian Journal of Educational Technology*, 26(3):341–354, 2010.

[15] J Larusson and R Alterman. Wikis to support the collaborative part of collaborative learning. *International Journal of Computer-Supported Collaborative Learning*, 4:371–402, 2009.

[16] K Leung and S Chu. Using wikis for collaborative learning: A case study of an undergraduate students' group project. In *Proc. of Int. Conf. on Knowledge Mgmt.*, pages 1–14, 2009.

[17] N Mirza and A Perret-Clermont. *Argumentation and Education*. Springer, 2009.

[18] E Nussbaum. Argumentation in and student-centered learning environments. In *Theoretical foundations of learning environments*, pages 114–140. Routledge, 2012.

[19] J Osborne, S Erduran, and S Simon. Enhancing the quality of argumentation in school science. *Journal of Research in Science Teaching*, 41(10):994–1020, 2004.

[20] J Piaget. *The early growth of logic in the child*. Routledge and Kegan Paul, 1964.

[21] J Rick and M Guzdial. Situating CoWeb: A scholarship of application. *Int. J. of Computer Supported Collaborative Learning*, 2:89–115, 2006.

[22] M Scardamalia and C Bereiter. Technologies for knowledge-building discourse. *Comm. of the ACM*, 36(5):37–41, 1993.

[23] S Singer, N Nielsen, H Schweingruber, et al. *Discipline-based education research: understanding and improving learning in undergraduate science and engineering*. National Academies Press, 2012.

[24] F Wang and M Hannafin. Design-based research and technology-enhanced learning environments. *Educational technology research and development*, 53(4):5–23, 2005.

[25] A Weinberger and F Fischer. A framework to analyze argumentative knowledge construction in computer-supported collaborative learning. *Computers & education*, 46(1):71–95, 2006.

[26] K Yeh and H She. On-line synchronous scientific argumentation learning: Nurturing students' argumentation ability and conceptual change in science context. *Computers & Education*, 55(2):586–602, 2010.

[27] A Zohar and F Nemet. Fostering students' knowledge and argumentation skills through dilemmas in human genetics. *Journal of research in science teaching*, 39(1):35–62, 2002.