2014

Instructional Strategies in Diagram-based ITSs: Lessons Learned from Two Tutoring Systems

Enruo Guo
Iowa State University

Stephen B. Gilbert
Iowa State University, gilbert@iastate.edu

Follow this and additional works at: http://lib.dr.iastate.edu/imse_conf

Part of the Computational Engineering Commons, Digital Communications and Networking Commons, Engineering Education Commons, Ergonomics Commons, and the Instructional Media Design Commons

Recommended Citation
http://lib.dr.iastate.edu/imse_conf/87

This Conference Proceeding is brought to you for free and open access by the Industrial and Manufacturing Systems Engineering at Iowa State University Digital Repository. It has been accepted for inclusion in Industrial and Manufacturing Systems Engineering Conference Proceedings and Posters by an authorized administrator of Iowa State University Digital Repository. For more information, please contact digirep@iastate.edu.
Instructional Strategies in Diagram-based ITTs: Lessons Learned from Two Tutoring Systems

Enruo Guo\textsuperscript{1}, Stephen Gilbert\textsuperscript{2}

\textsuperscript{1} Department of Computer Science
\textsuperscript{2} Department of Industrial & Manufacturing Systems Engineering
Iowa State University, Ames, IA 50011, USA
\{enruoguo, gilbert\}@iastate.edu

Abstract. Unlike text-based input to an intelligent tutoring system, a diagram is perceived as a whole state; the operation sequence is less important. Traditional step-wise coaching is not as appropriate in diagram-based intelligent tutoring systems (DITS). From two previous tutoring systems, StaticsTutor and Thermo Cycle Tutor, we propose cross-domain pedagogical guidelines for DITS. In particular, instruction needs to be mapped to a hierarchical understanding of the diagram, where each level focuses on different characteristics of the drawing. Also, instruction needs to address conceptual knowledge and procedure expertise separately. Some practical suggestions are described to achieve these goals, such as 1) different tolerance for error at different level of evaluation, 2) use of Q&A to resolve diagram ambiguity and 3) early loading of expertise that is important for avoiding difficult-to-fix diagrammatic states.

Keywords: intelligent tutoring system, diagram, instruction, pedagogy

1 Introduction

Given the comprehensive advantages of pictorial representations, diagrams play a big role in scientific cognition, e.g., free-body diagrams in physics, Temperature-volume (T-v) diagrams in thermodynamics, circuit diagrams in electrical engineering, and Unified Modeling Language (UML) diagrams in software engineering. Cognitive models of graphics comprehension \cite{1} propose that graphics comprehension involves interaction between bottom-up perceptual processes of encoding information from the graphic as well as top-down processes of applying graph schemas and domain knowledge, which makes it a challenge to teach students how to use diagrams to represent information.

In this paper, we discuss lessons regarding pedagogy that we learned from two diagram-based tutoring systems, and provide cross-domain guidelines for the design of future diagram-based intelligent tutoring systems (DITS).
2 Background and previous work

We have designed and implemented two diagram-based intelligent tutoring systems in engineering statics and thermodynamics courses: StaticsTutor [2] for free-body diagrams, and Thermo Cycle Tutor (Guo et al., in preparation), for T-v diagrams of refrigeration cycles. Even though they focus on different domains, both feature pedagogy aimed at helping students’ conceptual understanding and decision-making at the earliest stage of problem framing.

StaticsTutor was developed to analyze student-drawn free-body diagrams and recognize misconceptions without requiring numerical force values or the need to provide equilibrium equations. Preliminary results with 81 engineering undergraduates in fall 2013 showed that StaticsTutor could detect students' misconceptions that were categorized as “missing basics,” “hinge issue,” “rope issue,” and so on. A post-survey indicated an overall positive experience with the tutor with a mean usability score of 3.5 (SD 1.11).

The Thermo Cycle Tutor implemented a teaching pedagogy (Hagge et al., in preparation) based on decision-making, where class concepts are posed as a set of simple questions that can be answered for all problems in the thermodynamics course. In fall 2013, 42 undergraduate engineering students were given a pre-test on refrigeration cycles and then given the Thermo Cycle Tutor to complete a homework problem. They then took a post-test. Students’ post-test scores improved from 70% to 89% on average. To test retention, they were given a second post-test after four weeks, and they scored an average of 81% better than the pre-test with no additional lectures on refrigeration cycles.

Both tutors faced the challenge of how to analyze the students' diagrams computationally, and how to give appropriate feedback. Pedagogical questions that arise include, "If there are multiple issues with the diagram, which issue should receive feedback first?", "Given an error in the diagram, what can I infer about the student's misconceptions, if any?", and "When should I evaluate the diagram, at each step of construction, or only at the end?"

2.1 Previous work on diagram interpretation and DITS

Koffman and Friedman [3] designed an early instructional tool for diagramming to assist beginning programmers in learning to make a computer-aided flow diagram. They emphasized the problem-framing aspects of diagram planning, and wanted students to use the diagrams to learn the program logic before implementing the code. Usually it is difficult to analyze a diagram at each step of its construction, because there are typically graphic elements that must be added one at a time in no particular order, and the diagram can frequently exist in non-well-formed states that cannot be fully anticipated by the tutor author. However, in Koffman and Friedman’s case, the linear structure and the level of granularity of their diagram components helped this system avoid these open-ended ambiguities that usually occur during construction.

Constraint-based modeling (CBM) has been adopted in ITS community, where the domain knowledge is represented as a set of constraints. By focusing on violated con-
straints, CBM tutors are able to generate instructional actions even without having expert solutions. Instructional feedback is generated by focusing on one genuine misconception if more than one constraint is violated [4]; frequently one misconception will cause the violation of several related constraints. COLLECT-UML [5] is a CBM tutor to teach object-oriented design which supports both single user and multi-user for collaboration purpose. However, as Py. et al., [6] noted, instructional feedback directly generated from a violated constraint might not be a good solution from a pedagogical point of view. They separated the diagram diagnosis output from instructional feedback. However, they didn’t have much emphasis on diagram structure and how to generalize it.

Futrelle [7] attempted to apply levels of abstraction to diagrams by offering a diagram constraint grammar and process for automatic computational diagram analysis loosely based on computer vision. His approach, however, was focused on analyzing the diagrams, rather than tutoring using diagrams. Tutoring through a diagram not only needs to analyze the diagram, but also to understand the student’s knowledge and misconception within the abstraction in the diagram. Thus, a mapping between levels of abstraction in the diagram with domain-wide conceptual knowledge is highly desired. Here, we proposed a three-layer abstraction for diagrams used in engineering domain, where errors in lower layer need to be addressed first as it is more fundamental. Also, the three-layer abstraction follows a general process of knowledge acquisition: from superficial to deep, from rough to detailed.

It is worth mentioning the pedagogy in the Andes tutor [8] that allows students to pursue different correct solutions during problem solving instead of limiting them to a predefined optimal solution. A solution graph representation, which contains several types of nodes, is used to model all possible solution paths, upon which a Bayesian network is built. Then Bayesian inference is applied to designate student’s current goal node and a rule-application node where the student is stuck for lack of knowledge. A hint is then generated to coach that knowledge accordingly. Even though Andes focuses on text-based inputs, this step-by-step coaching strategy also applies to diagram-based systems. However, there are some differences that make pedagogy in diagrams challenging: 1) A diagram should be perceived as an entire state, no matter when and how an element is added to the diagram. Step-by-step coaching needs to be redesigned appropriately. 2) Even though sequence is less important in a diagram, it does require a series of actions to be applied in order to meet a certain requirement in a given state. This means that the diagram must be properly defined as several sequential stages, where each stage represents certain conceptual understanding. Within a stage, the sequence of actions do not likely matter.

3 Instructional guidelines for DITS

Guideline 1: Instruction needs follow hierarchical diagram understanding.
Even though diagrams vary across domains, there are usually underlying concepts that drive core questions that should be answered during the assessment process. The core questions can be defined through an expert module, which might vary based on
the expert’s instructional and pedagogical preferences. However, a general architecture that fits in a cross-domain evaluation system is highly desired, e.g., a version of the popular ontology editor Protégé customized for DITS authoring. For this purpose, we propose three levels of hierarchy for diagram evaluation. Before defining the levels theoretically, we offer an illustrative example from thermodynamics. Figure 1 shows an example with T-v diagrams. These diagrams are used to abstractly represent how pressures, temperatures, and volumes change within a mechanical refrigeration system, which may contain compressors, pumps, valves, etc. The Thermo Cycle Tutor basically asked six questions: 1) Is a vapor dome needed? 2) How many pressures are present in the cycle? 3) How is a pressure line drawn on a T-v diagram? 4) How should phase change P and T be labeled on the diagram? 5) What are the P, T, v relations for each component? 6) How can the problem information, and the decisions above uniquely identify each state?

While these questions are particular to refrigeration cycles, they have the following characteristic which applies across domains: some of them focus on the student’s conceptual understanding (1, 2, 5), and some focus on the procedural skill of how to make a diagram appropriately (3, 4). Of course, these two aspects are tightly coupled, and some questions apply to both (6).

It is noteworthy that the six questions follow a hierarchical understanding of the diagram. At Level 1, nine straight line segments are recognized on a vapor dome (Figure 1a), where each three connected segments represent a pressure line. At this level, the message that the diagram conveys is simply that there are three pressures in this system. At Level 2 (Figure 1b), more details are shown: some text labels are attached to the pressure line segments at the right-hand side, and tick marks are added to show the phase change temperatures. These additions give the viewer more concrete information about the exact value of the pressures and phase change temperatures. Then, at Level 3 (Figure 1c), by adding some points with labels on the pressure line segments, the diagram brings in details on the state information and how it interacts with the pressure and phase change temperatures.

![Fig. 1. Three levels in a refrigeration cycle T-v diagram.](image)

To generalize the levels just described, Level 1 focuses on basic graph-style structures and the spatial relations between each other. At Level 1, the tutor has a rough idea of what components are present and their connections. To give feedback at Level
The DITS needs to incorporate domain knowledge. Level 2 focuses on object identities and their object-type-specific relationships. At this level, attributes about an object will be identified through domain knowledge and some text labels. These include object name and possible values relate to object. Spatial relationships from Level 1 will be transformed to more specific numerical relationships. As is shown at Figure 1b, the numeric value has been explicitly shown in each pressure line, so it is easy to tell the second pressure is 0.67 (1 - 0.33) psi higher than the first pressure, whereas Figure 1a only tells the second pressure is above the first pressure. Level 3 focuses on properties or children of Level 2 objects. In this level, details on Level 2 objects will be revealed and examined. The details could comprise sub-objects that constitute a Level 2 object, or a sub-object that is attached to a Level 2 object but itself is not considered as a basic structure at Level 1. Instructional feedback can be composed based on the level of specificity. The lower level error should be tackled first, as it is more fundamental and serves as the basis of the higher level object. For instance, if a Level 1 object is missing, it doesn’t make sense to correct a Level 2 object as by definition its structure is based on Level 1 object. We propose this “divide-and-conquer” strategy where each piece can be mapped to one or more states of student’s understanding.

Guideline 2: Customized instruction from individual to individual.
A diagram embeds a student’s conceptual understanding, while evaluation by the expert module is trying to infer it. Thus evaluation questions need to be somehow mapped to domain-wide concepts. In order to track student’s knowledge on each concept, it is necessary to register them in student model. A complete set of evaluation steps will be applied to the student’s diagram at the beginning, as the domain-wide concepts in her student model is not determined. As she finishes a problem, her student model will get updated, with some concepts being checked as passed. How to define a concept as mastered is not in the scope of this paper. The next time, the expert module should consult her concept inventory before initializing the tutoring process. For example, we have implemented six questions in the expert module in the Thermo-Cycle tutor. However, for the student who has understood phase change temperature, how to use the reference form to locate the value, and how it should appear in a T-v diagram, expert instruction would skip question 4, which checks the label of phase change temperature in the future T-v diagram evaluation.

Guideline 3: Separate conceptual knowledge from procedure expertise.
As the evaluation engine assesses a student’s drawing based on the elements defined in the expert module and gives instructional feedback, there are some practical issues. How to handle these issues will affect the usefulness and quality of instructional feedback, student’s engagement, and finally affect learning gains.

In most cases, when a student starts to frame a problem, she doesn’t have a clear idea of what information needs to be drawn, and what might be a proper way to represent it. So a drawing with incomplete elements might be submitted to the tutoring system for help. In order to provide the most useful instructional feedback, the tutor is desired to “read” information from the drawing. The information includes what might
be her intention, what knowledge she might have known or not known and what other knowledge needs to be further determined from the drawing.

![Fig. 2. Examples of wrong drawings on refrigeration cycle T-v diagram. (a) and (b): incomplete pressure line. (c). Wrong pressure line representation which uses a negative slope.](Image)

The student might not be able to represent her conceptual understanding correctly in the drawing at the beginning. However, when she gets familiar with the procedure or gains expertise on how to represent the knowledge, she can focus more on the conceptual part. So a tutoring system needs to set apart these two types of questions, and give instructional feedback separately. Figure 2 (a) and (b) show two examples of sloppy drawing where a vapor dome and three coupled lines were present. To be considered as a correct representation of a pressure line, three connected segments should be included. However, the incomplete drawing still implies that the author thought there were three pressure lines. Assume three pressure lines are the correct answer in expert solution. In this case, the tutor’s instructional feedback should focus on how to help them to construct a pressure line, instead of correcting the number of pressures because zero "true" pressure lines were detected in the diagram.

Figure 2 (c) shows an incorrect representation of a pressure line since slope in the side lines should be positive. Many beginners tend to borrow the shape that they learned in P-v diagram, which is negative, and apply it to T-v diagram. Even though the tutor cannot detect pressure lines, it should be able to probe student’s intention as three pressures in the system, and give her appropriate instruction such as “This is not a P-v diagram. Would you like help on drawing pressure lines in a T-v diagram?” To facilitate this strategy, we provide guidelines for a DITS evaluation engine.

*Diagrams require different tolerances at different levels of evaluation.*

As we mentioned earlier, instruction could be based on evaluation of a three-level hierarchical structure of the drawing. A different tolerance could be assigned to each level. Tolerance could be a concrete value applying to check functions such as 10%, or it could have a conceptual definition, i.e., a pressure line can have two or three connected segments. A larger tolerance should be used to detect whether a basic structure is present. For instance, to check the number of pressures, a pressure line can be recognized if one horizontal line in the middle and two positive sloped lines at sides are found and well connected. The tolerance for gaps between line segments could be a large value. After it passes the Level 1 check, and goes to the next level, which checks the representation correctness, this tolerance would decrease, and any big gap would need to be filled by moving line segments closer to their neighbor. Another two examples with incomplete pressure lines (Figure 2a and 2b) would pass
the number of pressure check, given a higher tolerance on recognizing a pressure line. After a student passed the Level 1 check, the incomplete pressure line issue would be addressed due to a lower tolerance on how a pressure line should be drawn.

*Diagrams' inherent ambiguity can be resolved with Q&A.* Due to the intrinsic complexity and ambiguity of a drawing, it is safer to confirm the information that is conveyed in a drawing with some text inputs. For example, if the drawing fails on a *number of pressures* check, a multiple choice question pops up and asks the student to choose how many pressures are there in the system. If it is correct, it indicates that the student’s conceptual understanding is correct, but some procedural issue caused the failure, e.g., she accidentally clicked the submit button without finishing the pressure line. Another example is shown in Figure 3. The student had a good job on drawing pressure lines, labeling pressure and phase change temperature, and anchoring points on pressure lines to show the state changes in each component. However, feedback from the tutor said “There appears to be some misconceptions about the specific volume change in a compressor.” Then the tutor directed her to three multiple choice questions regarding pressure, temperature and specific volume change in a compressor. She answered all the questions correctly and was told to “modify state 3 and 4 to reflect this.” These successful answers imply that the student understood knowledge in a compressor, but didn’t incorporate it into the drawing.

![Fig. 3. (a). A refrigeration cycle T-v diagram. (b). Three windows that displayed questions about pressure, temperature and specific volume change in a compressor.](image)

*Conceptual and procedural performance in diagrams can be tightly coupled.* This problem is critical and stems from the fact that some aspects of constructing the drawing can make it difficult to edit elements later. This situation can frustrate a student if it occurs late in the problem solving process. As is shown in Figure 3, after the student realized state 3 should have a larger volume than state 4 (which means state 3 should appear on the right side of state 4 in the T-v diagram), it is impossible for her to change it in the diagram because there is no room. However, the student would not
realize this issue until she reached this step if she didn’t have much experience on solving this type of problem before. To alleviate this form of unnecessary frustration, when a particular problem is initialized by student, the evaluation engine should be able to load some practical expertise information about the base objects, e.g., the shape of the vapor dome should not be too thin and the distance between the horizontal lines should be greater than some percentage threshold.

4 Conclusion and future work

In this paper, we discussed cross-domain pedagogical strategies in diagram-based tutoring systems. In particular, instructional feedback needs to be mapped to a hierarchical understanding of the diagram. Personalized evaluation is desired which is based on student’s current knowledge state. Also, it should be able to separate conceptual knowledge from procedure expertise. To achieve that, we proposed: 1) allow different tolerances at different level of evaluations, 2) use Q&A to reduce ambiguity, and 3) determine if conceptual knowledge can be applied by procedure expertise in the current drawing. In the future, we will design a general authoring tool for DITS to support the above pedagogical strategies, allowing instructors to define a) concepts in the knowledge base, b) objects and tolerances in each hierarchical level, c) evaluation pieces which link to one or more concepts and d) guidelines of procedural expertise.

References