The Killer App for Sketch-Based Interfaces is …

Joseph J. LaViola Jr.
University of Central Florida
School of EECS
4000 Central Florida Blvd.
Orlando, FL 32816 USA
jjl@eecs.ucf.edu

CHI 2010, April 10–15, 2010, Atlanta, Georgia, USA.
ACM 978-1-60558-930-5/10/04.

Abstract
This paper argues that the killer application for sketch-based interfaces is STEM education. It presents several reasons for how natural, sketch-based interaction combined with computation and visualization can provide intuitive tools for learning. The paper also presents an example scenario for a sketch-based learning system and discusses some of the challenges that must be addressed in order to develop these types of interfaces.

ACM Classification Keywords
H5.m. Information interfaces and presentation (e.g., HCI): Miscellaneous. K3.1. Computers and Education: Computer Uses in Education – computer assisted instruction.

Introduction
In any new field of computer science, the question of what is the “killer app” is always asked. The field of sketch-based interfaces and recognition is no exception. This question is often philosophical in nature because of the struggle with finding such an application. However, in sketch-based interfaces, the killer app is right in front of our noses. In my opinion, the killer app for sketch-based interfaces is STEM education. Why? Well, in order to answer this question,

we must look at what sketch-based interfaces provide the user. The single most important component of a sketch-based interface is undoubtedly the ability to enter information into a computer with a pen or stylus on a 2D surface. This ability lets users transfer the knowledge they have gained with using pencil and paper to the computer. Given the 2D nature of pen input, users have the ability to enter information into a computer naturally, especially when that information is 2D in nature. It turns out that there is a plethora of information used for communication in the STEM disciplines that requires 2D notations including mathematics, chemistry, logic diagrams, and so on. With the ability to enter this information into the computer as easily as they can with pencil and paper, users can leverage the power of computation coupled with natural input.

Natural 2D input is only part of the reason why STEM education is the killer app for sketch-based interfaces. It is what we can do with pen input once it has been entered that can turn simple information entry into a learning vehicle. The key component here is that the system can interpret what the user has entered and then provide appropriate feedback. This feedback is not only showing the computer has understood what the user has written from a syntactic and semantic point of view, but also provides information about the meaning of the sketch. For example, providing a user with feedback on whether the answer to a particular physics problem is correct or not is invaluable. Providing the user with why the answer is incorrect is fundamental to STEM learning.

A Vision of the Future

A student taking an introductory physics class goes to the pen-computing lab to work on a homework assignment. The first problem the student needs to work on is based on the concepts of kinetic energy and work. She writes down the problem description from her textbook on a pen computer and the system performs natural language processing on her text to understand the problem’s conditions and the questions she must answer. The problem is asking her to find how much work is performed when block A (see Figure 1) falls two meters to the ground.

The student begins to work on the problem and draws a diagram similar to the one shown in Figure 1 as well as the key variables in the problem. However, she is unsure of how to translate some parts of the word problem into the mathematical equations she needs to solve it. She asks the system for help and it shows her the key parts of the word problem description she needs to focus on, to generate the starting equation

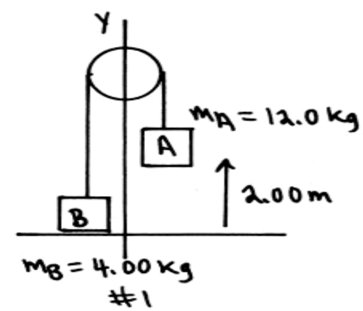

Figure 1. A drawing a student in a college physics course might make to assist in solving problems.

she will need. Because the system understands the semantics of the problem description and how it relates to the key variables, an automated cognitive assistant is able to help her in getting started. When the student writes down the work equation $K_1 + U_1 + W_{other} = K_2 + U_2$, the assistant provides initial feedback based on its current level of understanding, telling her this is a correct starting point. The student then writes down the necessary mathematics to solve the problem and associated the expressions to various parts of the drawing.. The system recognizes the mathematics, the diagram, and the various labels at the syntactic level, and using this information, determines the higher level semantic meaning of the sketch. Indeed, from the mathematics and the drawing, the system understands that she wrote down a solution to a problem involving a pulley system and the calculation of work done using kinetic energy.

She then clicks the application's run button and the pulley system animates with block B rising off the ground until it stops only 1.5 meters above it. Although the student did not specifically enter equations that would specify the motion of the two blocks, the system understands that the mathematics the student wrote down refer to work and kinetic energy and understands how these equations can be transformed into motion equations. This animation shows there was a problem with her calculations. She asks the system for help once again, and the cognitive assistant highlights the W_{other} term and tells her that work is done on the system only by gravity. Since the automated assistant has knowledge obtained through the high level semantics recognition of the sketch, as well as an understanding of the types of conceptual errors students make with this class of problems, it knows how to respond to the student to provide the best level of assistance. She realizes that this term should be zero, fixes the error and runs the animation again. This time the drawing animates correctly, visually showing the solution is correct, and she moves on to the next problem with confidence. Note that a teacher could go through a similar scenario in preparing a dynamic illustration for a class lecture.

Conclusion

The above scenario provides a reason for why STEM education is the killer app for sketch-based interfaces. I say again, STEM education IS the killer app for sketch-based interfaces. The problem is we are not yet at a point where we prove it. To prove the validity of my claim requires something that we have not been able to achieve yet. First, we must have robustness. Evaluations of this technology must be longitudinal in nature and this requires that the systems work as well as commercial software. Students and teachers will have to use these applications for months at a time. Not only must they be robust, but they must be easy to use or at least have mechanisms to teach application functionality. Second, these applications must have more intelligence. It is insufficient for an application to be able to recognize a user's sketch. The application must be able to recognize what is going on with the sketch. For example, the application must understand the user is solving an inclined plane problem and have the appropriate knowledge for this sub domain. I refer to this concept as high level semantic recognition. Until we can prove that teachers teach better and students learn better with sketch-based interfaces, the skeptics will continue to keep the sketch recognition community on the fringes of the mainstream. Thus, we must continue to push forward to prove them wrong.

Biography

Joseph J. LaViola Jr. is an assistant professor in the School of Electrical Engineering and Computer Science at the University of Central Florida as well as an adjunct assistant research professor in the Computer Science Department at Brown University. His primary research interests include pen-based interactive computing, 3D interaction techniques, predictive motion tracking, multimodal interaction in virtual environments, and user interface evaluation. His work has appeared in journals such as IEEE PAMI, Presence, and IEEE Computer Graphics & Applications, and he has presented research at conferences including ACM SIGGRAPH, the ACM Symposium on Interactive 3D Graphics, IEEE Virtual Reality, and Eurographics Virtual Environments. He has also co-authored "3D User Interfaces: Theory and Practice," the first comprehensive book on 3D user interfaces. In 2009, he won an NSF Career Award to conduct research on mathematical sketching. Joseph received a Sc.M. in Computer Science in 2000, a Sc.M. in Applied Mathematics in 2001, and a Ph.D. in Computer Science in 2005 from Brown University.

In terms of sketch-based interfaces, Joseph has been working in the field for the last 6 years. He is the inventor of mathematical sketching, an interaction paradigm for creating illustrations by combining sketches with handwritten mathematics. He has also worked in other areas in the pen- and sketch-based interface and recognition field including developing new mathematical expressions recognition algorithms, evaluating error visualization techniques, and developing new methods for gesture self-disclosure.