Leveraging Trends in Student Interaction to Enhance the Effectiveness of Sketch-Based Educational Software

Seth Polsley✉, Jaideep Ray, Trevor Nelligan, Michael Helms, Julie Linsey, and Tracy Hammond

Abstract With the rapid adoption of software-based learning in classrooms, it is increasingly important to design more intelligent educational software, a goal of the emerging field of educational data mining. In this work, we analyze student activities from using a learning tool for engineers, Mechanix, in order to find trends that may be used to make the software a better tutor, combining its natural, sketch-based input with intelligent, experience-based feedback. We see a significant correlation between student performance and the amount of time they work on a problem before submitting; students who attempt to "game" the system by submitting their results too often perform worse than those who work longer (p ¡ 0.05). We also found significance in the number of times a student attempted a problem before moving on, with a strong correlation between being willing to switch among problems and better performance (p ¡ 0.05). Overall, we find that student trends like these could

Seth Polsley✉
Sketch Recognition Lab, Computer Science & Engineering Department, Texas A&M University, College Station, Texas, USA, e-mail: spolsley@tamu.edu

Jaideep Ray
Sketch Recognition Lab, Computer Science & Engineering Department, Texas A&M University, College Station, Texas, USA, e-mail: jaideep.ray@tamu.edu

Trevor Nelligan
Sketch Recognition Lab, Computer Science & Engineering Department, Texas A&M University, College Station, Texas, USA, e-mail: trevornelligan@tamu.edu

Michael Helms
iDreem Lab at Georgia Institute of Technology, Atlanta, Georgia, USA, e-mail: mhelms3@gatech.edu

Julie Linsey
iDreem Lab at Georgia Institute of Technology, Atlanta, Georgia, USA, e-mail: julie.linsey@me.gatech.edu

Tracy Hammond
Sketch Recognition Lab, Computer Science & Engineering Department, Texas A&M University, College Station, Texas, USA, e-mail: hammond@tamu.edu

be paired with machine learning techniques to make more intelligent educational tools.

1 Introduction

Analyzing the trends of student interaction in sketch-based educational software may be the key to building tools that understand students better. As online learning is being adopted by an increasing number of universities [13], designing software that students can understand is important in keeping students engaged. Sketch recognition in these teaching tools may help alleviate that concern by providing an experience similar to the pen-and-paper approaches that students are already comfortable using. However, beyond helping students understand and use educational software, there is an important opportunity that should not be overlooked – each student is individually interacting with the software at some point in the course. This one-on-one interaction provides the chance to do more than just administer and evaluate work, as many online learning tools are designed to do. By building the software with a better understanding of student behavior, it can be used as a more effective tutor, providing feedback on their progress, guidelines to help strengthen their problem-solving skills, and recommended courses of action when they are stuck. There is a wealth of student data available today from existing learning systems, and exploring that information for trends in student behavior and interaction can yield better approaches for developing educational software of all kinds.

There is a strong impetus behind creating better software learning tools. Managing large classes at the university level can be very difficult for departments seeking to reduce cost. This is especially true in STEM fields, where the material may be complicated and finding qualified teachers and graders is hard and expensive. In STEM courses, where drawing and sketching are necessary steps in many classes, sketch recognition can be a good alternative to standard true-false, fill-in-the-blank, or multiple choice software because of its capability to interpret student input throughout a complete problem. Mechanix, a sketch-based tutor for helping students learn about trusses and free body diagrams, was designed for engineering students for this reason [22] [14].

While sketch recognition may be a significant step in making software that students will comfortably use, there is still enormous potential for improving the software. Even when they do not realize it, students follow trends of behavior in completing assignments. Whether they start each problem and work on them simultaneously, go through each problem quickly and ask for help frequently, or don't even complete half of the assignment, their patterns of activity can be monitored in software in the same way a human tutor might. By applying some decision algorithms to this data, educational tools could more closely mimic the intelligence of a human helper and offer students a better experience by avoiding potential problems.

2 Previous Work

Online learning is not new, but it continues to grow significantly as the number of Internet users worldwide increases[1] [1]. Making software tools that interact naturally with students is crucial in order to accommodate the diverse array of individuals using them. As tablets and touchscreens have become more commonplace, input technologies like sketch recognition have begun to appear more widely in educational applications. Many such applications also include some form of automated feedback to users. For example, Sketch Worksheets is an educational tool in which instructors can program certain facts about drawings which may be used to generate feedback [24, 12, 23]. One system helps teach users to draw through a sketch interface by comparing how similar the input sketch is to the goal and sharing the evaluations with the students [18]. A similar idea has been applied specifically to drawing portraits; iCanDraw? guides students through drawing faces step-by-step with feedback along the way [10]. Mechanix also includes an evaluation system through which students can request updates on their progress [3].

Tracking student activity in online systems may be foremost associated with preventing cheating and has been studied extensively in that area [17] [6]. In recent years, the field of Educational Data Mining (EDM) has begun to emerge as an open approach to examining educational data from a multitude of perspectives [15] [16]. Among those perspectives is tracking student behavior in an attempt to characterize their engagement and correlate learning outcomes [8]. Champaign et al. recently used such an approach to correlate student improvement with the time spent on each task [9]. This work uses a similar basis, but sketch recognition provides an entirely new dynamic in that it creates a very personal connection with the student, as opposed to entering numbers on a keyboard or some similar means. Interpreting this data intelligently could be more useful than just as a source for basic feedback.

3 Overview of Mechanix

Mechanix is the source of student data that was analyzed. As mentioned before, it is a sketch-based grading and tutoring system, primarily designed for engineering statics courses [11, 19]. It allows instructors to upload assignments and tutorials, drawing trusses and diagrams by hand and defining equations, values, and units through the provided interface elements. Students may then complete assignments by hand drawing their solutions into the system [21, 20]. Mechanix provides real-time visual feedback through color-coding and personalized instructional feedback as requested. In the process, it aims to help them grasp the basic concepts behind the problem [5]. Figure 1 shows a screenshot of the student interface in Mechanix

[1] Miniwatts Marketing Group, Internet Growth Statistics, 2014, http://www.internetworldstats.com/emarketing.htm

while solving a truss-related problem, and Figure 2 shows another screenshot of the interface while solving a free-body diagram problem.

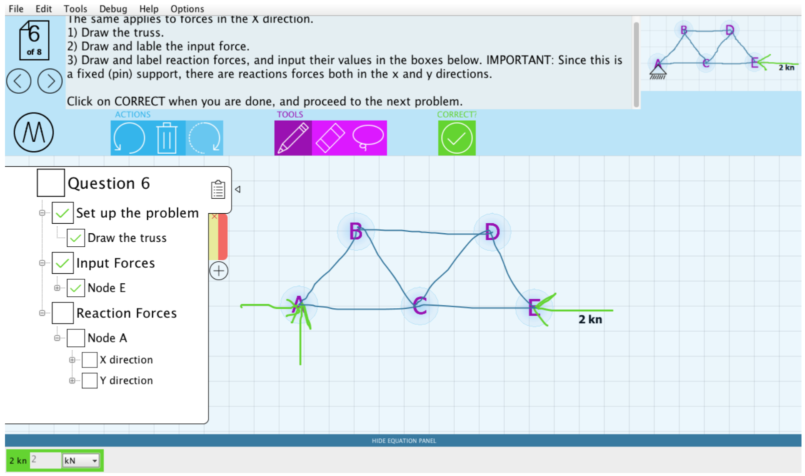

Fig. 1 A screenshot of the student interface in Mechanix, showing a truss with forces, labels, and color-coding.

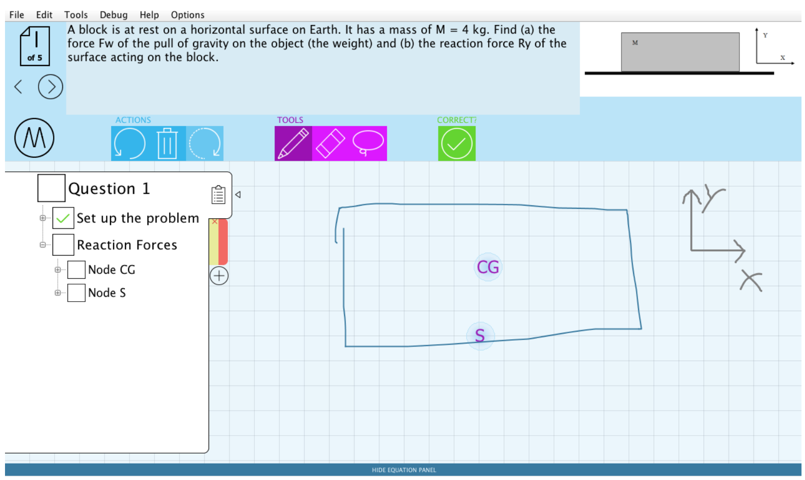

Fig. 2 Another screenshot of the Mechanix student interface showing recognition of free body diagrams and axes.

By using a touch and pen interface, Mechanix keeps the real experience of pen and paper mode intact [4]. There is also a benefit to students when training tools closely resemble real world scenarios. Most of the available software used in teaching statics and engineering drawing have complicated interfaces leading to a steep learning curve. Mechanix is much more direct in its interaction, and the intuitive user interface takes much less time to learn.

4 Data Analysis

Mechanix was used as a supplemental teaching tool during the fall 2014 semester. Data was taken from 51 students who were enrolled in an introductory course at the Georgia Institute of Technology, learning the fundamentals of statics and mechanics that would become critical to their future engineering work. The students ranged in majors from mechanical engineering, aerospace engineering to biomedical engineering among others and were primarily in their freshman years. Two class assignments were given through Mechanix, as well as a tutorial that was offered for additional credit. Hundreds of megabytes of sketching data, primarily drawings of trusses, were gathered from these students, providing a great deal of potential for mining. While much can be learned from the sketches themselves, a component of sketch recognition with implications in artificial intelligence, computer security, and personal health, this data included much more information that could reveal the learning behavior of the students in the field of technology-based learning. Each sketch submitted by a student is saved with a time stamp and the feedback generated by the server for that submission, among other attributes. Mechanix's feedback system provides each student with personalized responses relevant to his or her work. Some of the suggestions the feedback may provide would be to identify missing nodes or forces and incorrect values to the students. Feedback is really helpful if used judiciously. The data collected suggests that a fraction of users request feedback at every step instead of attempting the problem on its merit. This often leads to a student dropping out in the middle of an exercise. Mechanix also informs students, when they are correct. A lot of interesting information can be gathered through the analysis of time and feedback alone.

It is important to note that in the sections that follow, a focus is placed on the students' homework completion rates. Such completion rates may seem like an arbitrary measure, but Atilola et al. have already demonstrated the effectiveness of Mechanix as a teaching tool [2]. Those results suggest that students will benefit more by continual use of the system for their Mechanix assignments rather than ceasing to use the software. As such, this work examines less how Mechanix can improve students' grades and more how their interaction with Mechanix may be improved to encourage them to use the system more. Completion rates are used because they are the best standalone measure of determining a student's sustained usage of the software.

4.1 Time between Requests for Feedback

Because Mechanix is a feedback-based system, it is important to consider how students are using the feedback and whether or not it is benefiting them. One means of doing this is by examining the frequency of feedback requests against completion of problems for each student. By using the time stamp with each repeated submission of a problem, an average time between requests for feedback was generated for each student. Individual student behavior becomes apparent through this analysis, as we see that some students regularly ask for feedback at short intervals while others take more time thinking about the problem before checking. Patterns appear when considering the group of students. The results show that students with more time between feedback requests performed better on average than those that spent less time. This is not particularly surprising because students that spend more time on a problem without requesting help are more likely to think about the problem carefully to gain a good understanding of the concepts, which is reflected by a higher grade. Baker et al. demonstrated similar results in [7] in which students who were "gaming" the online learning system achieved lower learning rates than those who were not. Figure 3 shows the average time between feedback requests in minutes against the completion rates for the students. A trendline generated from regression analysis is overlaid.

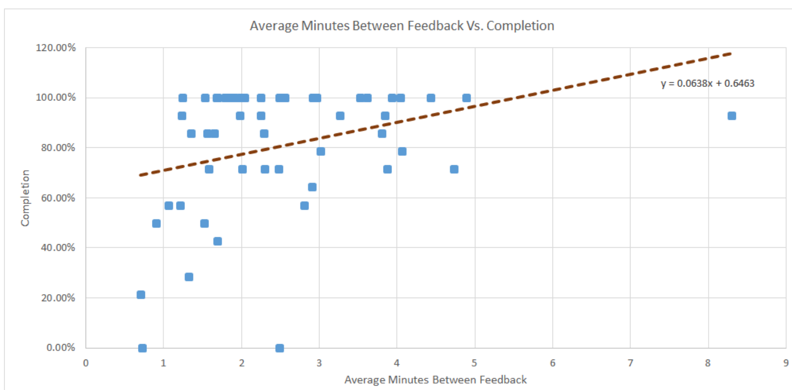

Fig. 3 Average time between submissions on a minute scale plotted versus the student completion rates.

From regression analysis, the linear prediction model fit to the data showed a statistically-significant pattern in the data ($p = 0.033$ — Nonzero slope). As mentioned above, this model is easily understood when looking at students with little time between feedback requests; these students are not spending enough time on the problem before requesting feedback from the system. The more interesting result from this model is that we clearly see there is no benefit to students working longer than 5 minutes on a problem. While working *longer than 5 minutes* did not seem to induce higher quitting rates, it obviously does not have any valuable return for the students. One way to apply this knowledge is to add recommendations to Mechanix. For example, by considering this metric across all the students of a given assignment, Mechanix could determine an *optimal time* for students to work a problem. If they seem to be taking too long, Mechanix could recommend they check the current status.

While it may not be very evident from the trendline, more detailed analysis of the data does show that students' completion rates increase significantly when they spend even two minutes on a problem as opposed to just one minute or less. In fact, many of the low completion rates are concentrated *below one minute*. The primary difficulty encountered by those in this category seemed to be an over-reliance on the feedback. A student might get stuck on a particular problem and check it repeatedly by testing different values or making slight alterations to their drawing but not take the time to solve the underlying issue. For instance, one such student (user58) received 14 errors (4) related to missing or incorrectly drawn forces on a diagram within four minutes. Each error was very similar, and the student may have benefited most from redrawing the problem and considering more carefully what forces to draw. As it was, the student moved on from the problem and, after trying a couple more assignment questions later, eventually gave up without getting any problems correct.

This over-reliance on the auto-grading capabilities of the system can lead to frustration. To counter this frustration, it is certainly recommended that students spend more time focused on solving the problem than on the teaching tools capabilities, but the data gathered from this experiment show that only a small amount of extra time might benefit students significantly. In this case, even thirty more seconds of thought before clicking the feedback button had could have a major impact on the student's willingness to finish the assignment. Perhaps cooldowns on Mechanix's feedback could be paired with recommendations to service students more effectively. They could receive more appropriate recommendations based on their personal activity, whether to request feedback more or attempt to move on to other problems rather than get stuck. In essence we think that frequency of requests for feedback should be monitored and assistance on problems should be given accordingly. Such a feature would need to be implemented carefully so as to be a help and not a hindrance to students.

4.2 Number of Attempts before Moving Forward

Students also encountered frustration in the form of not being able to solve a particular problem and giving up before moving forward. As it is important to spend time thinking about a problem rather than relying on feedback, it is likewise important to be able to move on when one is stuck. Some students do not practice this technique, but it is often recommended by teachers. By examining repeated submissions of the same problem before moving on without achieving a correct answer, an average number of errors to switching was generated for each student. When compared with the problem completion rate from before, it was found that the students who request large amounts of feedback before moving on are less likely to finish the assignment. These users seem to have difficulty recognizing when they are stuck and become too frustrated to move on after spending too many attempts on a single problem. For example, one student received 18 consecutive errors (4) with the same message and gave up on completing the assignments before getting any questions outside of the tutorial correct. Such persistence does not seem helpful to the student at all.

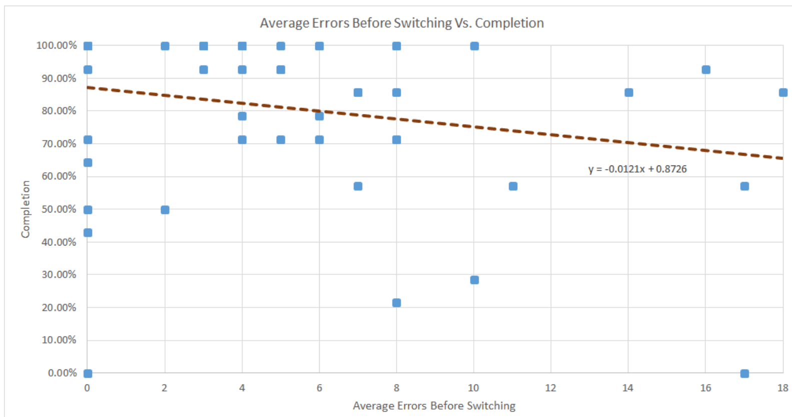

Fig. 4 Average number of submissions before switching without getting correct against the student completion rates.

Figure 4 shows the average errors to switching versus the average completion rates. Again, a linear trendline has been added using regression analysis. We can see a trend from this model which is interestingly the reverse of what we saw in the previous metric. Specifically, while spending increasingly more time on a single problem attempt before requesting feedback did not seem to harm students, however, spending more attempts on a single problem before switching does. Thus, multiple attempts lead to frustration more easily in students than single, long attempts. Unfortunately, though the trendline shows some interesting features, it is not statistically-significant for this metric with p = 0.22 (can not reject that slope is zero). This is mostly due to the fact that while students with more attempts before switching generally did more poorly, there were a number of students who never switched problems before completing them. These users added multiple zeros but may still have achieved reasonable grades on the assignments, indicating that the data may not be best approximated with a linear model.

To take a closer look at the impact of staying on a problem too long, the students were split into two groups: 1) those who completed everything and 2) those who quit without finishing. A t-test on these two groups did yield a statistically-significant separation (p = 0.012). Students with 100% completion spent *3.6 attempts* on average before switching, while those who did not finish the assignments spent *6.7 at-*

tempts on average. This approach more clearly proves the trend we discussed earlier that students are more prone to giving up after experiencing many repeated errors.

This result is particularly relevant to technology-based learning tools because they have the capability to help users recognize when they are stuck. For example, it may ultimately help students perform better to recommend that they move on to another problem after *five or six repeated errors*. Analysis of submission history data can help instructors detect specific concepts on which a number of students get stuck. Instructors can later create and upload more tutorials explaining these concepts. With real time analysis of submission data, one can implement automated tutors that identify higher-level concepts a student is struggling with and provide just-in-time instructional support.

4.3 Impact of Tutorials

In looking at general submission trends among all the students, another lesson may be learned that could be useful to other web-based learning tools or courses – the value of tutorials. Several students encountered difficulty in doing the assignments and so returned to the tutorial before completing the assignments. Because tutorials are so heavily relied upon, they should be thoughtfully crafted so that students can learn how to solve any type of problem they may encounter using the software tools. In this study, there was a noticeable jump in difficulty in the tutorial, plainly visible in the number of submissions for each question of the tutorial, which is shown in Figure 5. By more closely examining the feedback students were receiving, it is apparent that some fundamental knowledge for using the software correctly was being forgotten. Since success of assignments depends upon skills to use the software and a good grasp of concepts, tutorials should be targeted towards both of them. Using simple questions early on to reinforce the most basic concepts may be one method to allow students to progress through problems more smoothly in the future.

Mechanix serves as a sketch based educational software for engineering statics course. One of the primary goals of this software is to keep students engaged in the course by helping them with automated tutoring and assignment grading. This aspect of Mechanix can be extended to Massive Online Open Courses as well. One of the main challenges of MOOCs is student persistence. Data shows that most MOOCs have completion rates of less than 13%. Moreover auto graded MOOCs have a higher completion rates than peer graded ones [2]. To keep students invested and interested in an online course, short quizzes and tutorials that are automatically graded should be introduced between lectures. The data from Mechanix shows that students who complete the tutorials have a much higher chance of completing the assignments which shows that they have grasped the concepts better.

---

[2] Jordan, K. MOOC Completion Rates, 2013, http://www.katyjordan.com/MOOCproject.html

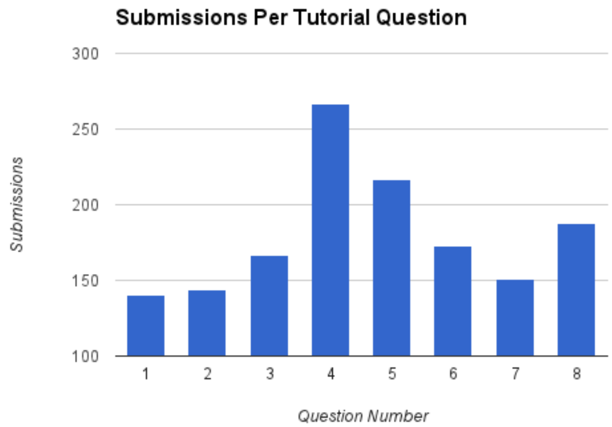

Fig. 5 Total number of submissions for each question in the tutorial.

5 Future Work

The best way to improve Mechanix's interaction with students is to take cues from data, and there are some important results. Feedback is helpful in keeping students moving along, but it can be a hindrance. By incorporating cooldowns or more informative messages to assist students in thinking more carefully about a problem, a lot of potentially frustrating issues may be avoided. Students benefit from being flexible. Moving on to another problem after about four failed submissions gave students the most success in this study. Mechanix could use this result to provide recommendations in certain instances where students appear to be stuck. Tutorials are also extremely valuable. Not only should they be designed to teach students all the information they should need to know to use the software but also provide examples of each type of problem that may be encountered. Simple problems are a good way to reinforce software usage principles, but all types of different problems should appear so that students are prepared. We also wish to look more closely at these metrics in the future and compare them with data collected from more students. By gathering more data, it may be possible to fit a better model to the existing metrics or determine new ones that could be other avenues of software improvement.

6 Conclusion

There are vast amounts of data available from software-based educational tools, and trends in this data should be used to inform the development of next-generation education software. This software should be as straightforward for students as possible, through sketch recognition, and it should also make use of student behaviors to act as a tutor. The results found here, though primarily targeted at Mechanix, may be applicable to many different learning programs. Interaction based on an understanding of students could lead to systems that more closely resemble human teachers, allowing for less expensive but still very personal ways for students to learn.

Acknowledgements This research was funded by NSF EEC Grant No. 1129525. The authors would like to thank Dr. Matthew Green for using Mechanix in his classroom for testing; the members of the iDreem and Sketch Recognition Labs, particularly Stephanie Valentine and David Turner for their significant contributions to Mechanix; and the Computer Science & Engineering department at Texas A&M University.

References

1. Allen, I.E., Seaman, J.: Online Nation: Five Years of Growth in Online Learning. ERIC (2007)
2. Atilola, O., Field, M., McTigue, E., Hammond, T., Linsey, J.: Evaluation of a natural sketch interface for truss fbds and analysis. In: Frontiers in Education Conference (FIE), 2011, pp. S2E–1–S2E–6 (2011). DOI 10.1109/FIE.2011.6142959
3. Atilola, O., Field, M., McTigue, E., Hammond, T., Linsey, J.: Mechanix: A sketch recognition truss tutoring system. In: ASME 2011 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, pp. 645–654. American Society of Mechanical Engineers (2011)
4. Atilola, O., Valentine, S., Kim, H.H., Turner, D., McTigue, E., Hammond, T., Linsey, J.: Mechanix: A natural sketch interface tool for teaching truss analysis and free-body diagrams. Artificial Intelligence for Engineering Design, Analysis and Manufacturing **28**(02), 169–192 (2014)
5. Atilola, O., Vides, F., Mctigue, E.M., Linsey, J.S., Hammond, T.A.: Automatic identification of student misconceptions and errors for truss analysis. In: 119th American Society for Engineering Education Annual Conference & Exposition (ASEE). American Society for Engineering Education (2012)
6. Baker, R.S., Corbett, A.T., Koedinger, K.R.: Detecting student misuse of intelligent tutoring systems. In: Intelligent tutoring systems, pp. 531–540. Springer (2004)
7. Baker, R.S., Corbett, A.T., Koedinger, K.R., Wagner, A.Z.: Off-task behavior in the cognitive tutor classroom: when students game the system. In: Proceedings of the SIGCHI conference on Human factors in computing systems, pp. 383–390. ACM (2004)
8. Baker, R.S., D'Mello, S.K., Rodrigo, M.M.T., Graesser, A.C.: Better to be frustrated than bored: The incidence, persistence, and impact of learners cognitive–affective states during interactions with three different computer-based learning environments. International Journal of Human-Computer Studies **68**(4), 223–241 (2010)
9. Champaign, J., Colvin, K.F., Liu, A., Fredericks, C., Seaton, D., Pritchard, D.E.: Correlating skill and improvement in 2 moocs with a student's time on tasks. In: Proceedings of the first ACM conference on Learning@ scale conference, pp. 11–20. ACM (2014)
10. Dixon, D., Prasad, M., Hammond, T.: icandraw: using sketch recognition and corrective feedback to assist a user in drawing human faces. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, pp. 897–906. ACM (2010)
11. Field, M., Valentine, S., Linsey, J., Hammond, T.: Sketch recognition algorithms for comparing complex and unpredictable shapes. In: Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence - Volume Volume Three, IJCAI'11, pp. 2436–2441. AAAI Press (2011). DOI 10.5591/978-1-57735-516-8/IJCAI11-406. URL http://dx.doi.org/10.5591/978-1-57735-516-8/IJCAI11-406

12. Forbus, K., Usher, J., Chang, M.: Sketch worksheets: A brief summary. In: A.A. T. Hammond S. Valentine, M. Payton (eds.) The Impact of Pen and Touch Technology on Education. Part III, Chapter 10. Springer. New York (2015)
13. Hiltz, S.R., Turoff, M.: Education goes digital: The evolution of online learning and the revolution in higher education. Commun. ACM **48**(10), 59–64 (2005). DOI 10.1145/1089107.1089139. URL http://doi.acm.org/10.1145/1089107.1089139
14. Nelligan, T., Polsley, S., Ray, J., Helms, M., Linsey, J., Hammond, T.: Mechanix: A sketch-based educational interface. In: Proceedings of the 2015 ACM International Conference on Intelligent User Interface (2015)
15. Romero, C., Ventura, S.: Educational data mining: a review of the state of the art. Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on **40**(6), 601–618 (2010)
16. Siemens, G., d Baker, R.S.: Learning analytics and educational data mining: towards communication and collaboration. In: Proceedings of the 2nd international conference on learning analytics and knowledge, pp. 252–254. ACM (2012)
17. Siyao, L., Qianrang, G.: The research on anti-cheating strategy of online examination system. In: Artificial Intelligence, Management Science and Electronic Commerce (AIMSEC), 2011 2nd International Conference on, pp. 1738–1741. IEEE (2011)
18. Soga, M., Matsuda, N., Taki, H.: A sketch learning support environment that gives area-dependent advice during drawing the sketch. Transactions of the Japanese Society for Artificial Intelligence **23**, 96–104 (2008)
19. Valentine, S., Field, M., Hammond, T., Smith, A.: A shape comparison technique for use in sketch-based tutoring systems. In: Proceedings of the 2011 Intelligent User Interfaces Workshop on Sketch Recognition (Palo Alto, CA, USA, 2011), IUI, vol. 11 (2011)
20. Valentine, S., Lara-Garduno, R., Linsey, J., Hammond, T.: Mechanix: A sketch based tutoring system that automatically corrects hand-sketched statics homework. In: T. Hammond, S. Valentine, A. Adler, M. Payton (eds.) The Impact of Pen and Touch Technology on Education. Springer, New York (2015)
21. Valentine, S., Vides, F., Lucchese, G., Turner, D., Kim, H.h., Li, W., Linsey, J., Hammond, T.: Mechanix: A sketch-based tutoring and grading system for free-body diagrams. AI Magazine **34**(1), 55 (2012)
22. Valentine, S., Vides, F., Lucchese, G., Turner, D., Kim, H.h., Li, W., Linsey, J., Hammond, T.: Mechanix: A sketch-based tutoring system for statics courses. In: IAAI (2012)
23. Wetzel, J., Forbus, K.: Increasing student confidence in engineering sketching via a software coach. In: A.A. T. Hammond S. Valentine, M. Payton (eds.) The Impact of Pen and Touch Technology on Education. Part III, Chapter 11. Springer. New York. Springer International Publishing (2015)
24. Yin, P., Forbus, K., Usher, J., Sageman, B., Jee, B.: Sketch worksheets: A sketch-based educational software system (2010)