Chapter 13
Classroom Uses for BeSocratic

Sam Bryfczynski, Roy P. Pargas, Melanie M. Cooper, Michael Klymkowsky, Josiah Hester, and Nathaniel P. Grove

Abstract This paper describes how BeSocratic can be used to improve learning and class interaction. BeSocratic is a novel intelligent tutoring system that aims to fill the gap between simple multiple-choice systems and free-response systems. The system includes a set of interactive modules that provide instructors with powerful tools to assess student performance. Beyond text boxes and multiple-choice questions, BeSocratic contains several feedback driven modules that can capture free-form student drawings. These drawings can be automatically recognized and evaluated as complex structures including Euclidean graphs, chemistry molecules, computer science graphs, or simple drawings for use within science, technology, engineering, and mathematics courses. This paper describes three use-cases for BeSocratic and how each scenario can improve learning and class interaction throughout the curriculum. These scenarios are: (1) formative assessments and tutorials, (2) free-response exercises, and (3) in-class real-time activities.

S. Bryfczynski (✉) · R. P. Pargas · M. M. Cooper · J. Hester
Clemson University, Clemson, SC, USA
e-mail: sbryfcz@clemson.edu

R. P. Pargas
e-mail: pargas@clemson.edu

M. M. Cooper
e-mail: cmelani@clemson.edu

J. Hester
e-mail: jhester@clemson.edu

M. Klymkowsky
University of Colorado, Boulder, CO, USA
e-mail: michael.klymkowsky@colorado.edu

N. P. Grove
University of North Carolina at Wilmington, Wilmington, NC, USA
e-mail: nathaniel.grove@uncw.edu

T. Hammond et al. (eds.), *The Impact of Pen and Touch Technology on Education,*
Human-Computer Interaction Series, DOI 10.1007/978-3-319-15594-4_13

13.1 Problem Statement and Context

Since the early days of personal computing, software has been developed for educational purposes. The number of such applications continues to increase, and the sophistication of the systems is constantly evolving. There exists a wide spectrum of educational software meeting needs from Pre-K to industry.

Today, most higher education institutions use broad learning management systems such as Blackboard, Moodle, or Instructure Canvas to aid in assessment. Additionally, specialized systems (such as the Mastering software series, OWL, etc.) exist for individual disciplines and courses. A subset of these systems includes intelligent tutoring software such as MathTutor and CogTutor, which provide students with step-by-step guidance during problem solving. While these systems have been shown to enhance student learning in a range of domains [4–6], they tend to be difficult to author, and the majority of questions they can ask fall into one of two categories: free-response text-based questions or multiple-choice/matching questions. Free-response systems allow teachers to ask meaningful questions that require students to have a deep understanding of the subject in order to answer correctly; unfortunately, they are difficult to assess quickly and without bias. Multiple choice and matching questions are more restrictive by nature, and research has suggested that these questions cannot be used to properly assess deep knowledge on a subject since the exercises often only involve memorization [2, 3, 7, 8].

Which system a teacher uses generally depends on the amount of time available to evaluate student solutions. Free response systems require teachers to manually check each submission; this is too time-consuming for teachers to perform on a regular basis. Because of this, teachers instead rely on using multiple-choice or matching questions that can be quickly or automatically evaluated. The ideal system combines the best parts of both question types: for example a teacher would be able to pose questions that require students to have a deep understanding of the material and allows students to reply in an intuitive free-form manner. Upon completion of activities, student responses would be automatically evaluated, analyzed, and made available to the teacher.

13.2 Method Employed

With this motivation, we built BeSocratic, an online intelligent tutoring system that contains a variety of question types that are free-form in nature yet well-defined to the point in which they may be automatically evaluated and analyzed [1]. BeSocratic's interactive tutors are referred to as *activities*. Each activity is made up of one or more slides. And just as slides in a slide-show have various content elements within them (e.g., text, images, videos), BeSocratic slides contain one or more *modules*.

BeSocratic modules are divided into two general categories: *non-interactive* and *interactive*. Non-interactive modules, which include text boxes, images, videos, ink canvases, and 3D molecular models do not provide students with feedback. These

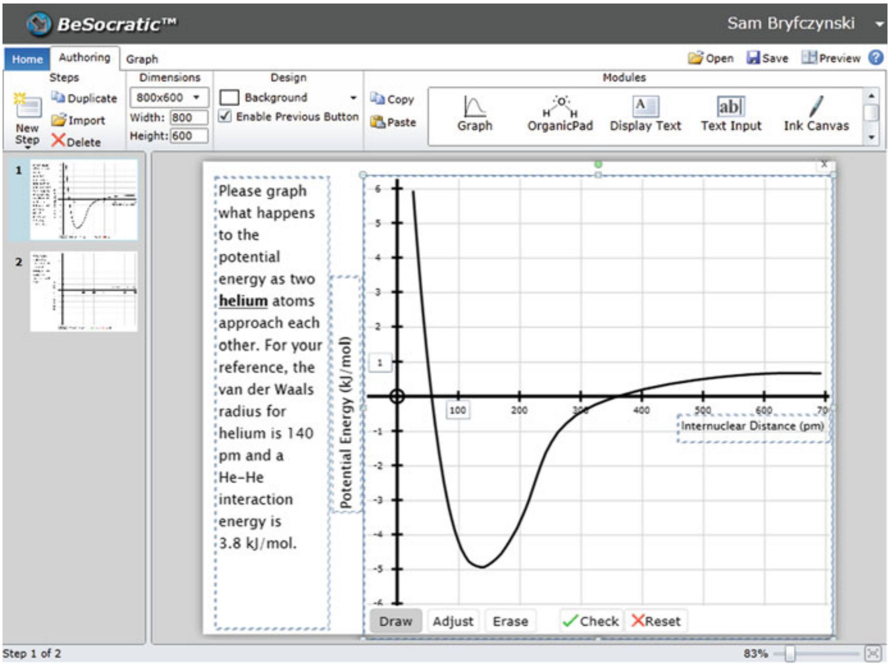

Fig. 13.1 BeSocratic's activity authoring tool

are generally used to convey information and/or instructions to students or are used to gather information for manual analysis. Interactive modules, on the other hand, allow BeSocratic to pose free-form questions, provide automatic response-driven feedback to students, and enable automatic analysis for teachers. These modules currently include SocraticGraphs, OrganicPad, and GraphPad. All of these modules may be used together within slides to build rich, interactive activities.

Teachers construct activities from the modules in BeSocratic's Activity Authoring Tool. Since it has been shown that tutor authoring is often a difficult and time-consuming task, BeSocratic's Activity Authoring Tool has been designed to resemble Microsoft PowerPoint using the Microsoft Ribbon interface, as seen in Fig. 13.1. Similar to PowerPoint, modules are added to activities by dragging a module from a list onto a slide; then teachers can customize the look and behavior of the module by changing various options. The Authoring Tool also contains a preview function for the teacher to preview how the activity will appear to the students. This allows teachers to quickly prototype various configurations for the activity.

Once an activity is created, teachers specify a roster along with start and end dates so that students in the roster may log into BeSocratic and complete the activities within the time allotted. Activities and their student data are stored in the BeSocratic database so the teacher may access them from any computer's browser connected to the Internet. After a teacher makes an activity available, students can log into

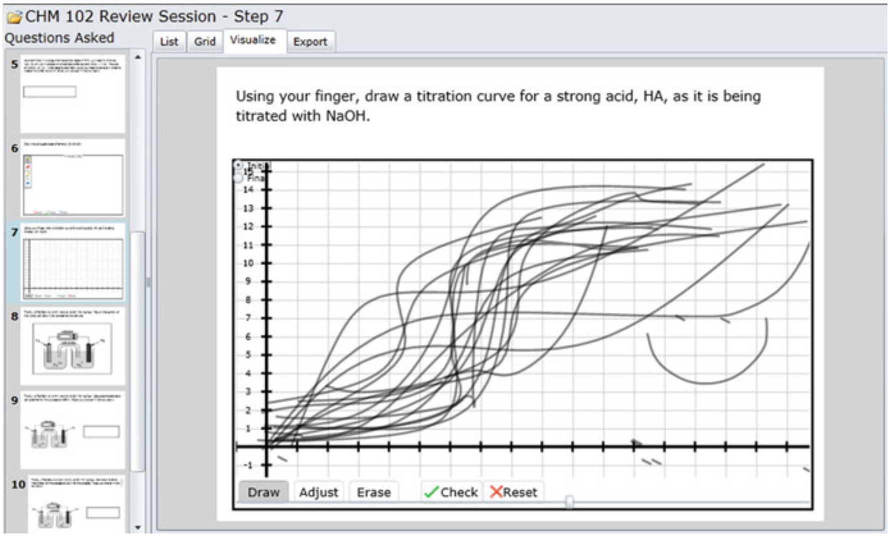

Fig. 13.2 Visualization of many student graphs drawings

the system and complete the activity. As students complete the exercise, BeSocratic records each action that is performed in the system. This information is stored to the database upon completion of each slide for analysis later. BeSocratic contains a set of powerful analysis tools that allow teachers to view, replay, and visualize individual student submissions as well as groups of student submissions as seen in Fig. 13.2.

BeSocratic currently runs on a variety of devices with varying levels of functionality. The activity authoring and analysis tools requires the Microsoft Silverlight plug-in to run inside of most common browsers (e.g., Internet Explorer, Chrome, Firefox, and Safari) on Windows and Mac computers. In its current form, BeSocratic is targeted for use with Tablet PCs and touchscreens; however, it is by no means necessary to use either. In addition, we have also developed a prototype iOS application. A screenshot of the app is shown in Fig. 13.3. This application only contains the student's viewing tool. We feel that because of the fine control needed to author a BeSocratic activity, iOS devices are not yet appropriate for tutor authoring.

The next three sections detail the primary ways teachers have been using BeSocratic: (1) formative assessments and tutorials, (2) free-response exercises, and (3) in-class real-time activities.

13.2.1 Formative Assessments and Tutorials

The primary goal of BeSocratic is the creation of intelligent tutors with the most natural interface possible. We are able to achieve this goal using our interactive modules. They enable the construction of representations (such as Euclidean graphs,

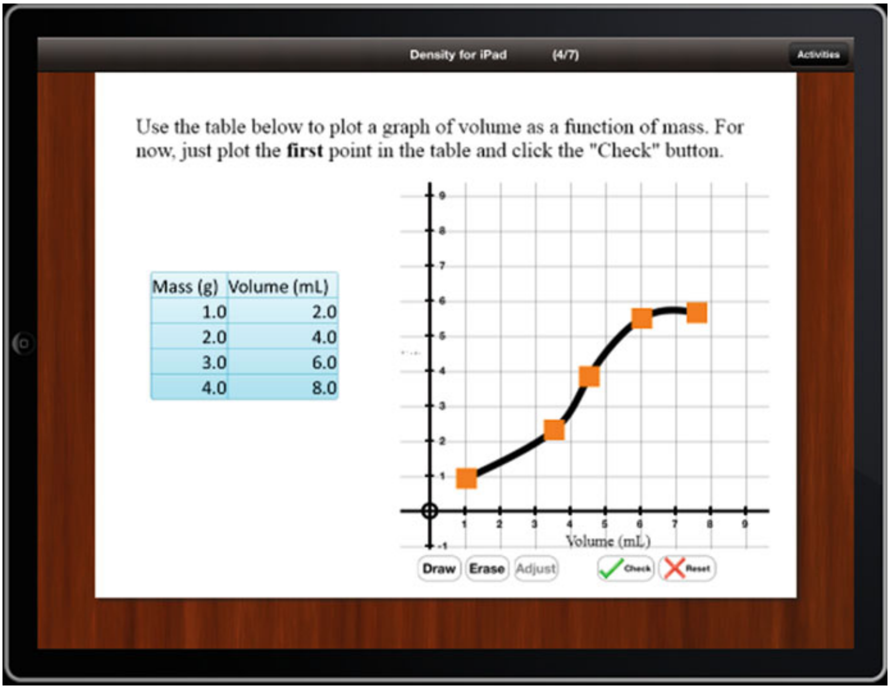

Fig. 13.3 Student completing a question on the iPad

organic chemistry molecules, computer science graphs, and simple drawings) using free-form input (via pen, finger, or mouse) while still being able to respond with tiered contextual, Socratic feedback. Teachers compose this hierarchy of rules (with associated feedback) visually, trying to address different failure areas as well as gently guiding students toward a correct understanding. The end result of this work is an effective, formative assessment system that can serve a wide range of student populations in many STEM disciplines.

We have created many formative assessment and tutorial activities using a process similar to that shown in Fig. 13.4. To start, a topic is selected with the intent to improve or assess student understanding. Once the topic for the activity has been chosen, the first step in the construction process is to interview students in order to determine specific problem areas and identify gaps in understanding. The information obtained from these interviews is used to build the initial version of the activity. This preliminary version is then tested using one-on-one interviews with students to observe how they interact and navigate through the activity; from these observations further refinements are made to address the students' comments and feedback. The revised version of the activity is then pilot tested with approximately 20–40 students to evaluate how a larger group of students progress through the activity. This is an attempt to determine if there are any issues that have not been previously addressed. Once any problematic areas have been worked out, the activity is administered on a

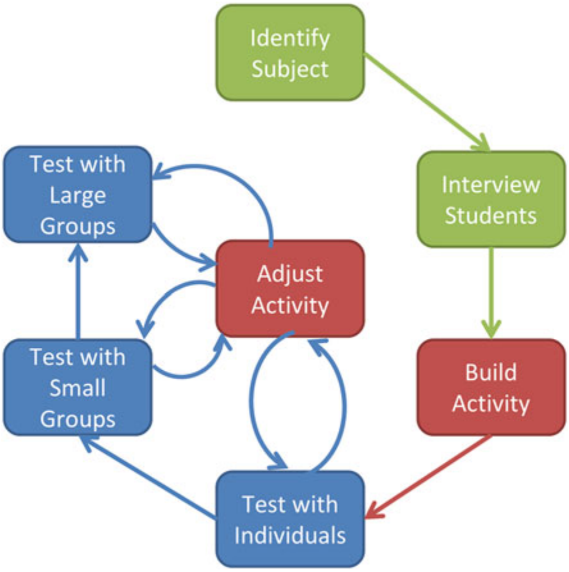

Fig. 13.4 Typical BeSocratic assessment activity creation process

large scale with approximately 100–200 students. Following this process, we have developed many rich, formative assessment activities for introductory chemistry, molecular biology, and computer science courses.

13.2.2 Free Response Exercises

While the formative assessments provide students with a high level of interactivity, we recognize that they are time-consuming to construct and test. Constructing an effective hierarchy of rules paired with feedback requires a lot of time, energy, and piloting. Often teachers want to quickly create an activity that asks students questions that don't necessarily require Socratic feedback. We have been doing exactly that by frequently having students complete short free-response homework assignments. Using just the non-interactive modules, teachers can rapidly author and deploy interesting homework questions. While these activities cannot be graded automatically, teachers can project interesting student answers (anonymously) from the homework assignment as a means to teach by example. While other systems are capable of similar features, BeSocratic has a greater number of customizations that can be utilized to tailor the interactivity of the assignments. In particular, we find that teachers are asking many ink-based questions which require students to answer questions through drawings. Teachers can then project the student submissions during lecture and have discussions on the merits of various answers. In our opinion, having students discuss answers that they have generated themselves leads to improved student engagement.

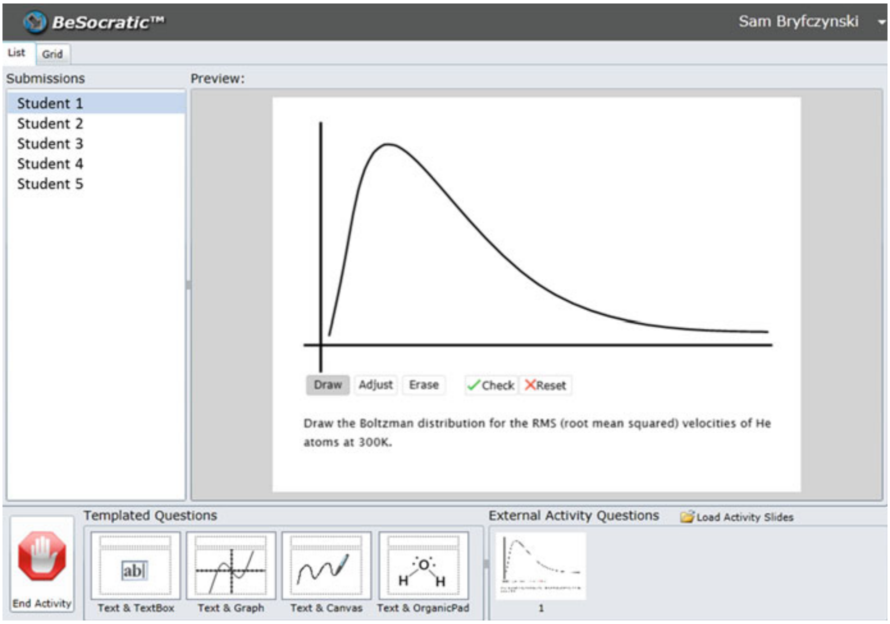

Fig. 13.5 BeSocratic's uRespond system

13.2.3 uRespond Classroom Activities

The functionality described so far has been asynchronous in that teachers create activities, student complete activities and then teachers analyze the student submissions; however, BeSocratic also contains a system called uRespond which provides a way for teachers to interact with students in real-time. This work is part of collaboration with the University of North Carolina at Wilmington. In a manner similar to Clicker systems, uRespond's goal is to allow teachers to ask BeSocratic questions and receive real-time responses from students. This allows teachers to identify misunderstandings and address them in class instead of having to wait until tests or assignments to identify and fix problems.

Internally, the uRespond system works in a similar fashion to the standard BeSocratic activity model. Teachers start by initializing a uRespond session with a name, save location, and roster. This starts an initially empty activity. Once students have loaded the blank activity associated with the uRespond activity, the teacher may either send students previously created activity steps from other activities or send predefined template questions. Figure 13.5 displays an example of the teacher's view during a uRespond session. Template questions and previously-created activity steps are shown along the bottom of the figure. Clicking one will send the question to all the students (Fig 13.6).

Fig. 13.6 Image taken during a trial of a BeSocratic activity in a chemistry class. This trial included 93 students in general chemistry

Students respond using the same activity completion tool that was previously described. When a student responds, a new entry into the submission table is added with the student's user key, replay and final submission. The teacher's uRespond client periodically polls the database for any new submissions for the current question. If new submissions are found, the teacher's view is updated with the new submissions as seen along the left side of Fig. 13.4. The student's activity also polls the database for additional questions added by the teacher. If a new question is found, it is downloaded and presented to the student.

uRespond is a powerful teaching tool that allows teachers to be extremely flexible. By flexible, we mean that questions during a uRespond session can be mixed and matched depending on the feedback received from students. For example, teachers may have a series of planned questions that they create in the authoring tool ahead of class. As they send students these questions and start seeing the answers that students respond with, the teacher may want to change the plan and ask a follow-up question that had not initially been planned on. This is where the teachers may select from a template question, such as a question with text and an ink canvas where teachers ask students to draw something. Alternatively, teachers can load an appropriate question from another activity or quickly create a new question altogether with the authoring tool. Using uRespond in this manner, teachers can control the flow of class by recognizing difficult concepts and immediately addressing them.

13.3 Results and Evaluation

BeSocratic has been implemented in chemistry, molecular biology, and computer science courses in three different universities (Clemson University, the University of Colorado Boulder, and the University of North Carolina at Wilmington) and has collected over 200,000 student submissions. Instructors are using BeSocratic inside the classroom with lecture sizes sometimes exceeding 100 students. We are currently pilot-testing a variety of formal assessment and tutorial activities. Several of the pilot tests show positive learning gains with students; however, further testing and refinement is required to verify the results. In addition to these assessment activities, we are testing BeSocratic's effectiveness as a homework management system in several chemistry courses at Clemson University. We have found that the ability to quickly and easily record and project student drawings without the need to collect paper-based assignments is particularly appealing. uRespond is undergoing pilot-tests at the University of North Carolina Wilmington where it has been used with students during review sessions before chemistry exams.

13.4 Future Work

Our current primary focus is evaluating the effectiveness of the activities we have created. Further testing is required before results can be properly reported. So far, BeSocratic activities have primarily been created for chemistry, molecular biology, and computer science classes. We have begun developing activities for physics and mathematics courses as well and believe that properly developed exercises could improve student learning in these disciplines.

As mentioned in Sect. 3, BeSocratic was created with the Microsoft Silverlight framework. Unfortunately, Microsoft has recently indicated that it will not be updating Silverlight in the future. Because of this we have been actively developing a standards based, HTML5 web application to emulate portions of the BeSocratic system. We believe moving to the HTML5 platform will ensure BeSocratic's future on the plugin-less web; especially since the technologies that comprise HTML5 are nearly universally supported on all devices by their native browsers. Emerging devices such as the Apple iPad, Android Tablet, Kindle Fire, and Microsoft Surface each use a different software development kit yet each of these devices has a built-in web browser capable of running HTML5 applications.

Acknowledgements We would like to acknowledge and thank the NSF for providing funding for this project (TUES-1043707, TUES-1122472). We would also like to acknowledge our collaborators at Clemson University, the University of Colorado at Boulder, and the University of North Carolina at Wilmington.

References

1. Bryfczynski, S., Pargas, R., Cooper, M., & Klymkowsky, M. (2012). *BeSocratic: Graphically assessing student knowledge*. IADIS International Conference on Mobile Learning. Berlin, Germany.
2. Glaser, R. (1988). *Cognitive and environmental perspectives on assessing achievement*. In Assessment in the service of learning ETS Invitational Conference, Princeton.
3. Glaser, R. (1991). Expertise and assessment. In M. Wittrock & E. Baker (Eds.), *Testing and cognition*. Englewood Cliffs: Prentice-Hall.
4. Koedinger, K. R., Anderson, J. R., Hadley, W. H., & Mark, M. A. (1997). Intelligent tutoring goes to school in the big city. *International Journal of Artificial Intelligence in Education, 8*, 30–43.
5. Lesgold, A., Lajoie, S., Bunzo, M., & Eggan, G. (1992). Sherlock: A coached practice environment for an electronics troubleshooting job. In J. Larkin & R. Chabay (Eds.), *Computer assisted instruction and intelligent tutoring systems: Shared goals and complementary approaches* (pp. 201–238). Hillslade: Erlbaum.
6. Mark, M., & Greer, J. E. (1995). The vcr tutor: Effective instruction for device operation. *The Journal of the Learning Sciences, 4*(2), 209–246.
7. Resnick, L. B., & Resnick, D. P. (1992). Assessing the thinking curriculum: New tools for education reform. In B. R. Gifford & M. C. O'Connor (Eds.), *Changing assessments: Alternative views of aptitude, achievement and instruction* (pp. 37–76). Boston: Kluwer Academic.
8. Shepard, L. (1991). Interview on assessment issues with Lorrie Shepard. *Educational Researcher, 20*(2), 21–23.