PROFESSIONAL DEVELOPMENT THROUGH COLLABORATIVE ANALYSIS OF STUDENT WORK

Andee Rubin
TERC, United States of America
Andee_Rubin@terc.edu

The original goal of this work was to create software support for teachers to use in analyzing the work their students had done using the educational data analysis software TinkerPlots. In pursuit of this goal, we designed three presentations of students' work to be automatically generated for teachers. To evaluate these three presentation styles, we carried out a user test with a group of teachers. The conversation that ensued turned out to be an excellent example of professional development; we will continue to pursue this possibility in our future work.

BACKGROUND

This paper reports on a project whose primary purpose has been to create software support for teachers to use in analyzing the work their students have done using educational data analysis software. The project goal is to help teachers in settings like a computer lab, where they can't follow all of the students' work, and therefore have trouble basing pedagogical planning on students' understanding. Our approach to this issue is to automatically provide organized reports of students' interactions with the software. Much of the work of the project, described in more detail below, has been to design several representations of students' work that we hypothesized would be helpful to teachers. We have conceptualized the process of building a student model as a collaboration between the computer–which can efficiently create representations of student work–and a teacher, who can interpret and assign meaning to these representations and decide on next pedagogical steps. (Babaian et al., 2002).

In order to find out whether or not our work would meet teachers' needs, we performed a "user study" with teachers in which we asked them to discuss what information each representation provided to them – and which they preferred. As the user study progressed, we realized that the conversations teachers were having provided them with an opportunity for professional development. In order to discuss their preferences, they had to consider the structure of the problem, the ways in which students might approach it, and what they could infer from the set of representations. While we do not have extensive experience using this kind of user study as professional development (as of the writing of this paper), we consider in this paper how our work in building representations of student work can support new possibilities for teachers to learn.

The rest of this paper is organized into five sections: Section I introduces TinkerPlots, the educational software that students used in our experiments. Section 2 introduces plan recognition and state-based algorithms, both computer science tools that are useful for building representations. Section 3 presents the representations we built to support teachers in their use of TinkerPlots. Section 4 details a user study conducted with teachers to get feedback on our representations, and Section 5 discusses future work and open questions moving forward.

TINKERPLOTS

TinkerPlots™ (Konold & Miller, 2004) is an educational software system used world-wide to teach students in grades 4 through 8 about statistics and probability. It provides students with a toolkit to actively model stochastic events and to create and investigate a large number of statistical models. As such, it is an extremely flexible application, allowing for data to be modeled, generated, and analyzed in many ways using an open-ended interface. To demonstrate our approaches towards representing student work in TinkerPlots we will use the following example, called RAIN.

RAIN: The probability of rain on any given day is 75%. Use TinkerPlots to compute the probability that it will rain on each of the next four consecutive days.

Because of TinkerPlots's flexibility, there are many ways to use it to solve this problem. Figure 1 illustrates one of the most common approaches.

In C. Reading (Ed.), *Data and context in statistics education: Towards an evidence-based society. Proceedings of the Eighth International Conference on Teaching Statistics (ICOTS8, July, 2010), Ljubljana, Slovenia*. Voorburg, The Netherlands: International Statistical Institute. www.stat.auckland.ac.nz/~iase/publications.php

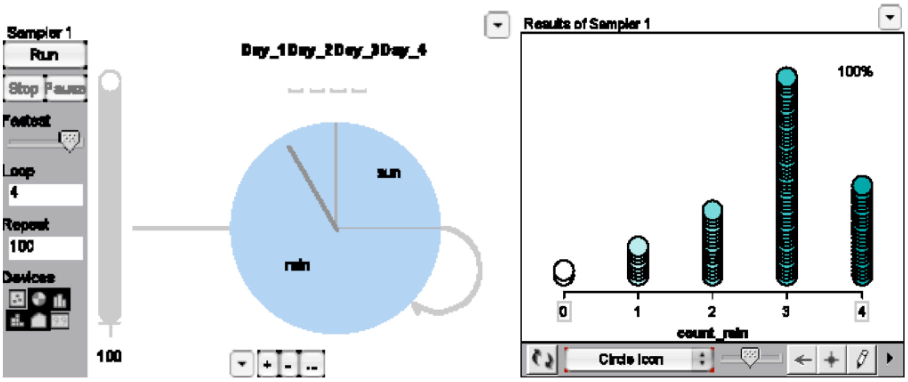

Figure 1. Using a TinkerPlots sampler to solve the RAIN problem

The left half of this figure is a stochastic device (a spinner) that is 75% rain and 25% sun. To estimate the probability of rain on four consecutive days, we spin the spinner four times (notice Loop 4 to the left of the spinner) to generate a series of four days. We repeat this sequence of spins 100 times and sort the results according to the number of rainy days. The right half of the figure is a graph of sequences of four days that have 0, 1, 2, 3 and 4 rainy days respectively. The probability of a sequence of four rainy days is the ratio of the height of the "4" column to the total number of sequences of 4 days. The other common way to approach this problem is to model a series of days as a sequence of four spinners, each of which is 75% rain and 25% sun.

PLAN RECOGNITION AND STATE-BASED ALGORITHMS

The main thrust of our original project was to design approaches to analyzing students' work based on the sequence of computer actions they carried out. These actions are automatically captured in a computer log, which we used as the data for our approaches. Our work focused on two main approaches: plan recognition and state-based algorithms.

The first type of algorithms that we applied fall under an area of artificial intelligence called plan recognition. (Lesh et al., 1999; Carberry, 1988). Plan recognition algorithms infer the plan of the agent given observations, or basic user actions, and a database of recipes. A recipe for a complex action includes a set of (complex and basic) sub-actions and restrictions for completing the complex action. There are often several recipes to accomplish the same goal. A plan is a set of recipes, arranged hierarchically, that accomplish the overall goal.

Recognizing students' plans using TinkerPlots, however, is a much more complex task than the traditional plan recognition algorithms tackle, so simply applying the traditional algorithms was nowhere near sufficient. Though we were able to extend traditional approaches to be robust to the types of complex behaviors demonstrated by students, we found that such recognition came at a cost: successful algorithms were computationally complex (i.e., very slow in the worst case) and assumed comprehensive recipe libraries, an impractical requirement.

In order to address the weaknesses experienced with plan recognition, we turned to a different variety of algorithms, called state-based algorithms, which use the log of users' actions to determine what is visible on the screen at any given time. Compared to plan recognition, these algorithms are computationally fast and accurate. However, these algorithms cannot infer meaning, as can plan recognition algorithms. Our hypothesis has been that teachers can infer meaning from the output of these algorithms, though they may need to spend more time analyzing student methods than in the plan recognition case.

PRESENTATION STYLES FOR STUDENTS' WORK

We created three presentation styles for students' work, one based on plan recognition and two based on state-based algorithms, which were built on images of the students' screen at selected times. (Gal et al., 2008; Reddy et al., 2009)

The *plans* presentation uses output from plan recognition algorithms to inform teachers whether a student has solved a problem, as shown in Figure 2. The student was able to complete part of the problem (shown in black), but left several parts undone (shown in red). For example, she created the correct spinner, but didn't use an appropriate number of repetitions. She created a new plot, but didn't add any points to it. All in all, this student performed many appropriate and correct actions, but was unable to finish the problem.

- Solve RAIN problem
  - Create correct sampler
    - Add new sampler [1]
    - Set draws to 4 [7]
    - Set repetitions to 100 [--]
    - Create correct device
      - Create "rain" element
        - Add element [1]
        - Change element label to "rain" [2]
      - Create "sun" element
        - Add element [1]
        - Change element label to "sun" [3]
      - Change probability of "rain" and "sun from 50-50 to 25-75 [8]
  - Generate Data [--]
  - Estimate Probability
    - Add new table [5]
    - Create "Outcome" attribute [6]
    - Add new plot [4]
    - Drag "Outcome" attribute to y-axis [--]
    - Add tallies to plot [--]

Figure 2. A *plans* representation of the work of a student who did not solve the RAIN problem

The other two presentation styles we created were based on the state-based algorithms that kept track of the TinkerPlots images on the screen as a student used the software. The *focal objects shifts* presentation tracks which software object the student is currently manipulating and prints the state of the screen every time the student selects a different object to work with. We don't know, of course, what the student was actually focusing on, but tracking which object is selected provides a glimpse of the students' process. Figure 3 is a focal objects shift representation of a student working on the same problem.

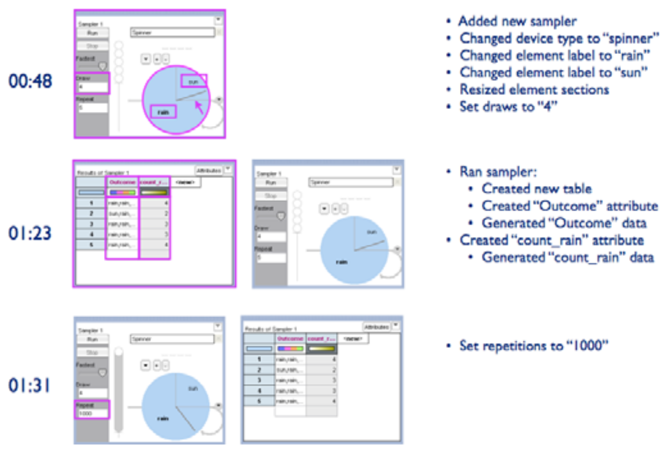

Figure 3. A *focal objects shifts* representation of a student working on the RAIN problem

In Figure 3, the student was first working on the spinner. After 48 seconds, he ran the sampler, which created and selected a table of results, which then became the focal object. At 1 minute and 23 seconds, he returned to work on the sampler by increasing the number of repetitions to 1000. At 1:31 he moved on again. In each screen snapshot, the parts of the object that have been modified are outlined in purple and on the right of the screen images is a description of the actions that happened during the period the student was working on the focal object. In this case, during the time that the table was the focal object, the student ran the sampler, generating data for both the Outcome attribute and the Count_rain attribute. This representation shows the steps the student took to solve the problem, in chronological order and with the amount of time each step took. However, these state-based of representations fail to arrange the long list of student actions into a coherent plan and it is left to the teacher to piece together how a student arrived at whatever answer he recorded.

The third presentation style, *selected snapshots*, was similar to the second in that it used screen images in chronological order. However, instead of displaying the screen image when the student moved on to a different object, this presentation style just included every fifth screen image, i.e., every fifth time the screen changed at all.

USER STUDY WITH TEACHERS

To assess our representations, we conducted a user study with nine teachers in Amherst, Massachusetts. Our goal was to study teachers' preferences for the plans, selected snapshots, and focal object shifts presentation styles. All participants were middle or high school teachers in day five of a five-day summer workshop on TinkerPlots. Most of the participants had not used TinkerPlots extensively before the workshop, but intended to use it in the current school year.

The presentations we used in the user study depicted real student sessions that we observed in June 2009 at the Amigos school, a bilingual Spanish-English immersion school in Cambridge, MA. For three days, we worked with the same 8^{th} grade mathematics class, with an average class size of 18, for at least 45 minutes. Each day, we began with a demonstration of unfamiliar TinkerPlots features and then asked students, mostly in pairs, to solve up to four worksheet problems using TinkerPlots. All student actions were recorded by both screenshot videos and a software logging facility. Presentations were based on data collected from students' work on the ROSA problem (described below), which occurred on day three.

The ROSA problem: Jessica has 4 letters printed on cards as shown below:

| A | O | R | S |

After mixing them up, she blindly picks the 4 letters one at a time and arranges them in line in the order she chose them. What is the probability that she chooses R, O, S, A in that order, spelling ROSA.

The ROSA problem shares many characteristics with the RAIN problem; it requires creating a sampler with four elements, drawing four times from the sampler and examining the results. The primary difference between the two is that the ROSA problem requires the student to do the sampling without replacement. This proved to be a stumbling block for some students.

The user study was divided into three parts. In Part 1, we introduced all three presentation styles to the teachers, using one student's log. In Part 2, we asked the teachers to fill out an online survey (http://www.keysurvey.com/survey/269620/test/). For the first part of the survey, teachers saw Alices's log in presentation style 1, Blair's log in presentation style 2, and Cameron's log in presentation style 3. (All student names are pseudonyms.) After viewing each log, teachers responded to the following questions:

- Did the student solve the ROSA problem? How do you know?
- How proficient is the student with TinkerPlots? How do you know?
- What kinds of mathematical misunderstandings did the student have? How do you know?

In the second part of the survey, the teachers saw Devon's and Erin's logs in all three presentation styles. After each set of presentations, teachers were asked to rate and comment on the following:

- This presentation style clearly demonstrates whether and how the student solved the ROSA problem.
- This presentation style clearly demonstrates whether or not the student is proficient with TinkerPlots.

- This presentation style clearly demonstrates which types of mathematical misunderstandings the student has, if any.
- How likely would you be to use this presentation style to understand a student's work after a classroom TinkerPlots session?
-

In Part 3, teachers participated in a free-form, 25 minute discussion, where they voiced their positive and negative reactions to each of the presentation styles. We report here the results of both the online survey and the open discussion. One of the most salient results of the survey is the contrast between the first and second questions, as shown in Figure 4.

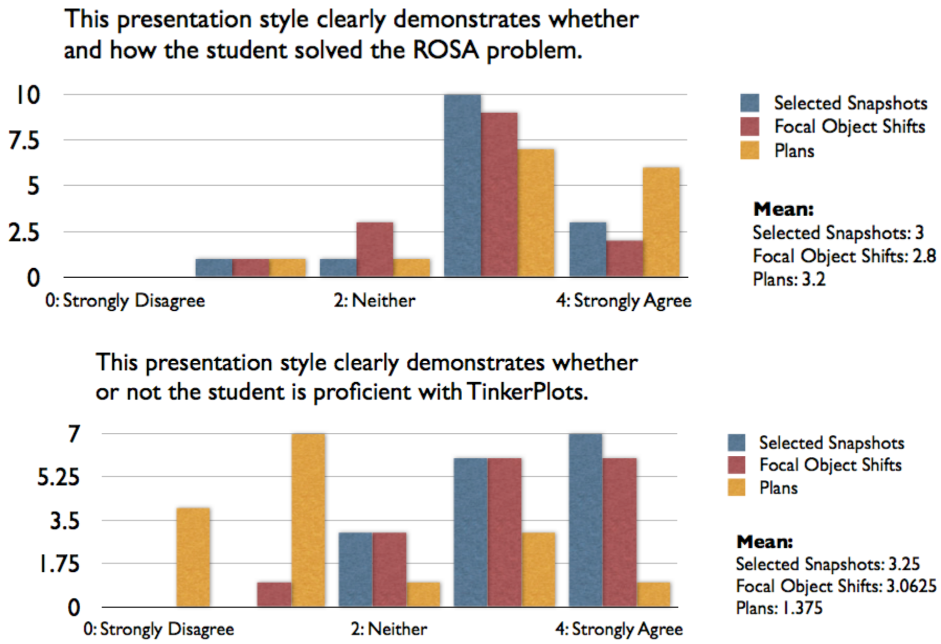

Figure 4. Teachers' responses to two questions after viewing all three presentation styles

The teachers in the study showed a strong preference for the plans presentation if their purpose were to figure out whether or how the student solved the ROSA problem. However, if instead they were interested in learning how proficient the student was with TinkerPlots or what mathematical misunderstandings their students might manifest, the teachers greatly preferred one of the two state-based presentations.

The open discussion brought out more interesting points. Most teachers felt that the plan presentation was the easiest to read and several voiced that they would not usually have the time to analyze the longer presentation styles, useful though they may be. In this vein, one teacher said, "With the [selected snapshots and attention shifts], what I eventually got to was tired of trying to dig through to find anything out other than did they solve it… It took too much work." The majority of teachers voiced a preference to begin with the plan for an overview and then move to the other presentations to get a more detailed idea of the student's work. One teacher said: "The plan gave you a quick synopsis -- did they get it? The selected snapshots gave me more of their process. And, I would probably use that intermittently. I wouldn't use it all the time. Because, when you have 85 to 100 kids, you don't have the time to do that."

However, not all teachers thought the plan presentation was useful. One teacher described herself as a "visual person" and felt that the selected snapshots "gave me better information faster." Another teacher commented on the lack of information provided by the plan: "I liked [the plan] for a quick glance for did they get all the steps? … But, to get the process… I don't get an insight as to what they were thinking of as they were working through the problem. … And what else did they do that was not successful? I have no idea."

Several teachers proposed methods of changing or adding to the information provided by the plan through interactive drop-down menus and other options. One teacher proposed, "Starting from the plan, because obviously there is a lot of information in the attention shifts and snapshot presentations…One thing I was wondering is if you could expand those steps to show maybe how many things or show what things related to that concept they did?" A second teacher proposed a presentation that allowed the teacher to "view [the student's actions] relative to the template but also view them as a list of ordered steps and sort of align but be able to expose what they did in between." Many teachers also expressed interest in aggregate reporting, which was consistent with the previous notion of creating zoom in and zoom out capabilities. The student responses might be aggregated, for example, by whether or not they had chosen "without replacement" or by the largest number of repetitions they tried. Then teachers would be able to zoom in on a particular student from the aggregate list in order to see more detail on what she or he did.

As the conversation progressed, the teachers began to focus on what they could find out about how much a particular student understood. One said, "The mathematical confusion, you could tell in different people.., you could tell in the graph if they have the same variable on both axes that wasn't going to help them. That's definitely mathematical confusion. And they could play with TinkerPlots all day long but they're not going to get what they want." Another commented, "I think the thing that I tracked the difference between mathematical confusion versus unfamiliarity with the program the best was the replacement idea. Because I think it's the most advanced concept in the task that's being asked. And it's also the least obvious thing they had to do in the program. Because it's not right out there the way that a lot of the graphics stuff is. You have to look for it. And there are several choices there. So I think depending on how far into the program they went for that, I saw that as a mark of proficiency with the program. And how many times they switched it seemed to be an indication of how comfortable they were with the mathematical concept."

Notice that in this conversation the participants are making comments that require knowledge of TinkerPlots, the mathematics students were exploring, and the interaction between the two. These are just the kind of conversations we want teachers to have in professional development settings. Our purpose in carrying out the user study was not to deepen teachers' understanding of how students might use TinkerPlots to build up and create evidence of statistical reasoning. But the topic and the setting turned out to be powerful supports for teachers to pursue those topics. We have created a new professional development approach, even though we didn't set out to do so.

CONCLUSION: LOOKING TO THE FUTURE

In the future, we will explore interactive methods for presenting aggregate data on students' actions to teachers and determine which aspects of students' work are most useful to aggregate. Given our recent experience with teachers, we will continue to study the professional development potential, an unanticipated bonus of our work.

ACKNOWLEDGEMENTS

This work was supported by NSF grant number REC-0632544.

REFERENCES

Babaian, T., Grosz, B. J., & Shieber, S. M. (2002, January). A writer's collaborative assistant. In *Proceedings of the Intelligent User Interfaces Conference* (pp. 7-14), San Francisco CA: ACM Press.

Carberry, S. (1988). Modeling the user's plans and goals. *Computational Linguistics*, *14*(3), 23-37.

Gal, Y., Yamagil, E., Shieber, S., Rubin, A., & Grosz, B. (2008). Towards collaborative intelligent tutors: Automated recognition of users' strategies. In *Proceeding of the 9th International Conference on Intelligent Tutoring Systems*, Montreal, Canada.

Konold, C., & Miller, C. (2004). *TinkerPlots™ Dynamic Data Exploration* (v1.0). Emeryville, California: Key Curriculum Press.

Lesh, N., Rich, C., & Sidner, C. L. (1999). Using plan recognition in human-computer collaboration. In UM '99: *Proceedings of the seventh international conference on User modeling* (pp. 23-32), Secaucus NJ, USA.

Reddy, S., Gal, Y., & Shieber, S (2009). Recognition of users' activities using constraint satisfaction. In *Proceeding of the 1st and 17th International Conference on User Modeling, Adaptation, and Personalization*, Trento, Italy, June 2009.