Enriching Students' Laboratory Experience: Using Software and Socratic Methods to Foster Reflective Thought in an Engineering Laboratory

Baba Kofi Weusijana, Christopher K. Riesbeck, Joseph T. Walsh, Jr.

Learning Sciences / Computer Science and Learning Sciences / Biomedical Engineering
Northwestern University – VaNTH ERC

Abstract

We have developed SASK (Socratic ASK[*]), a domain-independent and rule-based architecture for implementing Socratic dialogs to foster better learning in well defined tasks by encouraging deeper reflections by the student. We have used SASK to build the Dialysis Mentor, a program that uses Socratic questioning to improve student performance and learning in an undergraduate biomedical engineering lab. Small usability tests and a pilot run in a dialysis lab suggests that Dialysis Mentor and SASK systems in general can improve the value of pre-defined learn-by-doing task experiences. We are now working on improving our SASK Mentors[1] and building authoring tools for them.

Keywords

Learning Technologies, Rule-Based Intelligent Tutoring Systems, Socratic Tutoring, Inquiry-Based Education, Advanced Classroom Technologies, Interactive Learning Environments, Improving Classroom Teaching, Human-Computer Interaction, Authoring Tools, Multimedia Applications

[*] ASK is not an acronym. It is a technical term that we will define later in this paper.

Contents

- Overview
- Background
  - Improving a biomedical engineering laboratory course
  - The Socratic Method and PCK Expertise
    - Possible Solutions
    - Our Solution
- Description and Development of the SASK Dialysis Mentor
  - Specifying the Dialysis Mentor's Behavior: Rule-Based Linking
  - The Lack of Separate Student Modeling and Teacher Modeling Components
  - Socratic Dialogs and Rules
  - Handling of Natural Language Processing (NLP) and Human-Computer Interaction (HCI) Issues
    - ASK Systems
    - Overview of the Graphical User Interface (GUI)
    - Dialog Management and Button Theory
- Validation
  - Usability Testing
  - Pilot Testing
- Future Research Issues
  - Student-Computer Interaction and Interpretation Issues
  - Pedagogical Templates for Teachers' Authoring Tool(s)
  - Internet Accessibility
  - Conclusions
- Acknowledgements
- Bibliography
- Biography

Overview

This paper proposes that a feasible way to foster task-situated reflection in students during a lab is to provide them with a Socratic ASK rule-based tutoring software system.

Engineering teaching laboratories are predominantly inquiry-based environments and as such are ideally suited for the Socratic teaching methods. Experts in the field, e.g. professors, can readily raise a students' awareness toward problems and solutions through judicious questioning. Such questioning allows the student to find solutions and to develop expert-like thinking and problem-solving skills. Teaching assistants, who are usually graduate students, often lack the pedagogic training and broad content knowledge to use Socratic methods properly. Therefore, a lab needs an expert professor to assist students in their inquiry; however, in a large lab, such experts are a scarce resource.

We designed a Socratic software tutor that (while the professor was not available) prompted student inquiry, guided the discovery process, and fostered students' in-task reflection through a series of questions. The software used a graphical user interface and conducted dialog with the students, asking questions about their process and progress. A significant challenge was the development of the rule-based system's dialog paths to responded appropriately to student responses. The system was tested in a dialysis laboratory class of 52 students. Survey data, student user tests, in-lab videos, and the captured computer-student interactions indicate the effectiveness of the system as well as opportunities for further system improvement.

Background

Improving a biomedical engineering laboratory course

This work is part of a VaNTH ERC[†] initiative to develop technology to improve bioengineering education. Specifically our goal is to improve a Biomedical Engineering Laboratory course at Northwestern University that is designed to give advanced undergraduates in biomedical engineering experience in:

- performing measurements of biological systems
- designing experiments that yield quantitative results
- analyzing and presenting experiment results in oral and written form

The course consists of six labs. The lab topics cover equipment capabilities and instrumentation limitations, how to use an electrocardiogram (ECG), how dialysis works, sodium transport, muscle mechanics, and pulmonary function. We have focused on the dialysis lab in which students evaluate the ability for an artificial kidney to transport water, urea, and salt across its membrane. Students work together in groups of two or three.

The course's faculty has found that groups had many questions and problems. These included having trouble using the lab devices, not knowing how to interpret measurements, not understanding how one subsystem affects another, not remembering how to conduct scientific

[†] VaNTH is a National Science Foundation engineering research center (ERC) composed of the bioengineering and learning sciences faculties of Vanderbilt University, Northwestern University, the University of Texas at Austin and the Health Science and Technology Program of Harvard and MIT. See http://www.vanth.org for more information.

research or having trouble applying research principles, and failing to grasp the theoretical goal of the current experiment. The course's faculty often handled such problems he noticed, or questions from students, by using the Socratic method.

The Socratic Method and PCK Expertise

The Socratic method is a pedagogical technique in which a teacher tends not to give information directly but instead asks a series of questions. By answering the teacher's questions, the student eventually discovers the desired knowledge or gains a deeper awareness of the limits of knowledge[‡]. The name of the method originated from Socrates' use of it in the Meno dialog of Plato, but it has other names, such as the inquiry method or the case method[2]. A classic Socratic technique is tracing the consequences of expressed student beliefs to a contradiction. This forces students to actively debug their own theories, perhaps preventing them from making similar mistakes in the future. It also teaches an important scientific practice of testing the consequences of a theory in order to evaluate it[3]. There are also proven psychological benefits to knowledge that is generated by students instead of just received by them. The teacher can also gain great insight into how instruction is being received and applied[4].

Since he first taught the Biomedical Engineering Laboratory course in 1989, Dr. J. Walsh has used the Socratic method by asking students questions that would help them discover the information they needed or lead them to uncover their own misconceptions. For example, if Dr. Walsh can see that a student group is varying more than one parameter at once, which destroys the ability to isolate the cause of any effects, he often asks a question like "So, what are you varying and keeping constant here?" During their response, students may realize their mistake. If not Dr. Walsh might pursue the issue further, asking questions like “Well if you change X and Y, how do you know which change had an effect on Z?”

That sort of technique was hard for Dr. Walsh to instill in his graduate assistants. Not only does it require a broad understanding of the domain, but also one must be able to resist the urge to simply tell students the answer they should be looking for. Even worse, Dr. Walsh says that graduate assistants often know how to complete a learning task in only one way. Thus, they may tell students that any other way to complete the learning task is unacceptable. However, there are often alternate paths to successfully completing a learning task. Unlike him, Dr. Walsh’s graduate assistants were not yet pedagogical content knowledge (PCK) experts in regards to teaching about dialysis.[§] Unlike those who are only experts on the content of a domain, PCK experts also have specialized and advanced methods they use to teach a particular domain. PCK experts know about typical difficulties that their students encounter as they attempt to learn about a set of topics. PCK experts also know the typical paths their students must traverse in order to achieve understanding[5]. This is different from teachers who are general pedagogical experts. People who are general pedagogical experts are skilled in educational methods in general, but might not know a domain deeply. PCK experts are also different from content experts who know a domain deeply, but may not know how to teach it to their students.

[‡] *The American Heritage® Dictionary of the English Language, Fourth Edition* (2000) Houghton Mifflin Company.

[§] Dr. Walsh is not an expert on PCK expertise, rather he is a PCK expert on his dialysis lab.

Type of Teacher	Characteristics
Content Knowledge/Domain Experts	- Knows the content knowledge/domain deeply and broadly.
Pedagogical Knowledge Experts	- Knows general educational methods.
Pedagogical Content Knowledge (PCK) Experts[6]	- Knows the content knowledge/domain deeply and broadly. - Knows general educational methods. - Knows the problems their students typically have with the domain. - Knows the preconceptions (and misconceptions) their students have. - Knows the different paths their students should typically traverse through the domain, and the different helpful representations of the domain a student should see to gain a desired level of understanding.

Table 1: Some different types of teaching expertise.

Because Dr. Walsh was the only PCK expert on the dialysis lab available to the students working in the lab, students needed his Socratic tutorship more often than he could give it. Sometimes groups would wait 20 minutes or so for his guidance before they could make any progress. Sometimes groups wasted time doing something unfruitful and would have to continue the lab during the next class session. Even worse, sometimes a group would get too much information from a graduate assistant and may finish a lab without gaining the understanding Dr. Walsh had hoped they would.

Possible Solutions

Our project's goal is to solve the problem of a lack of tutorship (specifically partially Socratic tutorship) being available to all lab groups for the entire duration of the lab sessions. We believe that this is a problem not just in Dr. Walsh's biomedical engineering course, but in many laboratory classrooms. We believe human experts with PCK expertise in the relevant domain who are skilled in Socratic tutoring techniques will often be a scarce resource. Without such attention from teachers, students tend to rush through their lab. Even if they succeed in finishing the lab with a passing grade, they likely miss many opportunities to reflect on their thinking while they are doing their lab work. This reflection activity, and other meta-cognitive practices, increases the degree to which students will transfer their learning to new settings and events without the need for explicit prompting[5]. Thus, we need to increase the occurrence of reflective activity situated in the students' task, thereby improving the students' ability to perform analogous tasks in the future (in this case scientific lab research after college).

One possible solution is to provide an artificial teaching system, a virtual Dr. Walsh that mentors students in much the same manner he does. A software based system could be installed on the computers being used in the lab for data collection.

One type of software system that could be appropriate for such a case as the dialysis lab is a knowledge-based Intelligent Tutoring System (ITS). Typically such systems have a knowledge-base (a repository of the content and relationships within the domain), a pedagogical component (for general teaching expertise modeling), and a student modeling component (to keep track of what the student probably knows and is thinking). Some good examples of knowledge based

ITS programs are AutoTutor[7], GUIDON[8 9], and WHY[2 10]. However, building the knowledge base would have been a costly and lengthy development process. Since many ITS systems require knowledge to be expressed in the form of predicate rules, it might not have been possible for Dr. Walsh to explicitly enter all the needed knowledge into such a system. In the solution we chose, it was not necessary to build that type of knowledge base for such a learning task that is not very open ended and where the dialog between students and the teacher is well known.

Our Solution

Our solution to these problems is to provide a Socratic software mentor[**] capable of providing a similar service that Dr. Walsh provides his students. We are providing a rule-based system that we believe has an easier method of providing it with domain and pedagogical knowledge than many traditional knowledge based ITS programs. Our plan was to install it on personal computers previously used only for lab data collection. Students can interact with these mentors and get assistance when their professor is not available. Through that mechanism each student group in the lab can get assistance directed to their current needs, at any time they need it. The system being Socratic enables it to help students debug their own thinking and knowledge in regards to the lab. This is preferred in comparison to didactically providing all the commonly needed information and warnings needed to successfully finish the lab. Such a presentation of information would likely be boring for the students. This is because students would have to memorize information before they have a compelling need for it. Additionally such information would be easily, and perhaps even purposefully, forgotten after the lab was completed. Even a list of frequently asked questions (FAQ) and their answers, or a topic-based help system, or a simple ASK system (discussed later) would not be as effective as a system that is at least partially Socratic. There is some evidence that students tutored Socratically have a higher average gain between pretests and posttests than students tutored didactically[11].

Description and Development of the SASK Dialysis Mentor

The prototype software system we built, and used in the fall quarter of 2001, is called the Socratic ASK (SASK) Dialysis Mentor (DM). The SASK DM is a stand-alone computer program written in the Java programming language.

Specifying the Dialysis Mentor's Behavior: Rule-Based Linking

The program’s behavior is specified in a file that has a graph[††] of the entire Dialysis Mentor’s utterances and the utterances the students might make. We call this file the task dialog graph document. It is in an XML format. The current DM graph has 89 utterance nodes and 153 edges (rules that link student responses to DM utterance nodes) in it. To build this document, we collected data on how the dialysis lab is usually taught. Dr. Walsh was video taped lecturing a biomedical engineering undergraduate in a classroom and later tutoring the student in the dialysis

[**] We use the term "mentor" instead of "tutor" because many tutoring systems are strongly didactic and teacher centric. Mentoring implies that we are trying to provide a student centered, student expectation failure driven, inquiry activity.

[††] We are using the computer science definition of graph here, not the geometric one. Our graph consists of speakers' utterances linked by rules.

lab. The student was not exposed to the course's content before these experiences. The tapes were digitized and the laboratory tutoring session was transcribed. We analyzed the dialog and categorized the data into sections based on Dr. Walsh’s pedagogical goals. We then conjectured on how the conversation would have continued if the student had made other probable responses, requests, and mistakes.

In general, there are four types of occurrences that lead to Dr. Walsh's Socratic questioning:
1 Dr. Walsh asks for an update from the student (such as "What are you doing?", "What are you measuring?", or "What are you controlling and what are you varying?").
2 The student asks a question to confirm what their current activity should be.
3 The student asks a question to confirm if their conjecture is correct.
4 The student asks a question about how to use the laboratory equipment.
Dr. Walsh asking for an update was far more common than any questioning from the student. This implied that we should occasionally prompt a student to let the Dialysis Mentor system know where they are and what they are doing, and that students should also have the option to ask questions the system can currently address.

Dr. Walsh uses a mixture of Socratic and didactic educational methods. Although he would usually ask a student for an update and then lead them in a dialog that would help the student debug their thinking, occasionally Dr. Walsh would have to issue instructions or information. This was usually to warn the student about unsafe consequences of their actions or take them off an unfruitful path that would delay the group from finishing the lab within the time allotted. Certainly, we do not want to teach so Socratically as to never offer information to students, leaving them to vacillate long past the temporal confines of the lab session or exposing them to danger. Like Dr. Walsh the system we designed does occasionally cut a line of conversation short by telling the students they are mistaken and why.

Before we show you how we analyzed Dr. Walsh's tutoring and specified the behavior of the DM accordingly, we must tell you more about the dialysis lab itself.

In the dialysis lab, students are to learn principles involved in transferring blood through the filter of an artificial kidney (a dialyzer), thus replacing some of the cleansing functions of a kidney. Each student group is given a lab apparatus similar to the sketch below.

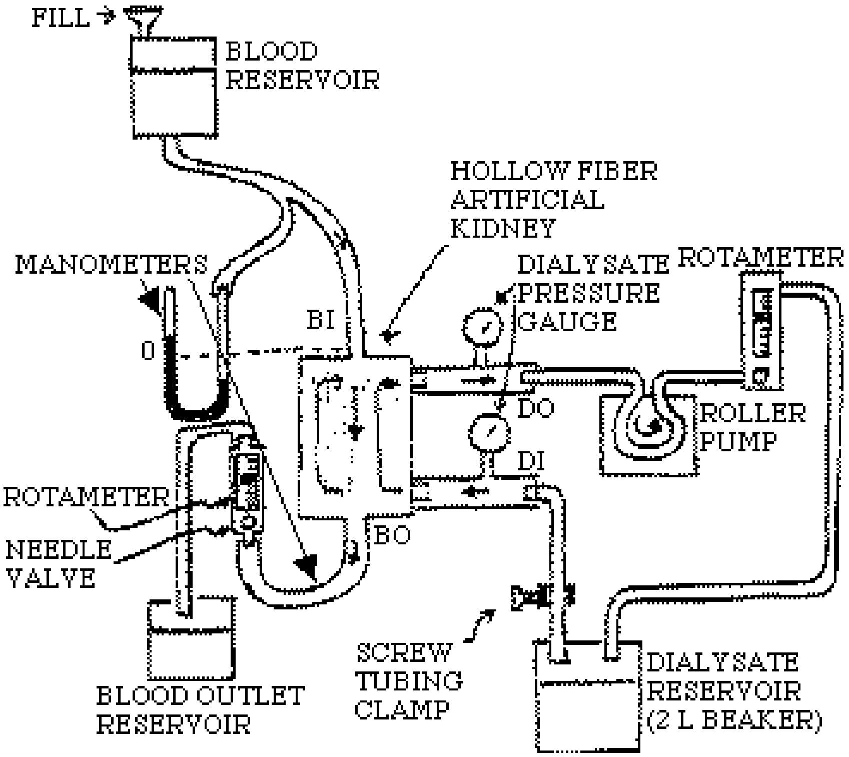


Figure 1: A sketch of the dialysis lab apparatus

On one side of the system, blood (which in the lab is just water) flows from the blood reservoir through the hollow fiber artificial kidney to the blood outlet reservoir. On the other side, water (called dialysate) is suction pumped through the artificial kidney. The blood and the dialysate only interact inside the dialyzer when certain toxins filter from the blood through membranes inside the hollow fibers into the dialysate. These toxins, called urea, filter through because the pressure on the dialysate side is kept lower than the pressure on the blood side. This process of forcing material through the filter with a hydrostatic pressure differential to called ultrafiltration (U). The average of pressures measured on the blood side of the dialysis system minus the average pressure on the dialysate side is called the transmembrane pressure (TMP). Ultrafiltration has a linear relationship with TMP characterized by the equation

$$U = Pwater \times A \times TMP$$

Where Pwater is the permeability of the filter's membrane to water and A is the area of the membrane. Pwater x A is called the ultrafiltration factor and is a quality characteristic of a dialyzer. Then higher the ultrafiltration factor of a dialyzer the more it can filter at a given Transmembrane Pressure.

Students are asked to make a graph of ultrafiltration over transmembrane pressure and find the ultrafiltration factor of the dialyzer they are given. By doing this, they are to learn the following concepts:

- How and why a dialyzer works.
- How could an engineer improve a dialyzer's quality or a doctor assesses a dialyzer's quality.
- How Ultrafiltration works via a pressure differential and not by a flow rate differential.
- How to find a dialyzer's ultrafiltration factor by only changing TMP.

The following figure shows part of a transcript of Dr. Walsh assisting a student in his dialysis lab. Dr. Walsh's goal at the beginning of this transcript is to get the student to understand what parameters should be controlled and what should be varied.

Dr. Walsh	Yes...so, then I'm gonna ask you...so which controls the ultrafiltration? The transmembrane pressure or the flow rate?
student	Indirectly the flow rate.
Dr. Walsh	How do you know its not indirectly the transmembrane pressure?
student	[pause]
Dr. Walsh	So you have to set up your experiment to tell me which parameter controls ultrafiltration. Transmembrane pressure or flow rate?
student	Ok.... transmembrane pressure.... because it's directly related to the equation you gave me.
Dr. Walsh	Right...right, but that's the theory. That's the theory that we presented. But now we're in lab, and I want you to prove to me that it's transmembrane pressure that controls the ultrafiltration and not flow rate that controls ultrafiltration.
student	But the only relationship we have is through the theory.
Dr. Walsh	Right
student	Unless you wanna see that this thing [the monometer pressure reading] for flow rate and plot that against ultrafiltration, so that there's not an indirect relationship. You need to show that pressure is part of the flow rate. Is that right?
Dr. Walsh	What I'm saying is.... I wanna know what controls ultrafiltration. Is it transmembrane pressure or is it flow rate? And my point is...if you vary both of them..
student	You're not gonna get...
Dr. Walsh	at the same time, you don't know. So you're gonna have to figure out how to run this experiment so you vary only one parameter at a time.

Figure 2: Partial Transcript of Dr. Walsh Assisting a Student in His Dialysis Lab.

Based on such transcripts we made a graph of possible utterances (questions and statements) that students and the teacher might make during lab dialog. Dr. Walsh reviewed our dialog graph and improved upon it.

The following figure shows a representation of part of our dialog graph document. This piece is the result of analyzing and codifying some of the transcript in Figure 2 above and hypothesizing on other possible paths the conversation could have taken. Notice how the students can be Socratically lead to test a hypothesis about a factor that is related to a dependent variable and thus discover a mistake in their thinking. This is just the sort of dialog that draws students into reflecting on the learning task at hand.

DM Utterance:

"How do you know U is not directly controlled by the TMP?"

Possible Responses:

1: "We do not know."
2: "We know U is directly controlled by ____ based on _____."

DM Rule:

If the students choose the response #2 and field #1 contains "TMP" or "Transmembrane Pressure",

then DM says

" That's the theory we presented. You must use the lab to prove that TMP controls U and not flow rate. How do you change only TMP?"

DM Rule:

If the students choose the response #2 and field #1 does not contains "TMP" or "Transmembrane Pressure",

then DM says

"Please set up your experiment so you can prove what controls U, TMP or flow rate."

Figure 3: A Representation of Part of the Dialysis Mentor's Task Dialog Graph Document

Although the Dialysis Mentor's dialog is a large graph of question and response utterances linked by rules, we found it easier to develop and think about the task dialog graph by dividing it into stages. Based on Dr. Walsh's expressed and implicit pedagogical goals we partitioned the Dialysis Mentor's task dialog graph document into six stages:

1. Measure Blood Pressure Stage: How to measure blood pressure. This stage is first for mercury safety reasons.
2. Main Goal Stage: Getting students to understand the point of the experiment: Graphing transmembrane pressure over ultrafiltration to find the linear relationship between transmembrane pressure and ultrafiltration. This allows students to calculate the ultrafiltration factor, a measurement of the effectiveness of the dialysis machine. Students should realize, and perhaps prove, that transmembrane pressure (not any flow rate of fluids) effects ultrafiltration.
3. Vary Stage: How to vary dialysis pressure and why (while controlling blood and dialysate flow rates as well as blood pressure).
4. Measure Ultrafiltration Stage: How to measure ultrafiltration
5. Calculate Ultrafiltration Stage: How to calculate ultrafiltration
6. Measure Dialysate Pressure Stage: How to measure dialysate pressure properly to get the right transmembrane pressure measurement.

Conversations tend to flow from one stage to the next, but student questioning or responses might cause the system to jump into another stage out of order, perhaps later returning to the main conversation topic. No stage should be repeated unless a student response or question requires it. Only stages 2 and 3 were implemented for use in the fall 2001 lab.

The Lack of Separate Student Modeling and Teacher Modeling Components

Many intelligent tutoring systems have a student model component that represents the cognitive state of an individual student interacting with the program. Usually another diagnosis module builds and augments the student model based on observable behavior and cognitive theory[12]. This is particularly true for ITS systems that use a component for representing the domain being taught. Such systems are called knowledge-based ITS systems and the component for representing the domain is called a knowledge base.

Our system is deliberately simplified. It lacks a knowledge base component that contains only domain knowledge. Our rule-based system represents possible dialogs a teacher could have with a student. We organized our system in that manner because building a knowledge base can be a timely task that not all professors, even PCK experts, can do. Building a knowledge base requires some scheme for entering all of the relevant words, facts, concepts, and relationships into the system, usually in an awkward and context free way. Just as content is entered for the knowledge component, teaching strategies are entered into a pedagogical, or teacher modeling, component. All this entering of input into an ITS is often very difficult. For instance, it is hard for a teacher to describe, particularly in a predicate logic format, the characteristics of a teaching strategy divorced from the actual cases of teaching.

Our method of providing our system with content, pedagogical, and student knowledge is to ask the professor to provide and elaborate on how a dialog with a student progresses. What are the statements and questions the professor will usually start with? How do students typically and atypically respond to such utterances? How might the professor continue the conversation? Therefore, we have the professor build a graph of possible dialog paths that he or she feels are helpful to the student's learning. We think, and experienced with Dr. Walsh, that professors who had many of these conversations find it relatively easy to reproduce and talk about them.

Here is an example of how it can take a great deal of time to program a knowledge-based ITS with a new domain. AutoTuor is a knowledge based ITS designed with a knowledge domain on computer literacy. Its knowledge base was later converted to a new physics domain. To give AutoTutor the ability to tutor 3 physics questions/problems this required

- Changing the glossary of terms and definitions (about 15 work hours)
- Supplying text books, chapters, and technical articles after converting them into a clean electronic form (about 10 work hours) and declaring certain parameters (less than 1 work hour)
- Supplying a curriculum script (in English) with deep reasoning questions and associated answers for physics (about 20 hours)

That is over 45 work hours to handle 3 physics questions/problems[7]. Our hope is to reduce that amount of time considerably. Currently our process involves video taping the human mentor and student(s) (4 hours for the entire dialysis lab), transcribing the tapes (about 8 hours for the dialysis lab), analyzing the transcripts and authoring all the relevant possible dialogs in the task dialog graph document. The amount of time we used for those final stages is not certain because were designing the software and building the dialysis dialog graph at the same time. However, if we can keep the length of those final stages below 33 work hours we will be superior to AutoTutor in terms of domain development time, and we will have a system that can handle 2 stages of dialog covering 1 to 1 1/2 hours of lab time. We will talk about how we plan to reduce domain development time in the section on authoring tools.

Socratic Dialogs and Rules

Dr. Allan Collins of Bolt Beranek and Newman Inc. has written about tutorship using the Socratic method in a number of papers. Collins analyzed the dialog between students and their Socratic tutors (the students were learning about geography). From this research Collins developed a theory for the implementation of a Socratic-tutoring system called WHY. The WHY software had four components that have become classic in ITS systems. It had a knowledge-base (domain knowledge component), a student model component, a simple pedagogical component, and a user interface[10]. Collins improved its ability to effectively teach students using a body of 24 production rules to improve WHY's pedagogical component[2]. We used some of these rules as a formal way to understand Socratic techniques, but we did not use them explicitly in the SASK Dialysis Mentor. For instance, the partial representation of the task dialog graph in Figure 3 is an example of Collins's Socratic Rule 15: "Request a test of the hypothesis about a factor", but we did not build that pattern based on Rule 15. We built it based on transcripts of Dr. Walsh teaching and discussions with him. We may use Collins' rules in an explicit manner in an authoring tool we are working on, but we will discuss that in the "Future Research Issues" section of this paper.

Unlike WHY, our system is not a knowledge-based system that needs a pedagogical component to improve its tutoring abilities. Similarly to Collins, we studied the dialog between Dr. Walsh and a student. However, instead of building the kind of components found in WHY (knowledge base, pedagogical component, and student modeling component), we found it easier to model the partially Socratic dialog in a task dialog graph document that mirrors the interaction of Dr. Walsh mentoring a group of students in his lab [‡‡]. This document is a graph of the relevant and probable tutor-student group interactions for the lab. That graph was codified into the domain document that is read by our system and used to direct its behavior. This means we have done the diagnosing in advance of interaction with students based on observed and probable student behavior. Both models of the teacher and students' cognitive processes are embedded both implicitly and explicitly in our dialog graph, along with the relevant domain knowledge.

A drawback to our design versus the common student modeling ITS design is that we cannot develop the domain knowledge separately from the pedagogical knowledge. In our system, we cannot have domain experts build a knowledge-base and then have pedagogical experts program our system with the abstract rules for teaching (such as the Socratic rules Collins developed). Since we do not have those components separated we cannot reuse one without the other. People who just want information about dialysis cannot query our system. People who just want a Socratic tutor cannot make a new knowledge-base and use it in our system. Also our system cannot respond dynamically to completely unpredicted student responses besides asking the student to rephrase their response, choose a different response, or start over at the beginning of a dialog stage. Thus, our system is limited to learning tasks that are not open-ended, where the dialog is well known to the experienced teacher.

David Littman and Elliot Soloway of Yale University have listed three common objections to

[‡‡] We did this even though different individuals respond at different times, by relying on the assumption that the students in the group are generally "on the same page."

doing a detailed evaluation of students' cognitive processes[13], a step necessary for us to build student models common now in knowledge-based ITS:

1. It is too hard.
2. We might use the wrong student models in our evaluations.
3. Student's internal representations are irrelevant to education

Although we subscribe to objection #1, our primary objection to doing student modeling for our task is closer to #3. When effective teachers such as Dr. Walsh are tutoring a lab, we do not believe they are doing the same sort of deep student modeling that many ITS systems are doing. They may have began their practice by creating and using such student-models, but as time passes we believe PCK experts tend to rely on a memory of what utterances are good to say after certain student utterances. This may be particularly true for Socratic teaching. To perform Socratic tutoring both human and artificial tutors must know how to get students to test their conjectures and face a contradiction they had not foreseen. The emphasis is on teaching students methods of handling problems that are well known to the PCK expert.

Handling of Natural Language Processing (NLP) and Human-Computer Interaction (HCI) Issues

To meet our design goals, we knew the Dialysis Mentor must have a means of collecting student input and expressing itself it in a manner as close to human mentor and human student conversational interactions as possible. If we allowed students to type in full English sentences in response to our questions, we would have been faced with the unsolved problem of getting the system to understand the text without having all the experiential and cultural background a human mentor brings to bear. Students could ask too many questions that the system cannot handle. It would be very frustrating for the students to be typing in lots of sentences that are often rejected for being improperly formatted, or worse misinterpreted, making the tutoring session incongruent to the students' needs. We needed easy to use methods for students to communicate responses the system could address.

ASK Systems

At Northwestern University in the Institute for the Learning Sciences (ILS)[§§] a series of programs have been developed called ASK systems. ASK systems are a form of hypermedia based on the metaphor of having a conversation with an expert, or a group of experts. During a conversation the user provides questions and the ASK system responds with answers. As in a real conversation, both participants in an ASK system influence the flow of discussion. The user influences the flow by selecting which questions to pursue and the ASK system influences the flow through the answers it provides[14]. When we started working on the Dialysis Mentor we postulated that it should be implemented as a Socratic ASK system (or SASK for short). The user still influences the discussion, but the tutor responds and directs it in a Socratic manner. Thus, what we needed from the system's user interface was a method of allowing students to answer questions, occasionally ask questions, and express the need for a change in the flow of a conversation.

[§§] The ILS institute has been dissolved, but the Learning Sciences Program and the Computer Science department are still pursuing much of its research.

Overview of the Graphical User Interface (GUI)

When students open the program and (dismiss the splash screen), they can choose from its two modes, “Overview” or “Lab Mentor”.

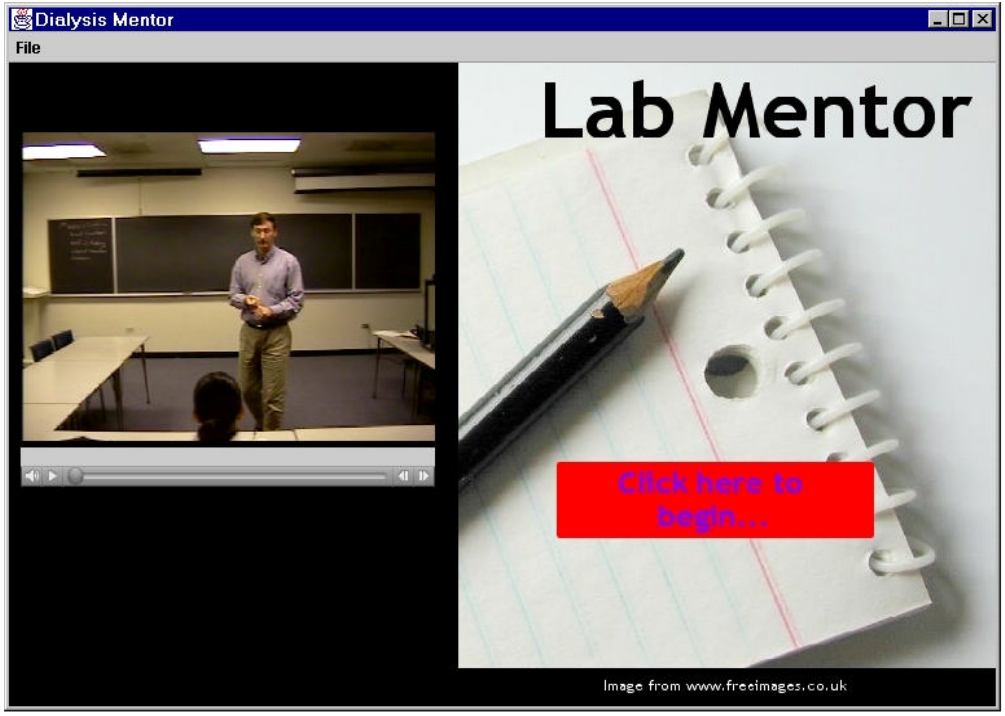
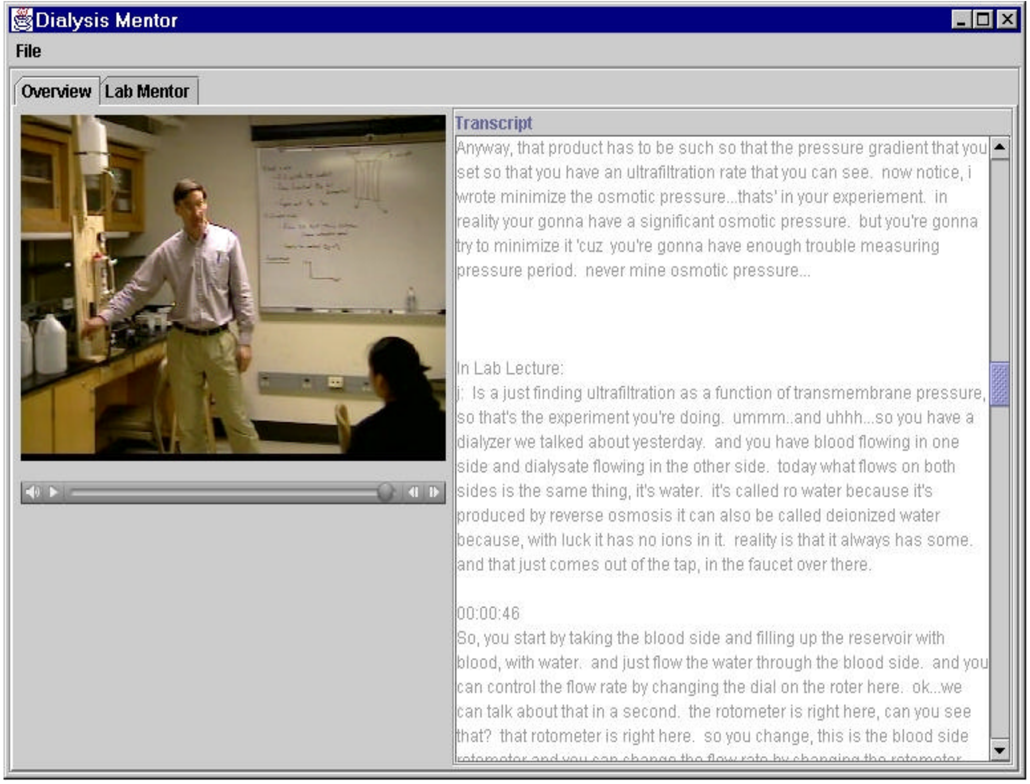


Figure 4: Splash Screen



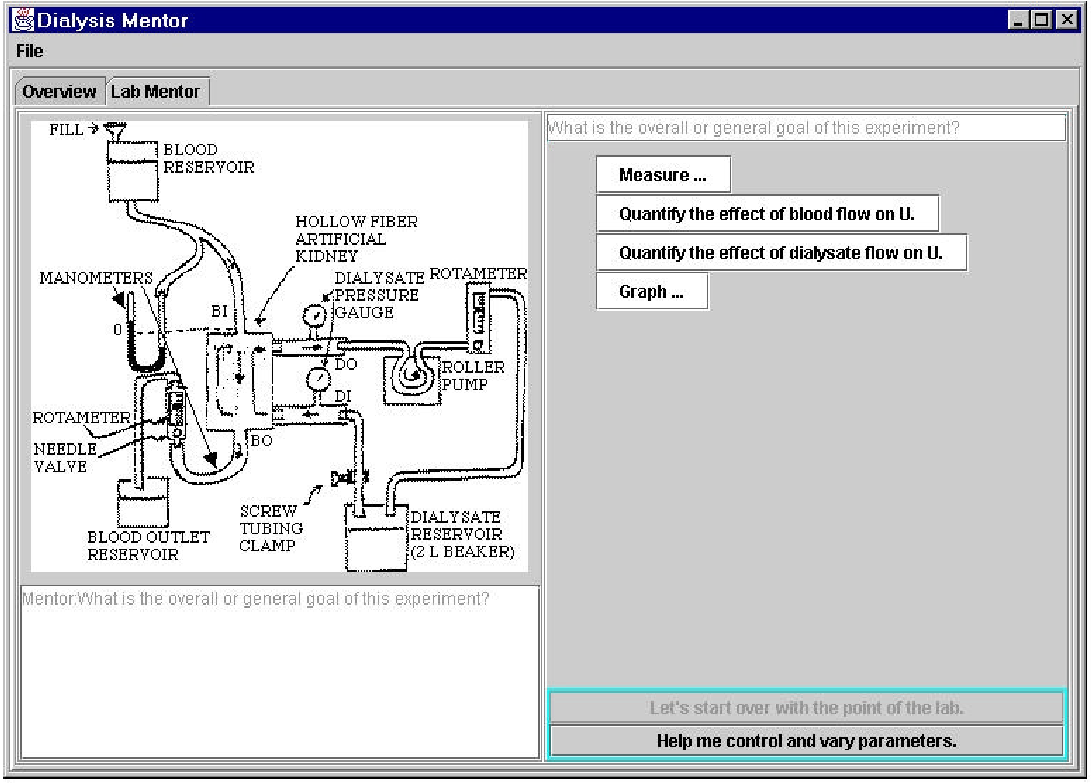
Anyway, that product has to be such so that the pressure gradient that you set so that you have an ultrafiltration rate that you can see. now notice, i wrote minimize the osmotic pressure...thats' in your experiement. in reality your gonna have a significant osmotic pressure. but you're gonna try to minimize it 'cuz you're gonna have enough trouble measuring pressure period. never mine osmotic pressure...

In Lab Lecture:
j: Is a just finding ultrafiltration as a function of transmembrane pressure, so that's the experiment you're doing. ummm..and uhhh...so you have a dialyzer we talked about yesterday. and you have blood flowing in one side and dialysate flowing in the other side. today what flows on both sides is the same thing, it's water. it's called ro water because it's produced by reverse osmosis it can also be called deionized water because, with luck it has no ions in it. reality is that it always has some. and that just comes out of the tap, in the faucet over there.

00:00:46
So, you start by taking the blood side and filling up the reservoir with blood, with water. and just flow the water through the blood side. and you can control the flow rate by changing the dial on the roter here. ok...we can talk about that in a second. the rotometer is right here, can you see that? that rotometer is right here. so you change, this is the blood side rotometer and you can change the flow rate by changing the rotometer

Figure 5: Overview Mode

The “Overview” mode is a reference utility that allows students to review the background information on the laboratory in case they have forgotten something. Students can watch a video of Dr. Walsh giving a lecture to a student regarding his dialysis lab, similar to one they themselves should have attended. This was accomplished using QuickTime for Java. Students can also read a transcript of the lecture. Students should also have on hand their written lecture notes and laboratory instructions from their course packet.



What is the overall or general goal of this experiment?

Mentor:What is the overall or general goal of this experiment?

Dialysis Mentor
File
Overview
Lab Mentor
Measure ...
Quantify the effect of blood flow on U.
Quantify the effect of dialysate flow on U.
Graph ...
Let's start over with the point of the lab.
Help me control and vary parameters.
FILL
BLOOD RESERVOIR
HOLLOW FIBER ARTIFICIAL KIDNEY
MANOMETERS
DIALYSATE PRESSURE GAUGE
ROTAMETER
BI
0
DO
ROLLER PUMP
DI
ROTAMETER
NEEDLE VALVE
BO
SCREW TUBING CLAMP
BLOOD OUTLET RESERVOIR
DIALYSATE RESERVOIR (2 L BEAKER)

Figure 6: Lab Mentor Mode

In the “Lab Mentor” mode students can be engaged in a dialog with the program and thus get assistance with the dialysis lab in a similar manner as they would from Dr. Walsh. Currently this mode includes a diagram of the laboratory set-up (upper left). In the diagram, each apparatus is labeled so students know what terms the program will likely recognize. In future, we may include video clips of Dr. Walsh speaking as the Dialysis Mentor with the students. Also in this mode is a transcript of what the Dialysis Mentor and the students have said (lower left corner). That transcript can be saved to a text file via a command under the “File” menu.

Dialog Management and Button Theory

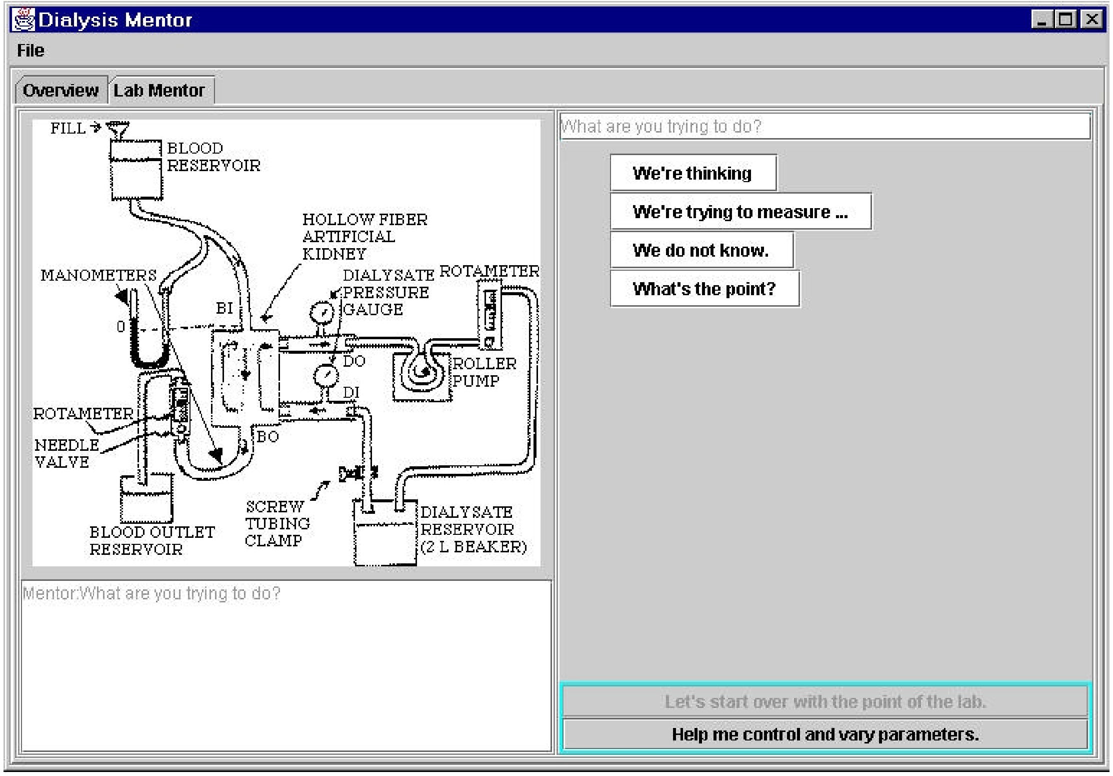
In his Ph.D. thesis at Northwestern University, Dr. Daniel Edelson explained how his program Creanimate, designed to teach elementary school children about animal adaptation, uses a sub-set of Socratic techniques[15]. We used some of Edelson's dialog management methods and Human Computer Interaction (HCI) techniques to allow the Dialysis Mentor to hold a conversation with student groups via a easy to use and not overly constraining graphical user interface.

Creanimate encourages students to make hypotheses and guides them through a exploration of them with thought-provoking questions. Creanimate's dialog manager does this by using a pattern Edelson calls the dialog cycle which includes the following steps:

- The student picks an animal and proposes adding a new feature to it.
- Creanimate asks for a possible reason the proposed feature would be useful to the animal.
- After the student responds, Creanimate tries to present a video example reminding the student of some animal who has a similar feature for a similar reason.
- After the student commits to adding a feature, Creanimate shows a picture of their new animal and then the process can begin anew.

Although Creanimate adheres to the basic dialog cycle, six different dialog sections characterize the exact nature of the interaction. Each dialog section is tailored to a different type of student input. Edelson also uses Button Theory (explained below) in his user interface. Creanimate is thus a system with a simple major cycle but with ample variation and opportunities to get help and the big picture. This was so young students would find it easy to use and not overly constraining.

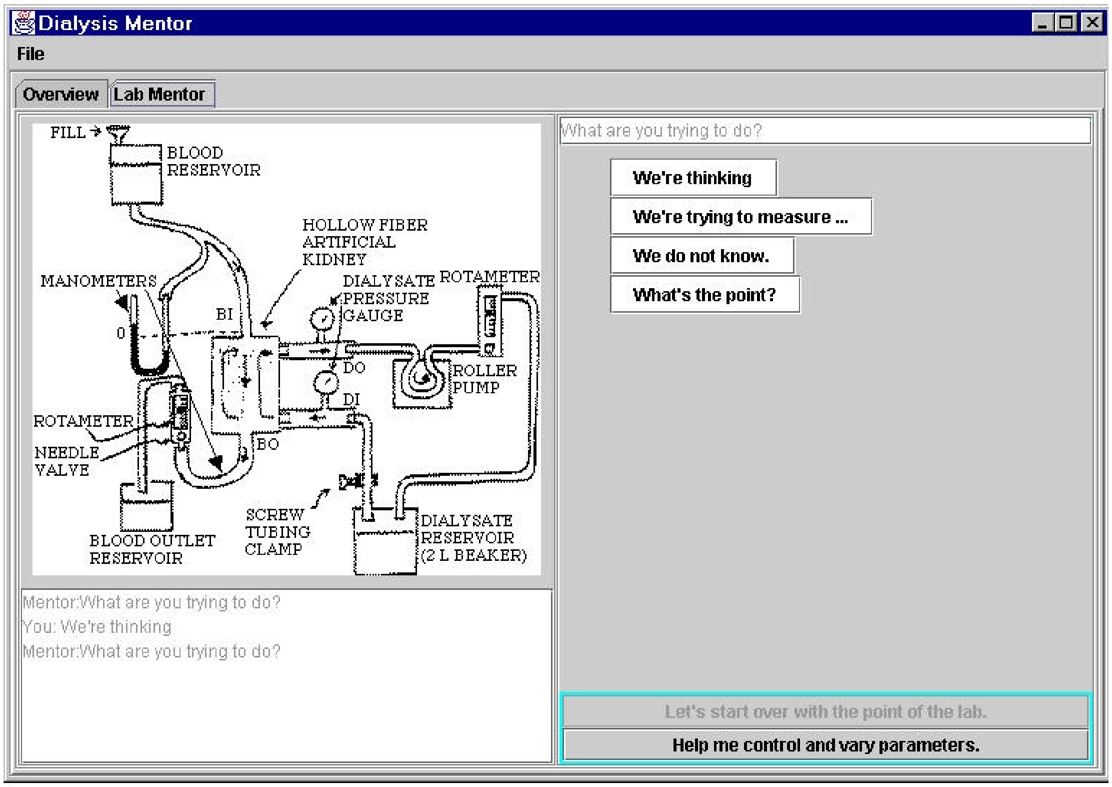
When Creanimate asks a student a question, it provides templates for the student to use in their response. Our system does the same. When we ask a question like "What are you trying to do?", we do not give the students just a blank text field to enter their response for them. We provide three choices in the form of buttons that say "We're thinking", "We're trying to measure…", "We do not know.", and "What's the point?".



Mentor:What are you trying to do?

Figure 7: Before Student Presses "We're thinking" Button.

If students click on "We're thinking" DM waits a couple of minutes and then asks the students the same question again.

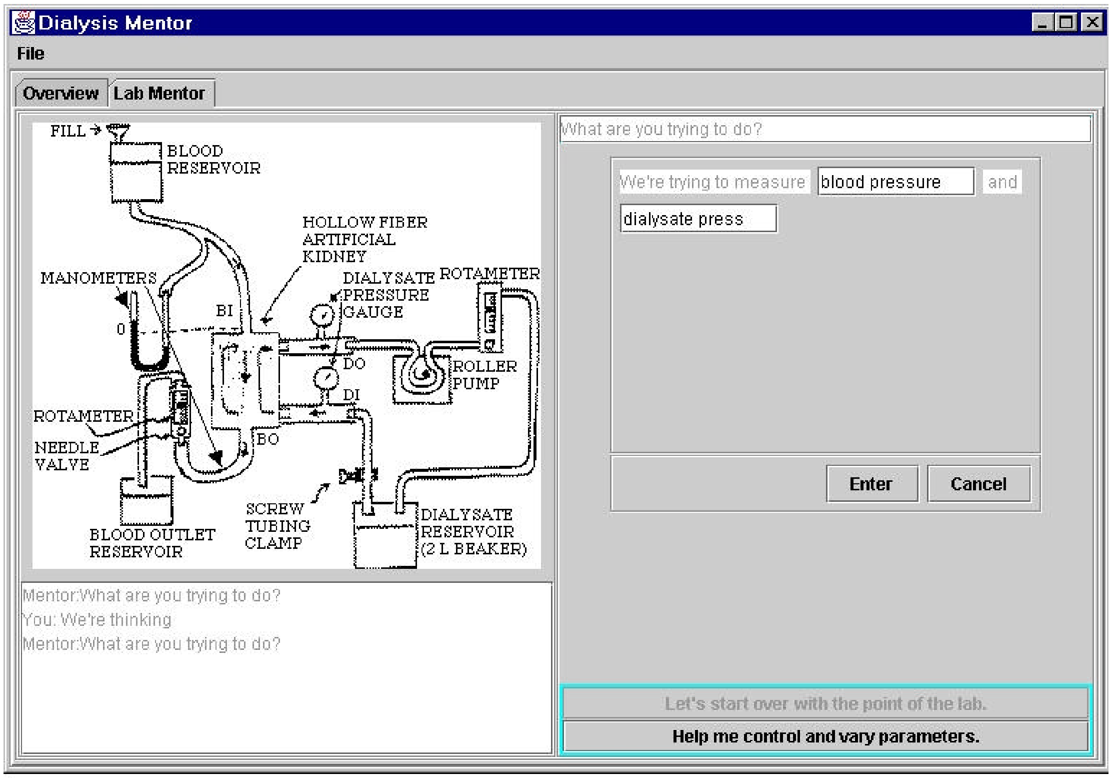


What are you trying to do?

Mentor:What are you trying to do?
You: We're thinking
Mentor:What are you trying to do?

Figure 8: After Student Presses "We're thinking" Button, and a 2 Minute Wait, the DM Asks the Same Question Again. Note the Change in the Lab Transcript on the Lower Left.

If the students click on a button whose label text ends with an ellipsis, such as the "We're trying to measure…" button, we present them with another template, which is similar to a form one would use on a web page, to provide their answer. This avoids the problem of trying to understand open-ended text and gives the students information regarding what type of response they can give.



What are you trying to do?

We're trying to measure blood pressure and dialysate press

Mentor:What are you trying to do?
You: We're thinking
Mentor:What are you trying to do?

Figure 9: After Student Presses "We're trying to measure…" Button and Begins Filling Out the Form.

As Creanimate did, we use Button Theory[16] by providing a certain set of buttons that are available to the students at almost any time. We have already included some buttons (see lower right corner of figures 6-8) that allow the system to let students control the mentoring session (for instance our "Help me control and vary parameters." button). In future, we also plan to include some buttons for students to express an emotional state (like a "Huh? I'm confused" button) or allow students to quickly ask standard questions (like "How do I do that?"). This will aid our efforts to allow a student group to have a conversation with our system and not only signal choices or communicate the state of the experiment. Button Theory will help our system collect signs of the students' confusion or incomplete understanding, signals human mentors perceive often through facial expressions, body language, or tone of voice. It also gives students opportunities to ask certain questions that we know the system can address. This will make our system's interaction with students more like the student group interactions with Dr. Walsh that are highly valued.

Again, it should be noted that the HCI and NLP techniques we utilize are limited to pre-defined learning tasks where the possible set of utterances is well known. Our system would not be useful for open-ended tasks because the dialog paths and phrases used would not be predictable.

Validation

Usability Testing

A small usability test was performed with a biomedical engineering undergraduate student. The student had done the lab under Dr. Walsh's tutelage. Her interactions were typical of those in the actual lab, and were used as a seedbed for the initial dialog graph. Several months later, we videotaped her interacting with the first version of the Dialysis Mentor. This test revealed a number of phrases that needed to be added, as well as some new dialog branches. It also showed that it was very easy for particular DM responses to lead the student to treat the system like a multiple choice guessing game. This was particularly true when the DM's immediate follow-up to a student response strongly implied that their response was wrong. To avoid this unfortunate phenomenon, some follow-ups were changed to avoid immediate indications of any judgment of correctness. This made the dialogs more Socratic because it gave students more chances to discover their conceptual errors on their own.

Pilot Testing

The Dialysis Mentor was used in Dr. Walsh's Dialysis Lab in October 2001. Fifty-two biomedical engineering undergraduates were involved in two sections, a morning section and an afternoon section. Each section had nine groups of two to three students each. The DM logged all interactions for later analysis. Some interactions were video taped. A day shy of 2 weeks after the use of DM in the lab, 47 students completed a survey regarding their experience. Although Dr. Walsh and the rest of the team in general consider the pilot test a success, preliminary analysis of the data revealed or emphasized issues for future research. In particular, Table 1 categorizes the DM's responses by whether the intended follow-ups to student responses were taken. "Missing phrase in task dialog graph document" means that the DM had a relevant line of dialog available, but did not have a rule to link an unexpected student response to that dialog. This was often due to the students misspelling a word or using a synonym we did not include in the task dialog graph's rules. "Wait too long" means that the DM told students that it would wait for them to do something, but waited an improperly long period of time (usually a 5 minute wait for an activity that only required under a minute to perform). "No existing branch" means that the DM's dialog graph did not have a relevant line of dialog for the student's response. "Appropriate action" meant that the DM selected the dialog that we had intended for the given response.

Mentor Action Category	% of Mentor Actions Across Groups	Count Across Groups	Mean Count per Group
Missing phrase in task dialog graph document	2.82%	25	2.27
Wait too long	3.83%	34	1.89
No existing branch	22.41%	199	10.47
Appropriate action	70.38%	625	27.17
TOTAL		**888**	**49.33**

Table 2: Dialysis Mentor's Categorized Actions with Students in Pilot Test

Again, in the cases where student responses did not match any rules they were asked to rephrase their answer and had the option of choosing a different response or starting at the beginning of a dialog stage.

The following chart from the surveyed opinions of 47 of the 52 students who were in the pilot test shows that the vast majority felt that the DM was at least somewhat helpful to them. Given the rough preliminary nature of the DM, this is quite encouraging.

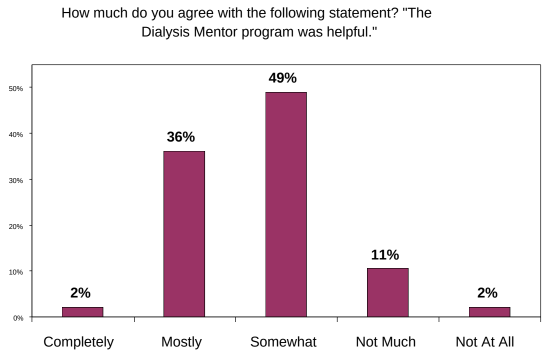


Chart 1: Student Opinions on the Helpfulness of the Dialysis Mentor

From the opinions student wrote in their surveys, they generally seemed most annoyed by the wait periods that were too long, which is consistent with our belief that students tend to want to rush through the lab.

The following chart shows the results of the conceptual assessment question in the same survey. Answers to the question “What parameters should one vary when quantifying the hydraulic permeability and what parameters should one hold constant?” was scored by first noting who answered correctly to what parameters should be varied and then who answered correctly to what should be held constant. 9 students (19.15%) were incorrect about what to vary and 12 students (25.5%) were incorrect about what to control. 6 students (12.77%) were found incorrect about both parts of the question and 9 students (19.15%) were partially correct. 32 students (68.09 %) answered the question completely correct.

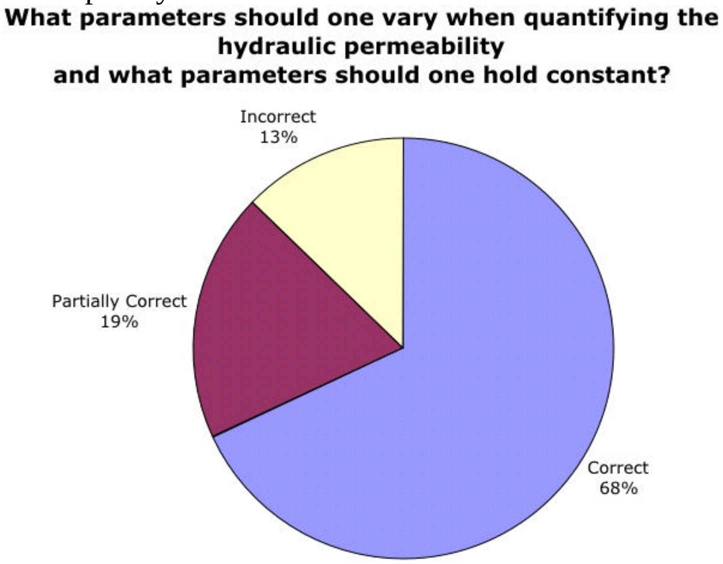


Chart 2: Student Responses to Conceptual Assessment Question

Future Research Issues

Student-Computer Interaction and Interpretation Issues

Videotapes of the pilot test shows cases where a student group was at a loss for an answer to a question presented by the DM and then received help from Dr. Walsh. In these cases Dr. Walsh asked the students to pretend he had just asked them the same question, and the students usually quickly came up with an answer. When they entered that answer into the DM, it usually handled it properly and the tutoring session progressed. In other words, students treated questions from the DM differently from the same questions from Dr. Walsh.

There are several possible explanations for this phenomenon. One is the difference between text and speech. Questions that are clear when spoken often become ambiguous in text form due to the loss of vocal cues. In addition, without video clips, the DM lacked the ability to give facial expression cues or body language based messages. Another factor is the difference in social relationships between a computer program and a faculty member. When their professor, Dr. Walsh, says "Think about it. I'll ask you again in 5 minutes," it seems that the students tended to stop and think. When the DM program said the same thing, students sometimes did what they could to get around the delay, including restarting the program. Since getting students to stop and think is a main goal of a SASK system, overcoming this difference in response strategies is a critical research problem.

We want students to attend to the questions proposed by the Dialysis Mentor with similar weight as they would if Dr. Walsh was standing in front of them and asking the same questions. To that important end we hope to collect video clips of Dr. Walsh actually saying all the utterances the Dialysis Mentor might say. These clips could then be played in conjunction with the textual presentation. Perhaps even the other parts of the graphical interface should be temporarily hidden from view while a clip of Dr. Walsh is played to help give the students the feeling that they are talking to him. Including such clips had always been a long-term goal for the project. Studies done by Moreno, Mayer, Spires and Lester show that the addition of just the audible speech may improve DM as a pedagogical agent and improve students' ability to transfer what DM teaches them to other situations[17]. It may also increase students' retention and feeling like they are talking to Dr. Walsh, or someone with a similar social status of a professor, therefore reducing the differences in their response patterns. Reeves and Nass have shown that interaction with computers can evoke social responses from people[18]. During our next in-lab test of an improved DM, we plan to use a control group along with pretests and posttests. These measures will help us ascertain the effect of DM on conceptual learning and transferability in a manner similar to studies done by Moreno, Mayer, Spires and Lester in their constructivist work with animated pedagogical agents[17].

We also need to make sure students do not casually guess at answers or assume their untested hypotheses are true. These situations may occur when students believe they can tell if their response choice was correct or not, based on the system's first follow-up. That leads students to try each response and see what happens, trying to use DM as a way to quickly find answers without having to analyze and test the system. Therefore, we need to minimize two kinds of initial follow-ups: negative "no, because…" follow-ups, and "I don't understand, please rephrase" follow-ups. The latter happened approximately 25% of the time in the DM, either because of missing phrases (3%) or missing branches (22%). Fortunately, avoiding bad follow-ups is mostly a matter of expanding and refining the Dialog Graph Document. This refinement process may be facilitated by an easy to use SASK Mentor authoring tool (discussed later).

One of the research issues we face regards what manner the interaction between the software and students will generally be initiated, by the software or by the human tutor. Our original intention was for the system to occasionally ask students a general question and the students should respond to the query, just as they would to a question from Dr. Walsh. Two other options were also before us. Dr. Walsh could talk to a student group enough to discover their problem, and then have the student ask the system a certain question in order to get assistance with their particular issue. Alternatively, we could give Dr. Walsh an interface to allow him to put the software in a mode to assist students with specific types of problems. For the fall 2001 lab we decided to have every group use DM when they were deemed ready to start with the main goal stage by Dr. Walsh or a graduate assistant. Next year we may run experiments designed to determine which interaction initiation scheme is best.

Another interesting issue is how we will structure the dialog. For instance, the Main Goal Stage begins with the Dialysis Mentor question:

- What is the overall or general goal of this experiment?

The goal of this main question is to remind students what the general point of the experiment is or help them discover it. Originally the proposed student responses were:

- Measure ultrafiltration (U)
- Measure transmembrane pressure (TMP)
- Graph U over TMP
- Find the ultrafiltration factor
- Measure the hydraulic permeability

However upon review, Dr. Walsh wanted to avoid directly stating the final goal (find the ultrafiltration factor) and proposed expanding the list to be:

- Measure the hydraulic permeability
- Graph U vs. TMP
- Measure ultrafiltration
- measure transmembrane pressure
- quantify the effect of blood flow on ultrafiltration
- quantify the effect of dialysate flow on ultrafiltration
- quantify the relationship between ultrafiltration and dialysate pressure
- quantify the relationship between ultrafiltration and blood pressure
- measure the transport properties of the polymer material that comprises the dialyzer
- determine the accuracy of the rotameters
- quantify the effects of osmotic agents on dialyzer transport

He argued this list would make our system more Socratic by offering more choices that students have made in his lab. Dr. Walsh however thought most choices would not lead to long dialogs before the students would discover an error in their thinking and return to the main question. We found that frequently providing that many choices would cause the dialog graph to be too big to code before the lab occurred in the fall quarter of 2001. We made the pragmatic decision to limit most responses to about five choices, particularly in the Main Goal Stage. Two or three of those responses are ones that students typically choose and seem plausible, but in fact lead away from the solution. Thus, we will still challenge students to think through their problem while we assist them in finding the solution. However, reducing the number of responses a student can give may make the program less Socratic. We plan to add more choices and we might change some of the software's architecture before next fall's lab. We need to ascertain how many student response choices are enough for the software to be sufficiently Socratic. This number will be different for each question, so we might provide a text-input area for students to type responses they wanted to ask, but could not because the provided graphical interface did not allow it.

Pedagogical Templates for Teachers' Authoring Tool(s)

Another exciting area of future design research will be the authoring tool. So far, we have been typing XML code into the task dialog graph document with a text editor. This is a tedious and technically complicated activity. Our plan is to speed up and improve this process by designing an authoring tool for use by expert teachers such as Dr. Walsh to build the task dialog graph document. Such a system will in some way be in competition to programs like Macromedia's Authorware and Click2Learn's Toolbook. Of particular interest will be the templates we will create, based on Collins' Socratic rules and other useful dialog patterns we have discovered while designing the first SASK Dialysis Mentor. Such templates would give teachers a faster way to produce pedagogically sound dialogs. A teacher could, for instance apply Collins' Socratic Rule 15: "Request a test of the hypothesis about a factor" to part of a mentoring dialog about photosynthesis, provided they are at least an expert in that domain.

An easy to use authoring tool is key to giving teachers the ability to make good SASK Mentors without working with a professional knowledge engineer, thus saving them time and possibly money.

It should be noted that the SASK architecture does not keep teachers from building SASK Mentors that have more than one solution or ending state. We expect SASK Mentors to be useful in many domains, including in the social sciences and humanities.

Internet Accessibility

We have a preliminary HTML interface to the SASK Mentor (that uses a Java Servlet) to ease deployment to many sites. We are also working on a web-based interface for the authoring tool.

Conclusions

Overall, our SASK Dialysis Mentor has the potential to be an effective tool for improving the biomedical engineering laboratory learning experience by fostering students' reflection on their tasks. The SASK architecture is limited to well-defined tasks where the possible dialog paths and phrases are well known. However, the biggest remaining challenges are designing such systems so they are treated as social and pedagogical agents and not as elaborate multiple choice guessing systems, as well as designing an authoring tool to help teachers easily make quality

SASK Mentors on their own. We look forward to improving the SASK architecture and making other SASK Mentors for other learning environments and domains.

Acknowledgements

We thank Dr. Brian Reiser, Dr. Daniel Edelson, and Dr. Allan Collins for their suggest ions and groundwork.

This work was supported in part by the Engineering Research Centers Program of the National Science Foundation under Award Number EEC-9876363.

Bibliography

[1] SASK Mentor Project: http://www.cs.northwestern.edu/~riesbeck/sask/
[2] Collins, A. (1977). Processes in acquiring knowledge. In R. C. Anderson, R. J. Spiro, & W. E. Montague (Eds.), Schooling and the acquisition of knowledge. Hillsdale, NJ: Lawrence Erlbaum.
[3] Collins, A. & Stevens, A. L. (1991). A Cognitive Theory of Inquiry Teaching. In Peter Goodyear (Ed.), Teaching Knowledge and Intelligent Tutoring. Norwood, NJ: Alex Publishing Corporation.
[4] Anderson, John R.. (1988). The Expert Module. In Martha C. Polsen & J. Jeffrey Richardson (Eds.) Foundations of Intelligent Tutoring Systems. Hillsdale, NJ: Lawrence Erlbaum.
[5] Bransford, Brown, Cocking (Eds.) (2000) How people learn; brain, mind, experience, and school. Washington, D.C. : National Academy Press.
[6] Shulman, L. (1986). Those Who Understand: Knowledge growth in teaching. Educational Researcher 15(February), 4-14.
[7] Graesser, Arthur C., Xiangen Hu, Suresh Susarla, Derek Harter, Natalie Person, Max Louwerse, Brent Olde, and the Tutoring Research Group (2001) AutoTutor: An Intelligent Tutor and Conversational Tutoring Scaffold. http://www-2.cs.cmu.edu/~aleven/AIED2001WS/Graesser.pdf 10th International Conference on Artificial Intelligence in Education, San Antonio, Texas.
[8] Clancey, William J. (1982). Tutoring rules for guiding a case method dialogue. In D. Sleeman & J. S. Brown (Eds.) Intelligent Tutoring Systems. New York, NY: Academic Press.
[9] Clancey, William J. (1987). Knowledge-Based Tutoring: The GUIDON Program. Cambridge, Massachusetts: The MIT Press.
[10] Wenger, Etienne. (1987). Artificial Intelligence and Tutoring Systems. Los Altos, CA: Morgan Kaufmann.
[11] Rosé, Carolyn Penstein, Johanna D. Moore, Kurt VanLehn, and David Albritton. (2001) A Comparative Evaluation of Socratic versus Didactic Tutoring. In 23rd Annual Conference of the Cognitive Science Society. Edinburgh, Scotland, August.
[12] VanLehn, Kurt. (1988). Student Modeling. In Martha C. Polsen & J. Jeffrey Richardson (Eds.). Foundations of Intelligent Tutoring Systems. Hillsdale, NJ: Lawrence Erlbaum.
[13] Littman, David & Soloway, Elliot. (1988). Evaluating ITSs: The Cognitive Science Perspective. In Martha C. Polsen & J. Jeffrey Richardson (Eds.). Foundations of Intelligent Tutoring Systems. Hillsdale, NJ: Lawrence Erlbaum.
[14] Cleary, Chip. & Schank, Roger. (1995). Engines for Education. Hillsdale, NJ: Lawrence Erlbaum.
[15] Edelson, D. C. (1993) Learning from stories: Indexing, reminding, and questioning in a case-based teaching system. Ph.D. dissertation, Northwestern University.
[16] Jona, M., Bell, and L. Birnbaum. 1991. "Button theory: A taxonomy of student-teacher communication for interface design in computer-based learning environments." In Proceedings of the Thirteenth Annual Conference of the Cognitive Science Society in Chicago, IL, 765-769.
[17] Moreno, Roxana and Richard E. Mayer, Hiller A. Spires, James C. Lester. (2001). "The Case for Social Agency in Computer-Based Teaching: Do Students Learn More Deeply When They Interact with Animated Pedagogical Agents?" In Cognition and Instruction, 19, no. 2, 177-213. Hillsdale, NJ: Lawrence Erlbaum.

[18] Reeves, B., & Nass, C. (1996). The Media Equation. New York: Cambridge University Press.

Biography

BABA KOFI WEUSIJANA is a Ph.D. student in the Learning Sciences Program at Northwestern University. He received his M.S. in Engineering from San Jose State University in 1999. His interests include educational software and distance learning. He is dedicated towards improving learning environments around the world through design research and the application of learning science and high technologies.

DR. CHRISTOPHER K. RIESBECK is an Associate Professor in the Department of Computer Science at Northwestern University. He received his Ph.D. in Computer Science from Stanford University in 1974. He is a Fellow of the American Association for Artificial Intelligence. His current research focuses on tools for mentored distance learning and intelligent learning environments.