Integrating Sketch Worksheets into GIFT

Kenneth D. Forbus, Thomas Hinrichs, Samuel Hill, Madeline Usher
Qualitative Reasoning Group, Northwestern University

INTRODUCTION

While there is evidence that sketching can improve student learning (e.g. Ainsworth et al. 2011; Jee et al. 2014; Scheiter et al. 2017), sketching has rarely used in intelligent tutoring systems because it has been difficult for software to understand what a student's sketch means. To tackle this problem, our group developed CogSketch (Forbus et al. 2011), which provides a robust model of high-level human visual processing and representation. It has been used to model a variety of human visual reasoning and STEM problem-solving (Forbus et al. 2017), providing evidence that its representations and reasoning can provide a solid basis for creating new kinds of sketch-based educational software.

One such new kind of sketch-based educational software are Sketch Worksheets (Yin et al. 2010; Forbus et al. 2017). In a Sketch Worksheet, students tackle problems by drawing and modifying sketches. At any point, they can request feedback, and then improve their sketch. Gradebook software built into CogSketch enables instructors to rapidly grade sketching assignments. CogSketch also provides instructors with detailed assessment data as to the student's process as well as their final product. Importantly, Sketch Worksheets can be authored by instructors, after learning some basics of CogSketch, without programming. This improves dissemnination, by broadening participation in authoring. Sketch Worksheets have now been deployed in several classrooms and subjects (Garnier et al. 2017; Forbus et al. 2018).

While Sketch Worksheets are useful, we believe there is much untapped potential to be explored for using sketching in new kinds of educational software. This paper describes our next step in exploring sketching in intelligent tutoring systems more broadly, by integrating Sketch Worksheets as a medium in GIFT, to benefit from the adaptivity that GIFT provides, and to provide a new capability for GIFT tutors. We describe the basic ideas of sketch worksheets, how we are integrating them into GIFT, and the prototype Simple Machines tutor we are building as an experimental vehicle. Planned experiments are discussed. While this integration is still in progress, we plan to demo a version of the Simple Machines tutor during the symposium.

SKETCH WORKSHEETS: A BRIEF REVIEW

Here we summarize the basics of Sketch Worksheets, more technical details can be found in [Forbus, et al. 2017]. A student tackling a Sketch Worksheet is trying to solve a problem, whose solution is expressed by them drawing or modifying a drawing. For example, in geoscience, they may be asked to mark up a photograph, indicating the properties of the geological strata it illustrates. In engineering graphics, a student might have to redraw a design shown in perspective projection in orthogonal projection. In cognitive science, a student might have to draw a concept map representing the semantic content of a sentence.

Being able to do this range of tasks with the same software requires a fundamentally different approach than the usual view that identifies sketch understanding with sketch recognition. The mapping from concepts in STEM education to visual shapes is many to many: Recognition typically isn't an option. Instead, people talk when they sketch with each other. CogSketch provides a simple interface that enables students to identify how they are considering their ink as partitioned into objects, and give them a label in terms of concepts from the underlying knowledge base (which, to the student, look like natural langauge words or phrases). CogSketch computes visual relationships between the ink entities that students draw, including a rich vocabulary of qualitative relationships that can be used to connect spatial concepts to language. When an instructor authors a worksheet, they draw their solution using CogSketch, which analyzes their ink. The instructor marks some subset of the facts CogSketch computes as important, assigning points to each such fact and providing text to be provided if the analog of that fact is not found in the student's sketch. When a student tackles a worksheet, they draw (or modify existing ink, depending on the worksheet) their solution. When they ask for feedback, CogSketch performs the same analysis as it did on the instructor's sketch, and uses analogy (Forbus et al. 2016) to compare the facts computed about the two sketches. Any differences that correspond to important facts lead to the appropriate advice being produced for the student, or an indication that they've successfully finished the worksheet. They are free to continue working on it as long as they like.

For assessment purposes, CogSketch records timestamps for all of the ink, as well as what order entities were drawn in. The state of the sketch at every time the student asked for help is also recorded, so the instructor (or educational data mining software) can examine their performance in detail and look for patterns across students.

INTEGRATING SKETCH WORKSHEETS INTO GIFT

Our approach is to integrate Sketch Worksheets as a new kind of media that can be used in GIFT tutors. Since GIFT is implemented via an Amazon-based cloud, we are building a cloud-based version of CogSketch to support these experiments.

The cloud-based version of CogSketch is called WebSketch. The services are implemented as Docker containers grouped together in a stack that can be deployed on various cloud services. In order to integrate with GIFT (as well as other educational software infrastructures) our WebSketch stack also contains services to support the LTI protocol (Learning Tools Interoperability, https://www.imsglobal.org/activity/learning-tools-interoperability). Figure 1 shows how this would work with GIFT.

GIFT communicates with WebSketch through LTI. When a GIFT course makes use of a Sketch Worksheet, GIFT uses LTI to handoff control to WebSketch. The student works through the worksheet and a score is returned to GIFT. WebSketch is functioning as an LTI Tool Provider and GIFT is an LTI Tool Consumer in this setup.

The LTI Authorization service in the WebSketch container stack handles the initial communications from a Tool Consumer (GIFT in this case). This includes confirming that the request is coming from a valid Tool Consumer that has permissions to use WebSketch and starting an LTI session. The initial communications from GIFT include a unique and consistent identification of the student (anonymized), which worksheet should be used, and a URI to which the student's score should be returned.

If the LTI request is valid and authorized, control is passed to the WebSketch Node Management service, which chooses an available WebSketch node from a pool of nodes. The selected node is used for the student's session with WebSketch. Each time a student requests tutoring advice from WebSketch, the student's score is updated and conveyed to GIFT. When a student is finished working, their sketch is saved in our Sketch Repository. The saved sketch can be accessed later as needed for assessment and aggregate data collection. If a student revisits a given worksheet through GIFT, the worksheet can be retrieved in the state they last left it.

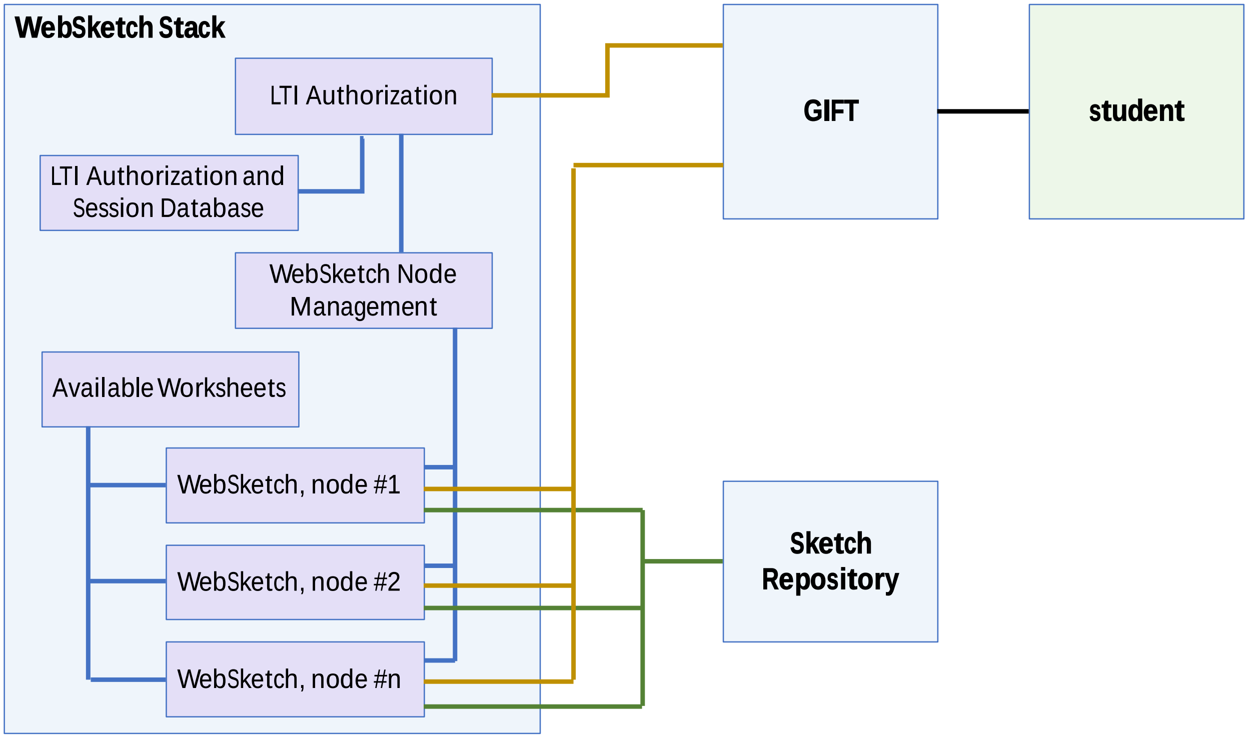

Figure 1. WebSketch/GIFT integration

There are several steps remaining before our initial implementation is finished. The first is a Sketch Worksheet service, which needs to have a repository of blank worksheets, and a registry that connects an ID used in the GIFT tutor with a sketch file. We have implemented a Sketch Repository to store student work, but it is currently running outside the Docker container, whereas for portability it needs to be part of the WebSketch Docker Swarm. There are also a variety of WebSketch UI improvements to be made, including support for CogSketch annotations. We are planning to have these improvements finished before the Symposium.

EXPERIMENT IN PROGRESS: A SIMPLE MACHINES TUTOR

A common topic in STEM instruction for K-12 students, and relevant to understanding and maintaining many kinds of Army equipment, are *simple machines*: Levers, pulleys, screws, and so on. Aside from their practical importance, simple machines provide an interesting application of scientific principles, and provides a bridge between intuition and qualitative understanding to mathematical models that support design and predication. They are also inherently spatial, which makes them a natural for sketching activities. Consequently, we are using GIFT and Sketch Worksheets to create a Simple Machines Tutor.

The learning goals for our curriculum are that, after working through it, a student should be able to

1. Understand the kinematics and force dynamics of simple machines.
2. Recognize structural components, salient relations, quantities and ratios relevant to their operation.
3. Recognize simple machines in the everyday world
4. Understand the tradeoffs between distance, force, and work and how these tradeoffs are manifested in physical systems

5. Have an improved physical intuition for how mechanisms can or will behave and be able to use calculations to verify that intuition
6. Understand the design space of alternative ways to achieve a given effect.

Simple Machines Curriculum Design

The medium of sketching is not limited to representational drawing. It also includes annotating existing sketches or photographs, labeling, and re-arranging constituent components of sketches. With these interactions, it becomes possible to go beyond simple presentation and multiple choice questions to tease out more subtle misconceptions and knowledge gaps.

The curriculum will initially focus on recognition and qualitative analysis of each type of simple machine. Starting with an informal definition and exposure to multiple examples of a simple machine in the context of common everyday devices and situations, the learner is encouraged to compare and make his own analogies to induce a general concept. That generalization can be tested with additional classification examples and near-misses.

Next, more subtle relations can be conveyed in a generative fashion by having the student modify a sketch to alter critical relationships. For example, they might be asked to move the fulcrum in a lever to change it from a first-class to a second-class lever. By not providing explicit choices, it is possible to detect a broader range of misconceptions (e.g., can they even recognize the fulcrum in this context?)

The next activity involves qualitative comparative analysis in which two machines of the same type but different quantitative relations are presented side-by-side. Here, the task is causal reasoning about differences, e.g. which machine would apply greater output force given the same input force. The learner must annotate the depictions to identify which quantity is larger (or smaller) in the selected machine, and also which visual property gave rise to this conclusion, giving a window into their reasoning.

With a solid qualitative foundation, the formal notion of mechanical advantage can be introducd. The three quantities involved (distance, force, and work) will be presented in the context of one kind of simple machine (e.g., inclined plane) and then by analogy those concepts are extended to other machines. So if distance travelled is straightforward in the context of an inclined plane, what does distance travelled correspond to in a screw? (translational distance? distance along the helix?) How about in a block and tackle?

Once correspondences between quantities across different types are established, it becomes possible to draw more abstract analogies between different types of machines. For example, a screw can be conceptually unrolled into an inclined plane. What activities might support comparing the mechanical advantage of one to the other?

As quantities are introduced, it becomes possible to present simple parametric synthesis tasks, in which the student labels a machine’s lengths, angles, and ratios with numerical values to achieve a desired performance. For example, a problem might specify the desired mechanical advantage and one structural parameter, leaving the last parameter open, to be added as an annotation. Finally, when exploring mechanical advantage, we want to avoid functional fixity, in which all machines are seen as force amplifiers, by illustrating the design tradeoffs in other directions. So for example, sometimes the problem will be to attain greater precision rather than force amplification. Vernier calipers exploit the ratio of rotational distance to translational distance to attain high precision. An exercise will have the student modify a sketch by swapping out one or more components of a simple machine (e.g., the pitch of a screw or the diameter) to achieve different kinds of goals, such as minimize displacement, or reduce overall physical size of the machine.

The last set of exercises will addresses structural synthesis. Here, we have to consider simple machines in the context of more complex compound machines. The first kind of synthesis exercise is the sketching analog of fill-in the blank questions. The learner will be presented with an incomplete kinematic chain and a desired global property. They must fill in the missing element by sketching and labeling it, along with its relevant parts and quantities. For example, if the direction of force needs to be reversed, a first-class lever could be used, or a pulley. If rotational to translational conversion is required, either a wheel and axle or a screw could be used.

Another synthesis exercise would be to arrange a fixed set of simple machines into a configuration that achieves a goal. Here, the machines are provided as building blocks and put together, although there could be more than one right answer.

A capstone challenge problem will be to assemble a complex machine in such a way as to demonstrate an understanding of the design space and tradeoffs. Rather than focus on practical quantitative design (which is beyond the scope of this curriculum), the problem may be presented more as the design of a "Rube Goldberg" type machine. The goal could be to translate one displacement into another (or one force to another) with particular inputs and outputs, but using as many types of simple machine as possible. Or it might be to use as few machines as possible. It is not yet clear whether this can be achieved with purely open-ended drawing or whether it would be more feasible to construct a solution from prototypical building blocks that can be stretched, flipped, scaled and positioned. In either case, there is no single right answer, but the ability to compare solutions to a generative grammar of compound machines and analogically compare kinematic pairs to teacher-authored prototypes should allow this exercise to be evaluated and scored.

EXPERIMENT DESIGN AND MATERIALS PREPARATION

In experimenting with the Simple Machines curriculum, we plan on using a two by two design. The first factor will be whether or not sketching is used, the second is whether or not GIFT’s adaptive tutoring capabilities are used. In the non-sketching conditions, additional examples presented via text and diagrams will be used to provide balance, to reduce time at task differences as being a source of confounds. Our qualitative predictions for these conditions are shown in Table 1.

	No Sketching	Sketching
Non-Adaptive	Least learning	In between
Adaptive	In between	Most learning

Table 1. Qualitative Predictions for Student Learning

We will measure learning by using a pre-test and post-test, both administered within the GIFT tutor, so that we can recruit participants on-line. We have created a bank of just over 90 questions, focusing on true/false and multiple choice questions for simplicity. The questions are drawn from open-license materials (e.g. the CK-12 Physical Science for Middle School textbook) or made up ourselves. We estimate that 20 questions for each test will provide enough statistical power to measure learning. We have already selected two sets of 20 questions, balanced in terms of difficulty by ensuring that for every question in the pre-test, there is a roughly equivalent, but not identical, question in the post-test. The pre/post tests will be identical for every participant. We will use different questions in the adaptive conditions from either the pre/post tests.

CONCLUSIONS AND RECOMMENDATIONS FOR FUTURE RESEARCH

This paper summarizes work in progress on integrating Sketch Worksheets with GIFT, to explore how sketching can be combined with adaptive tutoring to hopefully improve student learning better than either could alone. The Simple Machines curriculum we are developing as a testbed will cover each type of simple machine, and include recognition, analysis and synthesis activities, qualitative and quantitative concepts and relations, and includes parametric and structural synthesis tasks. The key to supporting these activities is the ability of CogSketch to permit open-ended sketch input and to extract meaningful relationships from that input. In particular, this allows a student to use annotations to show her work and justify answers – not just say what will happen, but also why. Another advantage of open-ended input over multiple choice is that it can permit vastly more possible answers than would be practical to enumerate explicitly, as in tasks such as unscrambling a shuffled machine or filling in gaps in a kinematic chain with missing elements.

Given where we are on this project, the conclusions and recommmendations we have only concern technology development, rather than tutor effectiveness. First, we suggest that finer-grained granularity on saving be supported, i.e. even when questions are incompletely filled out during authoring, and during long quizzes when taking a course. We recommend that future versions of GIFT consider introducing stronger relationships between test items, so that balanced pre/post tests can be automatically generated from a large question bank. We also recommend that development of the LTI interface continue, expanding as that protocol is fleshed out, to provide a richer channel between Sketch Worksheets (and other extensions) and GIFT.

REFERENCES

Ainsworth, S., Prain, V., & Tytler, R. (2011). Drawing to learn in science. *Science*, 333(6046), 1096-1097.

Forbus, K., Usher, J., Lovett, A., Lockwood, K., and Wetzel, J. (2011). CogSketch: Sketch understanding for Cognitive Science Research and for Education. *Topics in Cognitive Science*, 3(4), pp 648-666..

Forbus, K., Ferguson, R., Lovett, A., & Gentner, D. (2016) Exttending SME to Handle Large-Scale Cognitive Modeling. *Cognitive Science*. DOI: 10.1111/cogs.12377, pp 1-50..

Forbus, K., Chang, M., McLure, M., and Usher, M. (2017) The Cognitive Science of Sketch Worksheets. Topics in *Cognitive Science*, DOI:10.1111/tops.12262.

Forbus, K.D., Garnier, B., Tikoff, B., Marko, W., Usher, M. & Mclure, M. (2018). Sketch Worksheets in STEM Classrooms: Two Deployments. Deployed Application Prize paper. *Proceedings of AAAI 2018*.

Garnier, B., Chang, M., Ormand, C., Matlen, B., Tikoff, B., & Shipley, T. (2017) Testing the efficacy of CogSketch geoscience worksheets as a spatial learning and sketching tool in introductory Geoscience courses. *Topics in Cognitive Science*

Jee, B., Gentner, D., Uttal, D., Sageman, B., Forbus, K., Manduca, C., Ormand, C., Shipley, T., and Tikoff, B. (2014). Drawing on experience: How domain knowledge is reflected in sketches of scientific structures and processes. *Research in Science Education*. 44(6), 859-883.

Scheiter, K., Schleinschok, K, & Ainsworth, S. (2017) Why Sketching May Aid Learning from Science Texts: Contrasting Sketching with Written Explanations. *Topics in Cognitive Science*, DOI: 10.1111/tops.12261.

ABOUT THE AUTHORS

Kenneth D. Forbus *is the Walter P. Murphy Professor of Computer Science and Professor of Education at Northwestern University. His research interests include qualitative reasoning, analogical reasoning and learning, natural langauge understanding, sketch understanding, and cognitive architecture.*

Thomas Hinrichs *is a Research Associate Professor of Computer Science at Northwestern University. His research interests include commonsense reasoning, cognitive architectures and machine learning.*

Samuel Hill *is a graduate student at Northwestern University. His research interests include AI for assisting education and sketch understanding.*

Madeline Usher *is a computer programmer / researcher at Northwestern University. Her research interests include sketch understanding and analogy-based tutoring.*