# Mechanix: A Sketch-Based Educational Interface

**Trevor Nelligan**
Sketch Recognition Lab
Texas A&M University
trevornelligan@tamu.edu

**Seth Polsley**
Sketch Recognition Lab
Texas A&M University
spolsley@tamu.edu

**Jaideep Ray**
Sketch Recognition Lab
Texas A&M University
jaideep.ray@tamu.edu

**Dr. Michael Helms**
iDreem Lab
Georgia Institute of Technology
mhelms3@gatech.edu

**Dr. Julie Linsey**
iDreem Lab
Georgia Institute of Technology
julie.linsey@me.gatech.edu

**Dr. Tracy Hammond**
Sketch Recognition Lab
Texas A&M University
hammond@cse.tamu.edu

## ABSTRACT
At the university level, high enrollment numbers in classes can be overwhelming for professors and teaching assistants to manage. Grading assignments and tests for hundreds of students is time consuming and has led towards a push for software-based learning in large university classes. Unfortunately, traditional quantitative question-and-answer mechanisms are often not sufficient for STEM courses, where there is a focus on problem-solving techniques over finding the right answers. Working through problems by hand can be important in memory retention, so in order for software learning systems to be effective in STEM courses, they should be able to intelligently understand students sketches. Mechanix is a sketch-based system that allows students to step through problems designed by their instructors with personalized feedback and optimized interface controls. Optimizations like color-coding, menu bar simplification, and tool consolidation are recent improvements in Mechanix that further the aim to engage and motivate students in learning.

### Author Keywords
Guides; Computer-assisted Instruction (CAI); Graphical User Interfaces (GUI); Interaction Styles; User-centered design

### ACM Classification Keywords
H.5.2 Information Interfaces and Presentation (e.g. HCI): K.3.1Computer Uses in Education

## INTRODUCTION
Sketching is one of the most important problem-solving techniques students can use. No matter their discipline, students everywhere draw diagrams, tables, and scratch work to help them illustrate and understand concepts. Students benefit from learning to draw engineering drawings by hand [12] [5], yet, as high enrollment in university courses and efforts to reduce costs is pushing many schools toward software-based teaching tools, the natural interaction of sketching is often lost. STEM (Science, Technology, Engineering, and Mathematics) fields are particularly good candidates for software teaching systems, owing to the complexity of content and difficulty in finding qualified graders, but existing systems are often insufficient for these courses. Because of an emphasis on technique and understanding over calculating the correct answer, STEM courses are difficult to grade through purely quantitative question-and-answer mechanisms. It is important that teaching software in STEM courses be more flexible and intuitive so that the focus is on student learning.

Mechanix is an interactive teaching tool designed to help students through the complete process of creating and solving truss and free body diagram problems [13] [4]. Mechanix uses a combination of sketch recognition and numerical questions to evaluate student work and provide personalized feedback [1]. In this way, students are free to work through a problem by hand, but they still must adhere to the guidelines of the question, as defined by the instructor, in order to receive credit. Although this project has been under development for some time, Mechanix was recently deployed across classes in three universities in a newly-improved form. The new interface includes an enhanced layout with more natural groupings of features and controls.

## PRIOR WORKS
Technology-based solutions for teaching engineering, even some systems that include sketching, are not new. Sketch Worksheets is a domain-free sketch-based system for walking students through any problem designed by the instructor, but it requires that instructors be comfortable with a special language in order to define facts about the sketch that may be used to guide students [14]. Other packages such as WinTruss [11], Bridge Architect, and McGraw Hill Connects engineering section provide software-based truss problem solving. These systems do not include sketching, and so must fill in areas of the solution and ask students to provide missing


*IUI'15 Companion*, March 29–April 1, 2015, Atlanta, GA, USA
ACM 978-1-4503-3308-5/15/03.
http://dx.doi.org/10.1145/2732158.2732194

pieces. Though similar in parts to Mechanix, none of these systems include the combination of sketch recognition with trusses and free body diagrams in order to provide a complete solution for working students through an entire problem. The usage of sketch recognition gives Mechanix several advantages that benefit both students and instructors.

## MECHANIX OVERVIEW

Mechanix is built around providing instructors and students with an intuitive means of creating and solving truss and free body diagram problems [2]. Because of the focus on sketching and natural interaction, the software relies heavily on sketch recognition algorithms to analyze drawings and perform actions intelligently. Mechanix identifies and distinguishes among shapes without direct user intervention and provides automatic, yet personalized, feedback and grading. Furthermore, even though Mechanix is primarily directed at civil and mechanical engineers, it has been designed to be simple enough that anyone may use it with very little knowledge of its domain.

Mechanix is designed to emulate the pen and paper interaction that students are familiar with. In each interface mode, users are presented with a large canvas on which to draw, so that they do not feel restricted by not having enough room to draw in [9]. Users can keep on modifying their diagrams based on Mechanixs evaluation until they have met all of the criteria. The software may also be used on many different form factors with varying input devices: finger on tablets, mouse on desktop computers, digital pens/styli on other touch-based devices.

Mechanix is a modular client-server web application, but it is very easy for users to setup. Students and instructors download their respective lightweight client to start using Mechanix, and all of the server-side configuration is set automatically. User logins, permissions, and sketch submissions are all routed to the server, which verifies and stores information in a MySql database.

## INTERACTION

An important goal of the Mechanix project is to engage and motivate students in learning. Mechanix includes many advanced sketch recognition algorithms that run transparently to the user, wrapped under a carefully-designed interface with expertly-tuned features like real-time evaluation. The user experience begins with the interface.

### Interface

The interface has evolved through various iterations to ensure its effectiveness. We have used data points collated over multiple terms in different universities to iron out the main difficulties encountered by users in using the program. In reworking the design of the user interface, Gestalt Principles, a well-known set of design guides [10] [3], played a crucial role in informing the structure of the interface. The overall motif of the program has been altered to be more cohesive, and to incorporate the flat design philosophy that has become prevalent in recent years [7]. This design philosophy eliminates superfluous elements and was helpful in distilling the program to its most critical elements.

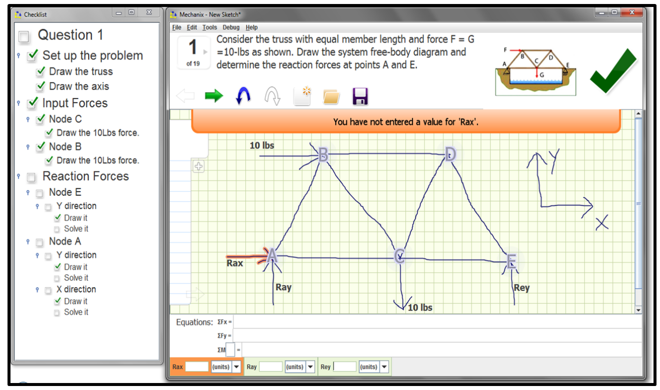


**Figure 1. The original Mechanix interface without color-coding or a unified design style.**

Mechanix has three interface modes - an instructor interface where instructor can upload the assignment problems and tutorial examples, a student interface where students can work on the assignment problems, and a review mode available to instructors that allows them to examine student submissions.

Instructors must first create assignments and problems in the instructor interface. Rather than learning a complicated language or writing code, they can just draw. Places for filling in equations, values, and units are all provided. After problems are submitted online, students from that instructors section are able to access the assignment and attempt the problems themselves from the student interface. Their submissions are compared directly against the instructors by the server and saved for later review by the instructor.

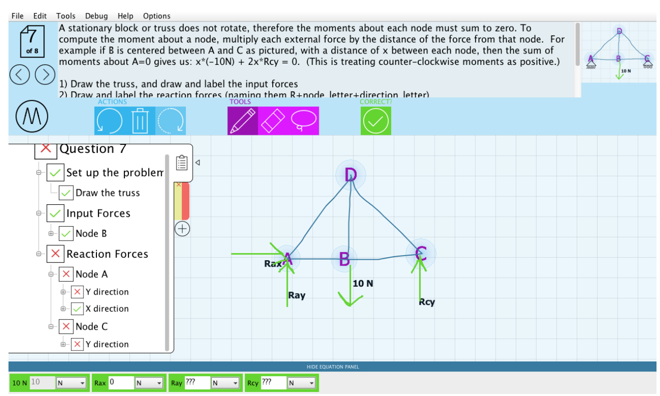


**Figure 2. The new student user interface. Of particular note is the automated color-coding of arrows so that students know immediately whether or not their arrows have been recognized. Other important changes can be seen as well, including the unified button styles.**

### Recognition

Though the workflow is straightforward, making Mechanix seem natural to users requires a great deal of underlying recognition algorithms. The building blocks of recognition are points, strokes, and shapes. Mechanix generates points from movement over the input device. A stroke is a collection of points from pen down to pen up, and a shape is a collection of strokes that is recognized by Mechanix.

Recognition starts with the generation of a stroke, which is then sent to a shape recognizer, PaleoSketch [8], for classification as a primitive shape like line, arc, circle, or polyline. After primitive recognition, the groupings of shapes are sent to complex shape recognizers for higher-level recognition. These are partly based on geometric constraints. Mechanix has specialized truss recognizers as trusses are the most important figure in any statics course.

Great recognition is only meaningful if the user interaction benefits from it. In the case of Mechanix, recognized shapes are used to drive intelligent behaviors that directly help instructors and students. When creating forces, arrows and nodes are automatically recognized, so Mechanix can suggest a force name using the direction and node labels. The instructor need not use any tool for this. Even after drawing a truss, nodes are immediately classified as such and recommended labels are made available. Students interact similarly as instructors do. Truss nodes are automatically recognized and labeled; arrows are color-coded. Color-coding acts as a visual guide to further direct students and reinforce good work [6].

User actions are greatly simplified through this recognition process. In other programs that focus less on organic interaction, one might have to click a button to submit a drawing for evaluation and only then know whether their drawing was correct. The system Mechanix employs is constantly evaluating and updating the user on the results of those evaluations. For example, once a truss is identified, nodes are automatically labeled. Arrows may be drawn in many different ways, but once recognized, they are colored green and can be labeled as a force. With these automatic responses, the program acknowledges the users actions as correct as soon as the user has lifted their mouse/stylus/finger. This type of affirmative feedback helps students stay encouraged and feel confident they are on the right track.

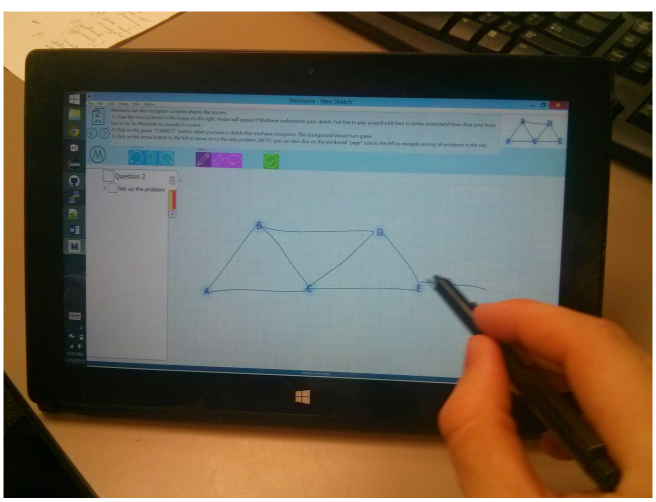

Figure 3. The interface being sketched on using a tablet, although any supported input may be used in Mechanix.

### Feedback

Because Mechanix aims to engage and motivate students in learning, it offers a key feature of real-time feedback to

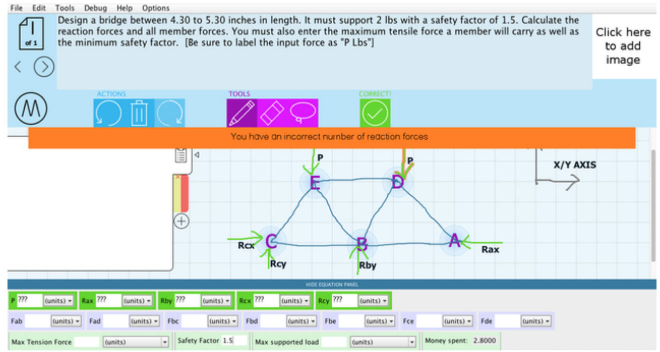


Figure 4. Shown here is the student interface for creative response problems. In this mode, the instructor defines constraints for the system, and students must design a truss that meets the criteria. The trusses, forces, and equations are all evaluated automatically by Mechanix, which can provide very beneficial feedback for students who are unsure about the truss design or forces.

students. Personalized feedback is the most prominent difference between student and instructor interaction. Color-coding and node labeling are important, but when students are attempting to solve equations and draw forces, they may need some extra instruction to get them on the right track. Using the its recognition and comparative capabilities, Mechanix generates directed feedback to inform students what they need to do. If a force is missing, the student is told what node needs an additional force. If a numerical result is wrong, the relevant value will be highlighted, and students are even informed if their units are incorrect. The sketches use thresholds in determining student accuracy that provides them some forgiveness without being too inaccurate.

Each problem in Mechanix also has a dynamic checklist. The dynamic checklist along with the user-requested feedback assists students to tackle a problem in a more structured order. The checklist updates as feedback is requested, and it helps reinforce good problem-solving strategies.

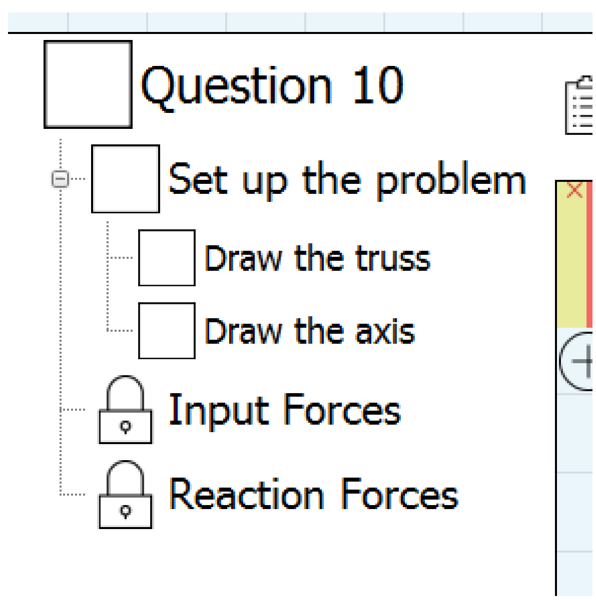


Figure 5. One of the ways the interface provides feedback – the checklist.

## CONCLUSION AND DISCUSSION

Mechanix is educational software directed at aiding students and instructors in learning and teaching free body diagrams in statics. It is uses advanced sketch recognition algorithms to provide real-time grading and feedback. These capabilities are paired with carefully-crafted interface elements like automated node labeling and color-coding to provide students with positive feedback to help direct the problem-solving process. One of the most valuable tools Mechanix provides students is real-time feedback, specific to the work they have done. This is helpful in all kinds of problems, but it can be especially useful in problems where multiple answers are allowed. All of these interactions are important in STEM courses as they reinforce problem-solving methodologies and encourage students to remain engaged in their engineering courses, characteristics that Mechanix seeks to exemplify.

## ACKNOWLEDGMENTS

The authors would like to thank the NSF for funding this research under NSF EEC Grant No. 1129525. The authors also wish to thank the members of the Sketch Recognition Lab who have provided significant contributions to Mechanix and support for this work, specifically Stephanie Valentine and David Turner. Additional thanks go to Dr. Matthew Green for testing Mechanix with his students at LeTourneau University, the iDreem and Sketch Recognition Labs, and the Computer Science and Engineering department at Texas A&M University.

## REFERENCES

1. Atilola, O., Field, M., McTigue, E., Hammond, T., and Linsey, J. Evaluation of a natural sketch interface for truss fbds and analysis. In *Frontiers in Education Conference (FIE), 2011* (Oct 2011), S2E–1–S2E–6.
2. Atilola, O., Valentine, S., Kim, H.-H., Turner, D., McTigue, E., Hammond, T., and 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 (2014), 169–192.
3. Chang, D., Dooley, L., and Tuovinen, J. E. Gestalt theory in visual screen design: A new look at an old subject. In *Proceedings of the Seventh World Conference on Computers in Education Conference on Computers in Education: Australian Topics - Volume 8*, CRPIT '02, Australian Computer Society, Inc. (Darlinghurst, Australia, Australia, 2002), 5–12.
4. Field, M., Valentine, S., Linsey, J., and 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, AAAI Press (2011), 2436–2441.
5. Lane, D., Seery, N., and Gordon, S. The understated value of freehand sketching in technology education. *Engineering Design Graphics Journal 73*, 3 (2009).
6. Michalski, R. The influence of color grouping on users visual search behavior and preferences. *Displays* (2014).
7. Page, T. Skeuomorphism or flat design: future directions in mobile device user interface (ui) design education. *International Journal of Mobile Learning and Organisation 8*, 2 (2014), 130–142.
8. Paulson, B., and Hammond, T. Paleosketch: Accurate primitive sketch recognition and beautification. In *Proceedings of the 13th International Conference on Intelligent User Interfaces*, IUI '08, ACM (New York, NY, USA, 2008), 1–10.
9. Plimmer, B., Tang, G., and Young, M. Sketch tool usability: Allowing the user to disengage. *People and Computers XX (Proceedings of HCI 2006)* (2006).
10. Puustinen, M., Baker, M., and Lund, K. Gestalt: A framework for redesign of educational software. *Journal of computer assisted learning 22*, 1 (2006), 34–46.
11. Sutton, M. G., and Jong, I.-C. A truss analyzer for enriching the learning experience of students. In *ASEE Annual Conference* (2000).
12. Ullman, D. G., Wood, S., and Craig, D. The importance of drawing in the mechanical design process. *Computers & graphics 14*, 2 (1990), 263–274.
13. Valentine, S., Vides, F., Lucchese, G., Turner, D., Kim, H.-h., Li, W., Linsey, J., and Hammond, T. Mechanix: A sketch-based tutoring system for statics courses. In *IAAI* (2012).
14. Yin, P., Forbus, K. D., Usher, J. M., Sageman, B., and Jee, B. D. Sketch worksheets: A sketch-based educational software system. In *IAAI* (2010).