The Casper Project:
Integrating Simulation, Case Presentation, and Socratic Tutoring to Teach Diagnostic Problem-Solving in Complex Domains

Alex Kass

Contributors:

Laura Allender
Noreen Burke
Scott Dooley
Will Fitzgerald
Wayne Schneider

The Institute for the Learning Sciences
Northwestern University
1890 Maple Avenue, Evanston, IL 60201
Phone: 708 491-3500
E-mail: kass@ils.nwu.edu

May 1994

The Institute for the Learning Sciences was established in 1989 with the support of Andersen Consulting, part of The Arthur Andersen Worldwide Organization. The Institute receives additional support from Ameritech and Northwest Water, Institute Partners.

1. Introduction

In this paper we will describe an architecture for computer-based learning environments that we have created to teach students to develop hypotheses about complex systems.* The architecture is instantiated in software as a domain-independent shell, and a set of authoring tools. The shell and the tools facilitate the creation of education and training applications which combine text, graphics, video, and sound to produce a class of systems that allow a student to learn by practicing a skill on the computer under the "watchful eye" of a tutor. The tutor can intervene when necessary to provide guidance, often in the form of real-world human experts captured on video. They answer a student's questions, explain why things might be going wrong, and also offer their own first-person stories to illustrate the underlying principles that the student must learn. The real-world videos are shown within interactive, context specific, Socratic-style dialogs through which the tutor helps students examine their causal reasoning and discover and fix gaps in their reasoning. We believe that the realistic practice environment, the real-world stories told at appropriate moments, and the Socratic tutoring compliment each other in a synergistic way to create a learning environment that is both engaging and effective.

1.1. Intelligent Learning-by-Doing Environments

The architecture discussed in this paper was developed as part of a larger effort to develop tools that facilitate the development of a broad range of systems that we call *Intelligent Learning-by-Doing Environments* (ILDEs). ILDEs have two main layers: the task-environment layer and the tutoring layer. In addition, most ILDEs also contain some type of hypermedia-based reference materials, which comprise an important third component. However, the interesting issues involved in building that component (see, for example [Ferguson, et al., 1992]) are beyond the scope of this paper.

We have developed a range of ILDE's to teach a variety of subjects. The tasks we have covered range from the fairly simple, such as operating a cash register, to the very complex, such as battlefield intelligence analysis. All the systems contain both a task environment and a tutoring layer, although the form these components take varies in interesting ways according to the needs of the students, and the nature of the subject matter. For instance, the *Broadcast News* program (Kass, et al., 1994a) is an ILDE designed to teach high school students about

* The system described in this paper represents the efforts of a large group of people, including: Laura Allender, Noreen Burke, Mark Chung, Scott Dooley, Dave Faloon, Ric Feifer, Kemi Jona, Frank Luksa, Faina Mostovoy, Wayne Schneider, and Howard Stamp. In addition, the underlying toolset owes a good deal to the ideas of Enio Ohmaye.

political science and current affairs by enabling them to put together their own television news show. The task environment in Broadcast News asks the student to modify the text and video in a rough draft of a story presented to them, using a structured multimedia tool. The tutoring layer responds to student actions in the task environment by presenting video clips of political scientists, journalists, and other experts who challenge the student's editorial decisions and explain the issues behind the story.

Another example of an ILDE is the *YELLO* program (Kass, et al., 1994b), designed to teach a student how to sell Yellow Pages advertising. The training that YELLO provides revolves around a set of engagements with simulated customers, with differing goals, plans and personalities. The student's task is to pick up on the (often subtle) cues that the simulated customers give about themselves, and to explain (in a style that meshes with the customer's personality) how the Yellow Pages can serve their needs. While the task-environment layer of YELLO is, clearly, very different from that in Broadcast News, the behavior of the tutoring layer is fairly similar in structure. The system presents clips of expert salespeople telling real-world stories that relate to the student's current situation.

The focus of this paper will be on our attempt to develop an ILDE that can teach a student to diagnose a complex system and develop hypotheses about the causes of a set of observed phenomena. These are core skills in many fields, including medicine and other scientific and technical areas. The particular prototype we have developed, called *Casper*, is designed to teach trainee customer service representatives for a British water utility to diagnose problems that customers report over the phone. This task involves interacting with customers, as with YELLO, but the personality of the customer is not as important to the task of diagnosing their water problems as it is when selling. Therefore, the simulation of the customer (which is at the core of both task environments) can be considerably simpler in Casper than in YELLO. On the other hand, because diagnosing a problem with the water system often requires more complex causal reasoning as well as considerable technical expertise, the tutoring component of Casper is more complex than those in Broadcast News and YELLO. In addition to presenting experts telling instructive stories, the Casper tutor engages the student in more interactive, Socratic-style dialogues to help the student re-examine his or her reasoning and discover his or her own misconceptions.

1.2. Overview of the paper

In this paper we first describe the theory of learning, and the philosophy of education that motivates the ILDE effort, and influences the specific approach taken in developing the Casper architecture. Then, we describe the water-system diagnosis application we have built. This discussion will cover the interface elements used to facilitate communication between the student and the simulated customer, and between the student and the tutor. We will also discuss the kind of tutoring required to teach this type of diagnosis, and the mechanisms Casper uses to implement that tutoring. Regarding the authoring tools we have developed to support the Casper architecture, this paper will make only passing reference, not nearly enough detail to satisfy prospective authors. That detail will be forthcoming shortly in another paper.

2. Theoretical underpinnings

The Casper architecture, and the ILDE effort generally, is based on two aspects of our theory of how people think and learn: (1) people reason best from specific cases rather than from abstract rules, and (2) people learn best by doing (particularly by making mistakes, discovering those mistakes, and then correcting them), not by rote memorization or by passively watching an expert perform. At a somewhat more detailed level, the educational philosophy underlying Casper can be summarized by the following four points[1]:

1. In order to learn any complex skill, students must be able to engage in realistic exercise of those skills, as opposed to engaging in drill-and-practice regimens in which small pieces of the target skills are learned out of context. This claim is consistent with research such as that reported in (Lave and Wenger, 1991).
2. Practice alone is usually not enough. As discussed in (Collins, Brown and Newman, 1989), students need guidance to make their interaction with a learning environment an effective learning experience. Furthermore, to best exploit the natural synergy between practice and instruction it is important that the two be tightly interleaved.
3. When teaching about complex systems, for which students must rely on rules of thumb, it is crucial to expose students to the real-world experiences of experts, not just abstract rules. Through real-world experiences, students can see how heuristics are applied in context (Burke, 1993, Kass, et al., 1994).
4. The most effective way to teach complex causal reasoning tasks is through an interactive, mixed-initiative dialogue (Carbonell, 1970), in which students are helped to discover and fix their own misconceptions.

2.1. Practice, instruction, and learning from failure

In formal schooling, knowledge and skills are often abstracted away from their uses in the real world. In contrast with traditional apprenticeships, where learning is grounded by hands-on experience, classroom teaching often fails to make contact with the practices it is meant to inform. As a result, motivation is often low, and much of what is learned is quickly forgotten, or remains

[1]See, (Kass, et al., 1994) for a lengthier discussion of points 1-3.

"unintegrated or inert" (Collins, Brown & Newman 1989). Conversely, some other methods of teaching bypass the abstract principles altogether. Instead, the student is taught the practice by rote. Such training produces students who have memorized the standard steps to take, but who may be unable to adapt when faced with novel situations.

Real-world human learning often depends on being able try something out, to fail, and to explain one's failures (Schank, 1982). In order to experience this, students need environments in which practice and instruction are tightly integrated. The instruction can then help students figure out where they are going wrong, and can help them fix their own misconceptions.

While most progressive researchers agree with the idea of learning-by-doing, some (perhaps most notably Papert [1980]) argue against explicit instruction. They argue against explicit because it restricts the student's freedom They maintain that if the task environment is rich enough it will be sufficient to entice students to explore and learn on their own. However, an open-ended environment without guidance can be frustrating. Explicit guidance is particularly critical when teaching complex problem solving, where long causal chains must be learned, because without such guidance, it can be very difficult for students to determine which part of their causal chain is faulty. Students performing a complex task may not know enough to explain their failures on their own, even when those failures are pointed out to them.

When instruction and problem-solving are combined appropriately, a powerful synergy can be achieved. The student can reflect on his or her experiences in the learning environment with the guidance of an instructor, and the abstract principles described by the instructor are motivated, operationalized, and made memorable by the problem-solving activity (Schank & Jona, 1991; Williams, 1993).

As we discuss in (Kass, et al., 1992), it is pragmatic constraints that typically keep the synergy between practice and instruction from being realized in a traditional classroom. It is rarely possible for a teacher to be available to deliver appropriate instruction at the moment the student requires it. Furthermore, because many activities cannot be performed in the classroom, there is often a physical separation between the location of instruction and that of practice. Thus in the most common modern training regimens, instruction and realistic problem-solving are divorced from one another in time and place. A principle advantage of computer-based instruction is the opportunity it affords to overcome the practical obstacles that often force a separation between problem-solving and instruction.

2.2. Using computers to integrate practice and instruction

Computer simulations can make the integration of instruction and problem-solving feasible where it otherwise would be quite difficult to achieve. (Kass, et al., 1994). A computer-based solution has several potential advantages. First, computer simulations can make it possible for students to practice tasks that might otherwise be too expensive, and/or too dangerous. In a simulated world students are safe to take risks and learn from their mistakes. Taking an example from the Casper domain, diagnosing the problems that customers experience with their tap water can have significant health and safety implications. It would be dangerous to allow students to learn by practicing on real customers. However, making a potentially harmful mistake with a simulated customer allows the student to be exposed to the negative consequences of the mistake without actually endangering anyone.

In addition, a nice feature of computer simulation is that it is possible for such a simulation to be suspended long enough to deliver instruction at any point in the practice activity. In the real world it is not always possible to slow tasks down or suspend them. It would not, for instance, be a good idea for an instructor to stop a customer service representative in the middle of real customer interaction in order to deliver instruction, but with simulated customers this is quite feasible. The instructor can even allow the student to "take back" something that has been said to the customer.

Finally, and perhaps most important, a computer can provide individualized instruction that is tightly integrated with the student's practice activity. Although individualized instruction by a human might be the ideal, teachers are not typically available to watch everything a student does during practice sessions and to make instruction available exactly when needed. Computers can do this.

2.3. Teaching Complex Problem Solving

Complex problem-solving is an issue that has only recently received significant attention in psychology (Sternberg & Frensch, 1992). In the sense used, for example by Funke (1992), the most important criteria for defining complex problems is that such problems are not transparent; the important variables are not immediately observable. This criterion applies to diagnostic problem solving as it occurs in real world situations. When customer service representatives try to resolve customer problems over the phone, for example, they often have multiple goals active at one time, and those goals sometimes conflict. When a CSR at a water utility interviews a customer to diagnose a water problem, he or she is trying to gather as much information in as little time as possible, and at the same time is trying to avoid upsetting the customer, and attempting to avoid unnecessary service calls. Particular actions that the student performs are often intended to serve more than one of these goals at once, but other times one of the goals must be partially sacrificed in order to serve another. Deciding which goal should have priority at any given time requires subtle judgments that cannot easily be reduced to hard-and-fast rules. It is the type of skill that requires practice and experience at least as much as learning specific rules.

Social issues and other real-world complications can make a straightforward model of diagnosis task more difficult to apply in practice. For example, consider the following simple model of diagnosis, which sounds reasonable, *a priori:* First, gather enough information to identify what class of problem a customer has. Second, consider each of the possible causes of these symptoms and generate questions based on these symptoms which might eliminate one or more of these causes. By limiting the number of likely causes we have also reduced the number of possible treatments. When there is only one reasonable treatment, the diagnosis is sufficient.

The complicating factors that arise in the real-world application of this model make the task more challenging than the simple model implies. For example, the most efficient way to rule out a certain cause may not be the best way to eliminate the cause. When attempting to solve a customer's problem, asking if he feels ill could cause him to be unnecessarily alarmed, negating the potential benefits of eliminating a set of possible causes. Casper is designed to teach both a comprehensive understanding of a complex system *and* procedural diagnostic problem-solving skills. It does so in an integrated environment where a tutor watches for both tactical errors (such as asking alarming questions) and domain-knowledge errors (such as failing to realize that a particular piece of evidence rules out a certain cause).

Most *intelligent tutoring systems* (ITSs) have been able to ignore such complicating factors because they were designed to teach introductory-level lessons to students who are almost complete novices. They rely heavily on an ability to model the student's knowledge, and the interventions they provide are driven primarily by a comparison between their expert model and their student model (Anderson, 1988; Wenger, 1987). However, students learning complex tasks cannot be productively modeled as novices. Even when students are just beginning to learn a new skill they do not come to the task as blank slates. Maintaining an accurate model of the student's knowledge is very difficult if the amount of (correct and incorrect) knowledge that the student has is large. The task is made even more difficult when much of the student's knowledge is acquired from sources other than the tutoring system. Complex skills are refined from knowledge students already have. At least as important as learning new concepts and new skills is learning how to fit those new concepts and skills into one's existing repertoire. This sense of refining is often quite absent for formal instruction in areas such as mathematics and science where the emphasis is to try to work around students' incorrect naive notions (Driver, Guesne & Tiberghien, 1985). It is much more like teaching advanced writing (Bereiter & Scardamalia, 1987): students' existing abilities have to be deepened and broadened. The tutoring approach of integrating Socratic-style dialogs and expert commentary employed by Casper achieves the goal of most ITSs, which is to teach domain knowledge, but it also helps the students make subtle adjustments to their problem-solving strategies.

When the student's model of the task is assumed to be large rather than small, and is assumed to be primarily derived from previous life experience rather than from interactions that the tutor can monitor, it is impractical to rely on a model of tutoring that requires a precise model of the student's mental state. Casper's tutor has a rather detailed *domain* model, but it does not attempt to build up a detailed student model. It doesn't pretend to know what the student is thinking at any given point. Instead, it attempts to judge the student's actions based on the negative effects they would have in the simulated world. Rather than attempt to deduce what the actions might indicate about the student's thought processes, the tutor responds to the student's inappropriate actions (*or* to the student's explicit request for help) with interactive dialogs that allow the students to explicitly communicate their reasoning. Then the tutor can critique the student's reasoning and help the student to identify their own misconceptions.

3. Casper: training students to diagnose problems in the water system

Casper was developed to address an important industrial training problem: To train employees of a British water company how to diagnose problems with the water system that customers report over the telephone. First we will describe what the system does in general terms; then we will give an overview of the main components of the system, including a description of how graphics, video and sound are used to present those components to the student; finally, we will give a step-by-step description of a sample session a student could have with the system.

3.1. What A CSR must learn to do

The customer service representatives (CSR) Casper is designed to train are responsible for handling all the telephone calls that customers make to the company. In addition to complaints about the water system, these calls can include general interest inquiries, billing inquiries, and even threats. However, our focus in the initial version of the application has been limited to training

CSRs to handle complaints and inquiries about water quality. This is probably the most challenging type of inquiry that CSRs at a water company handle.

When a customer calls the water company to complain about water quality, the conversation typically begins with the customer describing a problem that he or she has noticed with the water coming out of the tap. The description is often vague and incomplete, and may even include mistaken information. The CSR must ask the right questions, diagnose the cause of the problem, if possible, and prescribe the appropriate remedy, which may be something the customer can do, such as running the water, or may involve sending a water company worker to investigate further or make repairs. Among the factors that make the job complicated is that the cause of the problem may be difficult or impossible for the customer to observe directly, so the CSR must deduce the cause of the problem by combining indirect, often probabilistic evidence.

An on-line "customer contact system" (CCS) helps the CSR do their job. In addition to logging problems and issuing work orders, the CCS can be used to look up information that may be relevant to the diagnosis of a customer's problem. For example, the CSR can use the information in the CCS to determine the type of source from which the customer's water is drawn (river, lake, or well); whether others in the customer's neighborhood have been reporting similar problems; and whether any work on the water system has been scheduled in the customer's neighborhood.

In order to do their job well, the CSR should have a detailed understanding of how the water system works – from sources, though treatment, storage, and distribution, all the way to the tap. The CSR must also understand the mechanics of how one develops a hypothesis about the unseen causes of an observed phenomenon – for example, how does one search for evidence, notice clues, and choose between alternative hypotheses when a customer calls to complain about discolored water.

3.2. The Casper simulation

The two entities that a CSR must interact with to handle a real call are the customer (via the phone) and the computerized CCS job-aid. The practice environment that Casper provides includes simulations of both of these entities.

The Simulated Customer Contact System: Figure 1 depicts the main Casper screen. The lower right portion of the screen contains the simulated CCS. The Casper CCS windows are a fairly close replica of the windows that CSRs use when actually doing their job. The student can enter information about the customer, just as they do with the real system. In response, the simulated CCS is capable of providing the same sort of information about the customer's history, neighborhood, and so forth that the real system provides in the "Impacted

Domestic Customer Profile" window. As the student gathers information about the customer's problem, he or she can categorize the problem in the "Record Contact" window. On the basis of this categorization, the CCS can offer a limited amount of prompting about topics that the CSR should explore with the customer.

When the student enters information in the simulated CCS, the Casper tutor monitors that activity, and uses the information entered by the student to help the student correct the mistakes in his or her own reasoning. If the student makes certain sorts of mistakes, such as incorrectly categorizing the customer's problem, the tutor will note this, and will later use that information, when the student asks for help or when the mistake in categorizing the problem leads to a larger mistake, such as a misdiagnosis. The form of intervention provided by the tutor will be discussed in Section 3.2.3.

Communication from the simulated customer to the student: The simulated customer is the most important part of the simulation. Casper uses pre-recorded sound clips to allow the student to hear what the customer says in response to the student's questions. The audio clips help make the simulation feel more realistic. They also allow the student to get a feel for the *tone* of the customer's voice as well as the words. This allows Casper to use the customer's tone of voice to convey important factors, such as a customer's anxiety level.

In addition to playing an audio clip for each utterance made by the simulated customer, Casper provides a running transcript of the ongoing conversation, which is depicted in the lower left of Figures 1-6. (As depicted in Figure 6, the student can scroll back in the transcript to review the earlier portions of a long interaction.) Of course, because this transcript can help the student notice clues in the customer's answers that were missed when first uttered, in some ways it provides an unrealistic crutch. It is valid to worry that students will not feel the need to listen to a simulated customer as carefully as they should if they know that they can rely on the transcript. However, since the focus of the system is on teaching theory-building skills rather than audio-perception skills, the trade-off is worthwhile.

Communication from the student to the customer: It would probably be ideal if the student could literally speak into a microphone to the simulated customer, and the system could employ voice-recognition and natural-language processing to interpret that student's utterances. However, because that technology is not yet robust enough to rely on, Casper provides a different interface to allow the student to indicate what he or she wants to say to the customer.

To be more precise, there are two interfaces from which the student can choose. The first of these is a hierarchical set of menus that the student can use to construct an utterance. In the top left corner of Figure 1 are the five top-level choices the student has when constructing an utterance. The student's choices are as follows:

- The student can ask the customer about something.

  Example utterance: "What does your water smell like?"
- The student can explain something to the customer.

  Example utterance: "Your water is not safe to drink."
- The student can give the customer instructions.

  Example utterance: "Please run the cold tap for a bit and tell me what you see."
- The student can promise the customer some sort of action.

  Example utterance: "I'll send a system controller out right away."
- The student can end the conversation by saying good-bye.

Once the student decides which top-level type of utterance he or she wishes to make, a second-level menu appears to allow the student to refine the utterance. For example, Figure 2 depicts the menu that enables the student to choose a topic about which to ask the customer. There are approximately twenty topics that the student can ask about, and after the student chooses a topic, a third menu sometimes appears to allow the student to further refine the utterance. For instance, if the student chooses to ask about a leak, the sub-topic menu will appear to allow the student to choose between asking about the duration of the leak, or the location of the leak, or the rate of leakage. Sometimes as many as four menu choices are required to refine the utterance completely, although more typically only two or three are needed. When the student has fully refined the utterance using the menus, the English sentence corresponding to the student's chosen utterance appears in the box next to the "SAY THIS" button. If the utterance is satisfactory to the student, he can push the "SAY THIS" button to simulate saying it to the simulated customer. The student will then hear a response from the simulated customer (unless the tutor intervenes).

The *utterance constructor*, as we call this set of hierarchical menus, has some pedagogical advantages, but also some disadvantages. The main advantage is that explicitly presenting the choices helps reinforce in the student's mind the sort of choices that are valid. There are several disadvantages. First, because invalid choices (for example, leading questions) must also be available to the student, these may be reinforcing to the student—or worse, the student may feel tricked into making an invalid choice by its presence on the menu. Second, students might, in the tutorial environment, rely on a tool they wouldn't have in a real life environment to suggest a correct choice. Third, when the space of things to say is large enough, it can be time-consuming to find just what one intends to say.

Between the two poles of voice recognition on the one hand and hierarchical menus on the other, lies another sort of interface: a type-in box in which students can type what they want to say, and have the system map this to its own representations of things a student might say. A type-in box with an effective parser behind it overcomes, to some extent, the problems associated with hierarchical menus. Students neither feel tricked into saying the wrong thing, nor do they have appropriate answers unrealistically suggested to them; what the student types, we assume, is what the student wants to say.

As an alternative to using the utterance constructor, the student can type a sentence directly into the box next to the "SAY THIS" button. Casper then uses a specialized natural language processing technology called *indexed concept parsing* (Fitzgerald, 1994a) to decide which of the utterances it knows about are likely matches to what the student has typed. After the student presses the Return key, the system replaces the text typed by the student with the utterance in Casper's repertoire which the system considers to be the best match. If the student finds that the text provided by the system is a close-enough match, he or she can hit the "SAY THIS" button, just as if the utterance had been constructed with the menus. If the utterance is not a sufficiently-close match, then the student can try using the "Other Choices" menu (which appears on the upper right of the screen shown in Figure 1). This menu contains the other close matches that the system found. If none of these is a good match to the student's intended utterance, he or she can fall back on the utterance constructor menus or try rephrasing the sentence. In a pilot test using water-company trainees as subjects we found that many users preferred the type-in box to the utterance constructor and were able to say what they intended more quickly (Fitzgerald 1994b).

The job of the Casper parser treats parsing as a recognition task; its job is to use the text that the student types to select the most appropriate utterances from its database of the utterances that customers can respond to. This can be tricky, because while there are a fixed number of utterances that the simulated customers can respond to, there are many different ways to express each of those utterances. For example, in a pilot study, trainees who served as subjects asked the question, "What kind of bits are in your water?" eight different ways:

- Can you describe the particles?
- What colour are the bits?
- Describe the bits.
- Please describe the bits.
- What are the bits like?
- What do the bits look like?
- Could you describe what the bits look like?

- Could you describe the bits?

Indexed concept parsing works by tagging each of the utterances in its database with sets of indices, called *indexed concept pools*. Each of the pools associated with an utterance contains indices that would be activated by a particular way of expressing the utterance. When the student enters something in the type-in box, the parser scans for indices referred to in the student's text, and then returns the best matches. For example, the indexed concept pools associated with "What kind of bits are in your water?" are {`water-bits description`}, {`water-bits like`}, {`water-bits description size`}, and {`water-bits description colour`}. Each index has pattern strings associated with it; examples of pattern for the previous indices include associating "bits" and "particles" with `water-bits`; "similar" and "like" with `like` and "describe," "description," "what kind" and "what type" with `description`. If the student enters "Can you describe the particles," the parser finds references to `water-bits` (from "particles") and `description` (from "describe"). It uses a fairly straightforward algorithm (described in Fitzgerald 1994a) to determine that "What kind of bits are in your water?" is a better match than, for example, "Are there any bits in your water?" Because indices are arranged in a semantic hierarchy, the parser can properly interpret specific references as instances of general patterns. For example, it will recognize that "Are the bits in both the hot and cold taps?" is a way of expressing "Is the problem in both the hot and cold taps?" because `water-bits` is a type of `water-quality-problem`.

3.3. Hypermedia reference material on the water system

In addition to the simulation described above, and the tutor, described below, Casper provides the student with a hypermedia map of the water system that serves as a very useful reference tool. The *water map* is available to the student at any time to help if the student needs to get information to handle a simulated call. In addition, as we shall see in Section 3.3, the tutor explicitly refers to this tool when helping the student.

The water map can be brought up in the same space occupied by CCS, as depicted in Figures 2-6. When it first appears, it shows a high-level picture of the "trip to the tap" that water takes, from sources, through treatment, storage, distribution, and into the customer's property. Each of the stages along the way is a clickable region in the water map. If the student clicks on one of the regions, a brief description of that stage is shown to the student. In addition, a "more detail" button appears. The student can get a detailed look at the selected stage of the water-delivery system by clicking on that button.

Figure 3 shows what a detailed view of the distribution stage looks like. Every event that can take place in the distribution stage and can cause problems with the customer's water is depicted. Thirteen problems are shown for the distribution stage, including: work being done on the main, a leak in the main, and a worker disinfecting the water in the main. A similar number of problems are depicted for each of the other three stages of the water delivery system.

If the student clicks on one of the possible problems depicted on the water map, a box appears with text describing the observable symptoms the problem is likely to cause. If the student wishes to find out more about a problem, he or she can choose to ask for a more detailed description, which can be either text or video. For example, Figure 3 shows a video of a water-company engineer describing what can be caused by work on the mains. The student can watch as much of the video as is helpful. Students can thus suspend the simulation at any time to browse the water map, and get useful descriptions of any causes that they think might be relevant to a customer's problem. After the student views a clip describing one of the problems, as many as four follow-up questions become available to allow the student to explore other related issues, such as possible treatments for the problem, or how likely it is to occur.

Each of the clickable items in the water map is surrounded by a border. When the student indicates the likelihood of a cause, the color of the border around that cause changes. This serves as a reminder to the student the likelihood values that have been assigned. (The meanings of the colors can be seen in the key, visible at the top of the water map in Figure 4). When the student clicks on an item in the map, the system presents a menu from which the student can set the likelihood value of that item. (The menu can be seen in Figure 3, below the video). At the beginning of each simulated call, the system resets the likelihood values of all items to the neutral, "maybe" value. The student can set the value of an item to *likely*, *maybe*, or *unlikely*. If the student marks an entire stage on the overview level (the one depicted in Figure 2) as *unlikely*, this will set the values of all the items within that stage to *unlikely*. However, if the student changes any of the lower-level items to something other than *unlikely*, then the stage that the item is contained in will be reset to *maybe*.

To indicate the likelihood of potential causes, the learner must reason about the causes and investigate appropriately. The visual feedback given after a likelihood indication enables the learner to quickly see which potential causes may require further investigation. The tutor actively encourages the student to use the water map to record his or estimates of likelihood values on the water map. Doing so helps the student reason through the problem, and it also informs the tutor about the student's reasoning.

3.4. The Casper tutor

In this section we describe what sorts of behavior the Casper tutor exhibits, and then we describe the underlying technology that allows it to produce that behavior.

3.4.1. What the tutor does

The tutor in the Casper system can be invoked in two different ways. If the student is confused about why something happened or what to do next, he or she can explicitly invoke the tutor. The student asks the tutor "WHY?" or "NOW WHAT?" by hitting the button with that label (as seen above the transcript in Figure 1). In addition to responding to those explicit invocations by the student, the tutor will automatically intervene in response to certain actions. For example, the tutor will usually activate itself to challenge the student when the student announces a hypothesis to the customer which is not supported by the evidence thus far collected. It may also initiate a dialog with the student when he or she indicates an incorrect, or unsupported understanding of the situation, for instance through interactions with the CCS or the water map. The precise algorithm used by the tutor to decide when to invoke itself, and what to do once it is invoked is beyond the scope of this paper; we will just discuss illustrative examples here. The algorithm for determining when to tutor and what strategy to pursue is determined by the system designer through the use of a set of general purpose tutor-authoring tools partially described in (Jona & Kass, 1993).

The Casper tutor does not merely tell the student the correct answer to a question, but instead tries to lead the student through an appropriate chain of reasoning. The goal of the tutoring is not to reveal the solution to the simulated customer's problem, but to teach the student how to solve problems like it. For example, when the student clicks on "NOW WHAT?" the tutor responds as depicted in Figure 4; it says:

> To help you decide what to do next, we need to understand your current goal.
>
> Choose the item at the right that best describes your current goal.

The buttons at the right cover the range of activities that are appropriate when attempting to form a diagnosis and fix a problem. They are as follows:

- Gather Information
- Examine Possible Causes
- Narrow Down the Likely Causes
- Act On a Diagnosis

In addition to these buttons there is a final choice, intended for students who really don't understand the theory-building process:

• I Don't Have a Clue

If the student clicks this last button, the system presents a detailed explanation of what the other buttons mean, and describes the general sequence one should go through in determining how to solve a customer's problem. If the student chooses one of the other buttons, the tutor asks the student to attempt the next step that will help fulfill the chosen goal. If the student has chosen an inappropriate goal, the student will either discover this while attempting to meet the tutor's request, or will be informed by the tutor upon asking for assistance. For example, a student who indicates that he wants to act on a diagnosis is first asked to indicate his diagnosis using the water map, and is then led to a more appropriate goal if it is too early in the diagnosis process to settle on a hypothesis. When the student's goal is appropriate, the tutor reacts supportively; it will often present a video of a water company expert explaining how to do the sort of thing that the student has indicated he or she is trying to do.

When the tutor invokes itself to respond to a mistake the student has made, it does not simply announce what the student has done wrong and what should be done instead. Rather, it asks the student to explain his or her own reasoning, and it critiques that reasoning. For example, if the student announces an unsupported hypothesis about the cause of the customer's problem, the tutor will ask the student to defend the hypothesis. The student communicates the reasoning behind the hypothesis by selecting, from the transcript, specific utterances made by the customer which the student believes to be evidence for the hypothesis. (This is depicted in Figure 5.) The tutor then uses its expert domain model to critique the student's reasoning. For instance, the tutor might indicate that the items selected by the student give some evidence for the student's diagnosis, but that more likely causes exist. At this point the student may choose to receive a more detailed analysis of the evidence. The tutor might then ask the student to explore the water map to find other possible causes, and would ask the student to defend the new, alternative hypothesis in the same way as the old.

If the student makes a mistake, such as asking the customer to do something that is expensive or would inconvenience the customer without good cause, or is asking leading questions, the tutor will often break in with a video clip of a company employee explaining how the learner's previous action was inappropriate. When one is available, the tutor presents a video of an experienced CSR telling a real-world story about a time when he or she made a mistake similar to the one the student is currently making. (This is depicted in Figure 6.) By recounting the negative consequences of a mistake, just at the time when the student is making that mistake, the expert helps drive the lesson home in a very effective way. After the tutor offers negative feedback on a statement the student has made, the tutor allows the student to retract that statement before going returning to the customer.

3.4.2. How the tutor works

To support tutoring, Casper has two main kinds of knowledge. First, the tutor has access to domain-independent strategies for deciding when to tutor, and how to manage the teaching interactions. In order to implement those strategies, the tutor accesses the second kind of knowledge, which is knowledge of the specific domain. In the CSR trainer, this domain knowledge includes the causal chains that relate symptoms at the tap to root causes in the water system. Each symptom can be linked to various causes at one of several levels of certainty and each potential cause can predict the existence of several symptoms, also at one of several levels of certainty. For instance, the domain model encodes the fact that orange-colored water is usually a symptom of rust in the water, which is in turn caused by something stirring up rust in the mains, and the possible causes of that include a burst in the main, work on the main, or a fire truck drawing water from a hydrant.

Casper includes a set of authoring tools that can be used to develop the domain-independent strategies and the specific domain models. Applying the tutor to a new domain requires the use of the tools to author a new domain model, and perhaps to adapt the general strategies somewhat. No programming is required to do this.

Casper's strategies for intervention are contained in a list of rules that stipulate when a teaching interaction should take place, and which specific dialog with the student should result. An example of Tutoring strategy in Casper is:

IF the student makes a diagnosis of the problem
AND there is not enough evidence for the student's hypothesis
THEN execute the following tutoring sequence

1. Ask the student to justify his or her diagnosis.
2. Explain the insufficiency of the student's justification and why the diagnosis is premature.
3. Ask the student to retract the diagnosis statement.
4. Help the student with the next problem-solving step.

After using its domain model to determine that the preconditions for this strategy have been met, the tutor executes the strategy through the use of a series of rules and templates that allow the task-specific details to be spliced into a general interaction. For example, step 1 in the sequence above might be presented in a particular context as follows:

```
What evidence leads you to believe that Ms.
Hughes' milky coloured water has been caused
by work on the service pipe?
```

This query is generated from a general purpose template:

```
What evidence leads you to believe that
<agent-possessive-name> <agent-problem-
description> has been caused by <current-
student-hypothesis>?
```

Some of the fillers needed to instantiate some of the templates are a function of specifics of a call (for example, the name of the customer, or the specific hypothesis that a student has announced). Other fillers are drawn from the domain model. Still others are drawn from another source of tutoring knowledge, which is the system's case-base of video clips. For example, step 2 above is implemented using the following template:

```
show story for symptoms resulting from <current-
student-hypothesis>
```

which would be instantiated as:

```
show a story about the symptoms resulting from work
on the service pipe
```

When applying this strategy, the system would use the index 'symptoms resulting from work on the service pipe' to retrieve a video. If the system has a video stored under that index it will show it. If not, it would try a slightly different variation on the strategy.

3.5. Casper in action

Sections 3.1 and 3.2 described the main features of Casper. The goal of this section is to put the pieces together, and to make the description more concrete by describing a sample Casper scenario.

Each scenario takes the student through an entire phone call from a customer. The scenario begins with the system playing the sound of a phone ringing, and the tutor directing the student to begin by clicking the "Answer the Phone" button. The transcript below is an example of what might happen after the student answers the phone.

```
Student:   Hello, North West Water.  May I help you?
Customer:  Hello, this is Mr. Lamb of 44 Worthy Road in
           Liverpool.  My water is a funny colour, and
           I'm not sure about drinking it.
Student:   Is it in both the hot and the cold taps?
Customer:  I don't know.  I've only tried the hot tap.
```

Student: Can you run the cold tap for a bit and tell me what you see?

Customer: All right, I'll try it... both taps are affected.

At this point a typical student uses the CCS to correctly categorize the customer's problem. Our sample student might correctly enter the following information:

Contact category: Water Quality
Contact reason: Discoloured Water

Then the student can click the PROMPTS AND INFORMATION button on the CCS. The CCS suggests a number of actions relevant to various discolouration problems. One issue it suggests raising is: "Establish how long the problem has existed."

Student: How long have you had the problem?

Customer: My water was fine when I left for work this morning.

At this point the demonstration student is stuck. He pushes the "NOW WHAT?" button.

Tutor: To help you decide what to do next, I need to understand your current goal.

Click the button that best matches your current goal.

Gather Information

Examine Possible Causes

Narrow Down the Likely Causes

Act On a Diagnosis

I Don't Have a Clue

The student clicks on Gather Information.

Tutor: That sounds like a good idea. Here is the kind of information an Operations Officer suggests you collect.

(The tutor plays a video of an engineer explaining how to gather information about discoloured water problems. Among other things, the engineer suggest that one should ask how the water looks, tastes, and smells, and whether the neighbors have the same problem.)

The student then returns to talking with the customer:

```
Student:   Please describe the color of your water.
Customer:  My water is a sort of funny brown color, with
           a kind of sandy sediment in it.
Student:   Are any of your neighbours affected?
Customer:  Yes, I rang Mr. Wellman in the next house
           over.  His water is just the same.
```

The student proceeds like this for a while, gathering facts about the case. We will not take up space by giving all the question and answer pairs. We jump to a point where the student announces a frightening unsupported hypothesis.

```
Student:   Your water is contaminated.

Tutor:     Contamination is a very serious problem.
           Telling the customer that his water is
           contaminated may cause him to panic.

           Watch this video about a CSR who caused a
           customer to panic.
```

The tutor then plays a video about a CSR who once caused a customer to panic simply by *asking* the customer whether there was anything swimming in her water. The customer immediately assumed that the question implied that there *were* things swimming in the water. After getting off the phone with the CSR, the customer called the health department and told them that the water company told her that there were eggs in the water which would develop into swimming insects.

```
Tutor:     So you should be very sure not to panic the
           customer, especially if you are not sure what
           the problem is.

           Why do you think Mr. Lamb's water is
           contaminated?
```

The student clicks on some items in the transcript: The customer's indication that his neighbors are effected and that the problem has only existed since this afternoon. After giving specific explanations of how those items each give a small amount of evidence for the contamination theory, the tutor then summarizes as follows.

Tutor: Those examples give weak evidence toward the possibility of contamination being the cause of Mr. Lamb's problem. In addition, contamination seldom occurs. You should first consider more likely theories.

Would you like to retract your statement to the customer about the water being contaminated or leave the statement be?

The example student chooses to retract his statement. "Your water is contaminated." is taken out of the transcript and the simulation will continue as if it was never said.

Tutor: There are more likely causes for Mr. Lamb's problem.

Use the water map to indicate which potential causes you think are likely.

The student then browses the water map looking at some of the descriptions, ruling things out, and marking a few as likely. Eventually, the student decides to pursue "mains burst" as a likely cause.

Tutor: What evidence do you have that a mains burst might be the cause of Mr. Lamb's problem?

The student clicks on items in the transcript: The customer's indication that the problem has been going on a short time and that there are black bits in the water. Again, after allowing the student to see specific explanations of the evidence items, the tutor gives the following summary of the evidence.

Tutor: Those items give some evidence toward a mains burst as the cause of Mr. Lamb's problem.

You need more information to decide if that is the real cause of the problem.

The tutor then gives the student the choice of further investigating burst on the main or choosing another likely cause. Our example student chooses to investigate a burst on the main.

```
Tutor:      Here is what a Quality Compliance Engineer
            says about a burst on the main.
```

The tutor plays a video of an engineer describing the results of bursts in the main. Our sample student then returns to the customer.

```
Student:    Have you seen any works in your neighbourhood?
Customer:   Yes, the fire brigade were here earlier,
            checking the hydrants.
Student:    The fire brigade must have flushed the main,
            stirring up sediment.
Customer:   What should I do about it?
Student:    Try running the water from the cold tap for a
            quarter hour.
Customer:   Okay. But what if it doesn't clear?
Student:    Call me back if the problem continues.
Customer:   Okay. I'll do that.  Goodbye.
Student:    Thank you for calling North West Water.
```

4. Conclusion

In this paper we have attempted to show how multimedia can be used in an effective way to teach diagnosis and hypothesis-formation skills. We have described a system that combines audio, video, graphics, and text to produce an intelligent learning-by-doing environment in which a student can practice a diagnostic task, and can receive feedback from a Socratic-style tutor and from real-world experts – captured on video who respond to the actions the student takes in the simulation.

The system we have described is currently being pilot tested with employees of the water company that sponsored its development. The system they are testing consists of six scenarios. Each scenario presents the student with a very different sort of problem. Each of the six simulated customers in the current version of Casper has a few hundred things that he or she can say, depending on what the student does. Students vary in the amount of time they take to complete a scenario, but the average seems to be 30 minutes each, so the six scenarios provide about three hours of training. As described earlier, the tutor has an extensive model of the water system that allows it to give feedback on a wide range of mistakes. The system currently contains approximately two hours of video of experts (approximately 80 clips, usually between 45 seconds and 90 seconds in length).

While we are quite proud of the Casper application, it is the general learning-environment architecture we have designed, and the tools we have built to support future implementations of that architecture which we believe represent the truly important accomplishments of the Casper project. Our primary objective is now to improve the authoring tools we developed in service of the Casper system so that other, similar systems can be built rapidly, and the pedagogical architecture that Casper represents can be applied to teach many different subjects. We are currently using initial versions of the tools. These include a tool that is used to create the simulated entities for the task environment, and two tools that are used to create the tutor layer. One of the tutor tools facilitates the creation of domain-independent tutoring strategies, which can be reused from application to application. The other allows the creation of the specific domain model for a particular application. The tutor in a Casper-like application consists of the domain-independent strategies running on the specific domain model.

These tools were used by non-programmers to produce the current version of Casper. Creating a new Casper scenario can now be accomplished without any additional programming. The tools are also being used to produce other applications that are similar to Casper. We feel this is an exciting accomplishment. However, the tools are not yet as powerful or as easy to use as we would like. Although the tool users do not need to know how to program, they do need considerable training in the use of the tool. It currently takes one of our tool users a week or two to produce a new scenario in a domain that we've already investigated. This is probably an order of magnitude better than programming from scratch, but it is still not as fast we would like. Furthermore, we do not yet have a tool for producing user interfaces for programs like Casper. Building the user interface still requires programming. Our goal is to produce a tool that can be used to create an entire Casper-like application without any programming, and can be used by the general public to create a new scenario in a few days without extensive training. By making such a toolset available we hope to make authoring tools a vehicle for changing the way that education and training is performed.

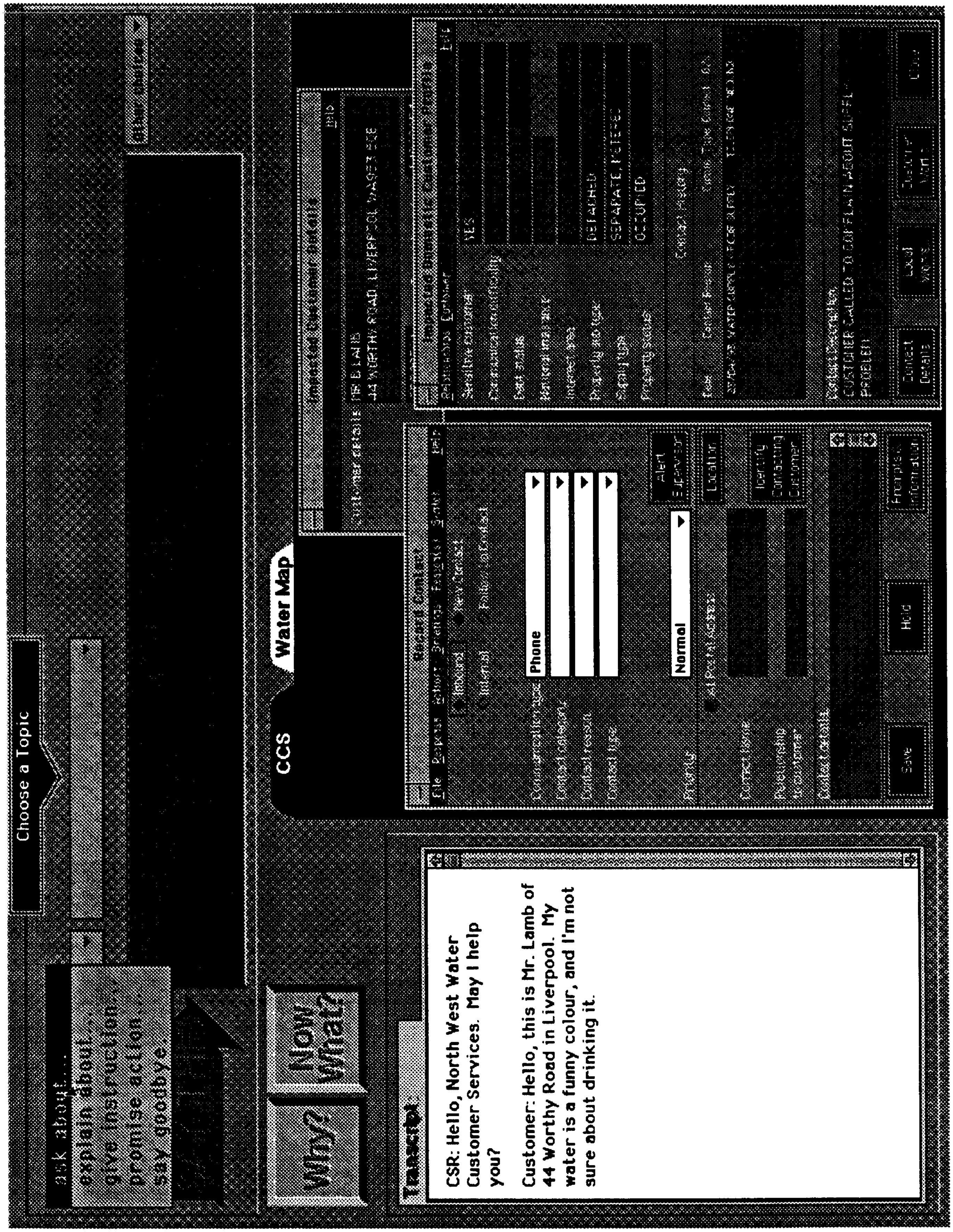
Choose a Topic
ask about...
explain about...
give instruction...
promise action...
say goodbye.
Why?
Now What?
Transcript
CSR: Hello, North West Water Customer Services. May I help you?
Customer: Hello, this is Mr. Lamb of 44 Worthy Road in Liverpool. My water is a funny colour, and I'm not sure about drinking it.
CCS
Water Map
Phone
Normal
Save
Hold

Figure 1

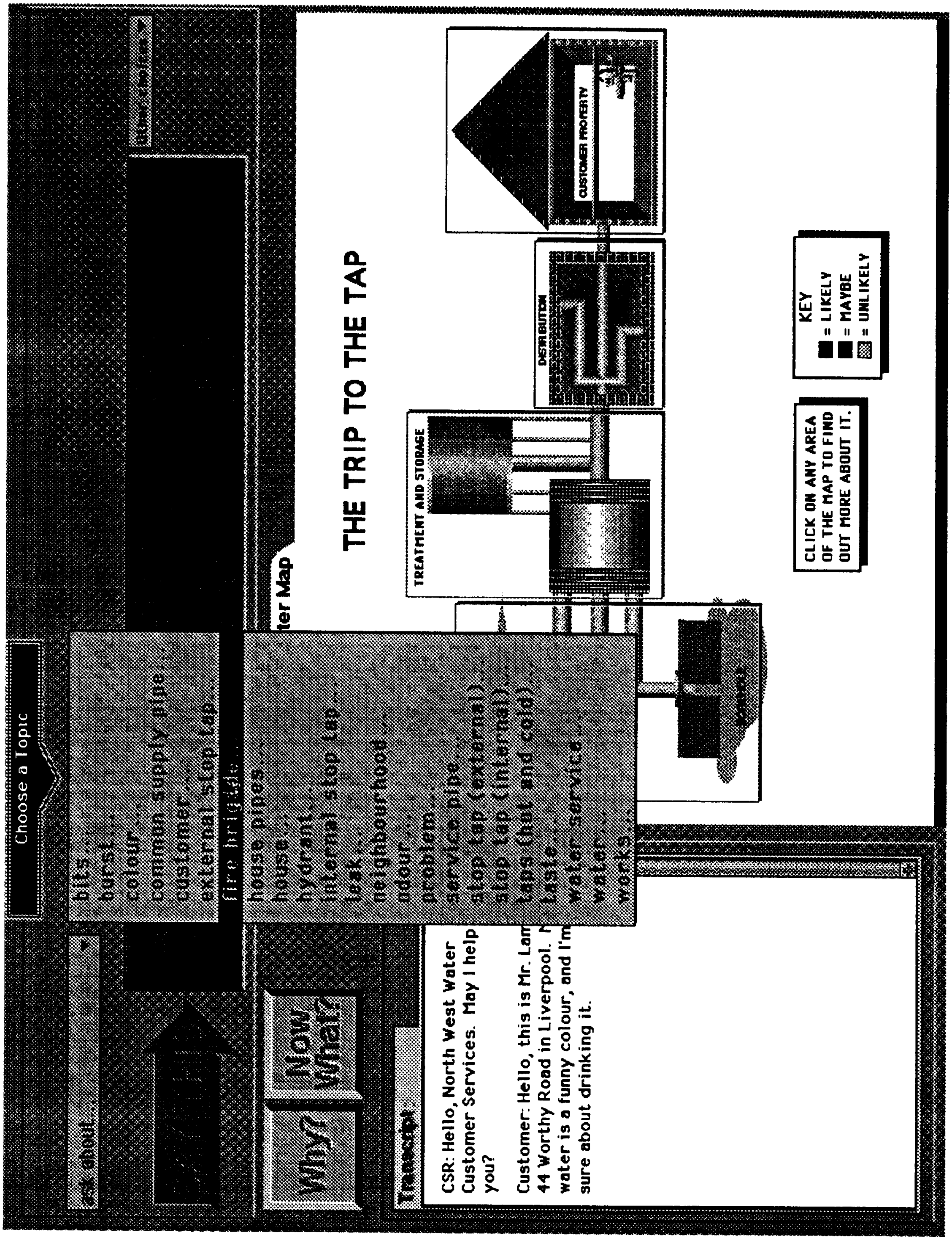
THE TRIP TO THE TAP
TREATMENT AND STORAGE
DISTRIBUTION
CUSTOMER PROPERTY
CLICK ON ANY AREA OF THE MAP TO FIND OUT MORE ABOUT IT.
KEY
= LIKELY
= MAYBE
= UNLIKELY
Choose a Topic
bits...
burst...
colour...
common supply pipe...
customer...
external stop tap
fire brigade...
house pipes...
house...
hydrant...
internal stop tap...
leak...
neighbourhood...
odour...
problem...
service pipe...
stop tap (external)...
stop tap (internal)...
taps (hot and cold)...
taste...
water service...
water...
works...
ask about...
Why?
Now What?
Transcript
CSR: Hello, North West Water Customer Services. May I help you?
Customer: Hello, this is Mr. Lam
44 Worthy Road in Liverpool.
water is a funny colour, and I'm
sure about drinking it.

Figure 2

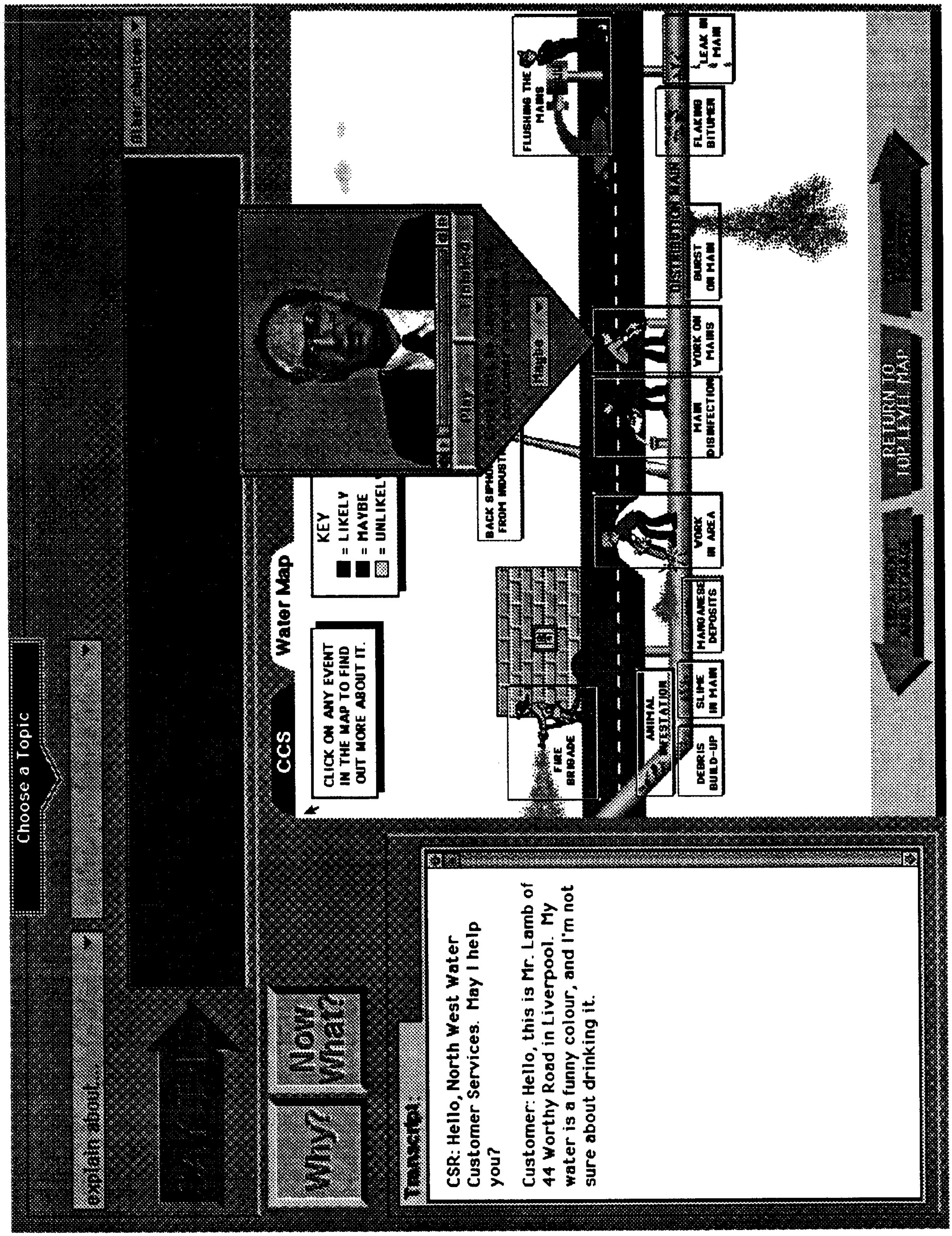
Choose a Topic
explain about...
Why?
Now What?
Transcript
CSR: Hello, North West Water Customer Services. May I help you?
Customer: Hello, this is Mr. Lamb of 44 Worthy Road in Liverpool. My water is a funny colour, and I'm not sure about drinking it.
CCS
Water Map
CLICK ON ANY EVENT IN THE MAP TO FIND OUT MORE ABOUT IT.
KEY
= LIKELY
= MAYBE
= UNLIKEL
FIRE BRIGADE
ANIMAL INFESTATION
DEBRIS BUILD-UP
SLIME IN MAIN
MANGANESE DEPOSITS
VORK IN AREA
MAIN DISINFECTION
VORK ON MAINS
BURST ON MAIN
FLUSHING THE MAINS
FLAKING BITUMEN
LEAK IN MAIN
RETURN TO TOP LEVEL MAP

Figure 3

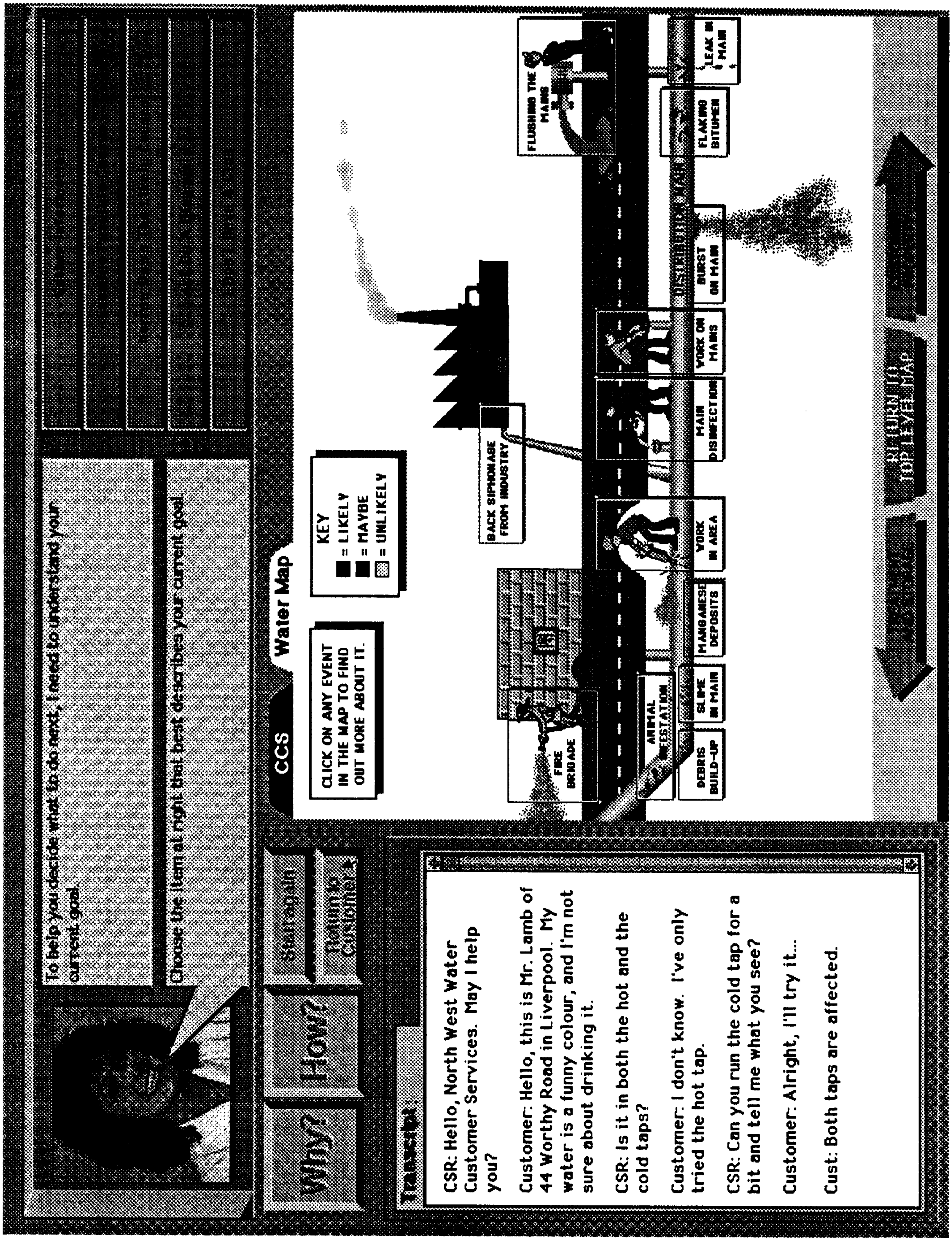
To help you decide what to do next, I need to understand your current goal.
Choose the item at right that best describes your current goal.
Why?
How?
Return to Customer
Transcript
CSR: Hello, North West Water Customer Services. May I help you?
Customer: Hello, this is Mr. Lamb of 44 Worthy Road in Liverpool. My water is a funny colour, and I'm not sure about drinking it.
CSR: Is it in both the hot and the cold taps?
Customer: I don't know. I've only tried the hot tap.
CSR: Can you run the cold tap for a bit and tell me what you see?
Customer: Alright, I'll try it...
Cust: Both taps are affected.
CCS
Water Map
CLICK ON ANY EVENT IN THE MAP TO FIND OUT MORE ABOUT IT.
KEY
= LIKELY
= MAYBE
= UNLIKELY
BACK SIPHONAGE FROM INDUSTRY
FIRE BRIGADE
ANIMAL INFESTATION
WORK IN AREA
MAIN DISINFECTION
WORK ON MAINS
FLUSHING THE MAINS
DEBRIS BUILD-UP
SLIME IN MAIN
MANGANESE DEPOSITS
BURST ON MAIN
FLAKING BITUMEN
LEAK IN MAIN
DISTRIBUTION
RETURN TO TOP LEVEL MAP

Figure 4

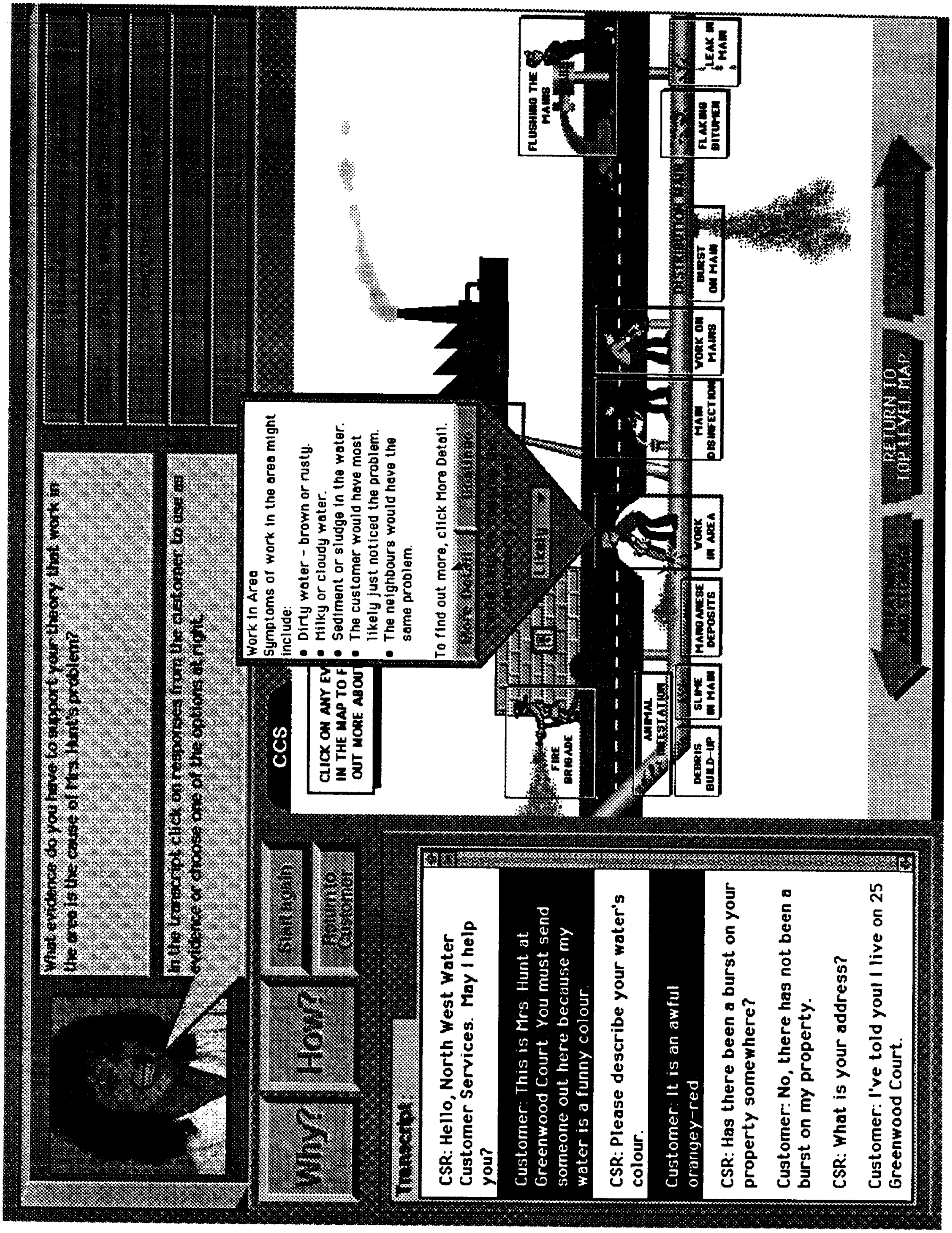
What evidence do you have to support your theory that work in the area is the cause of Mrs. Hunt's problem?
In the transcript click on responses from the customer to use as evidence or choose one of the options at right.
Why?
How?
Start again
Return to Customer
Transcript
CSR: Hello, North West Water Customer Services. May I help you?
Customer: This is Mrs Hunt at Greenwood Court. You must send someone out here because my water is a funny colour.
CSR: Please describe your water's colour.
Customer: It is an awful orangey-red
CSR: Has there been a burst on your property somewhere?
Customer: No, there has not been a burst on my property.
CSR: What is your address?
Customer: I've told you! I live on 25 Greenwood Court.
CCS
Work in Area
Symptoms of work in the area might include:
• Dirty water - brown or rusty.
• Milky or cloudy water.
• Sediment or sludge in the water.
• The customer would have most likely just noticed the problem.
• The neighbours would have the same problem.
To find out more, click More Detail.
More Detail
FIRE BRIGADE
DEBRIS BUILD-UP
ANIMAL INFESTATION
SLIME IN MAIN
MANGANESE DEPOSITS
WORK IN AREA
MAIN DISINFECTION
WORK ON MAINS
BURST ON MAIN
FLAKING BITUMEN
LEAK IN MAIN
FLUSHING THE MAINS
RETURN TO TOP LEVEL MAP

Figure 5

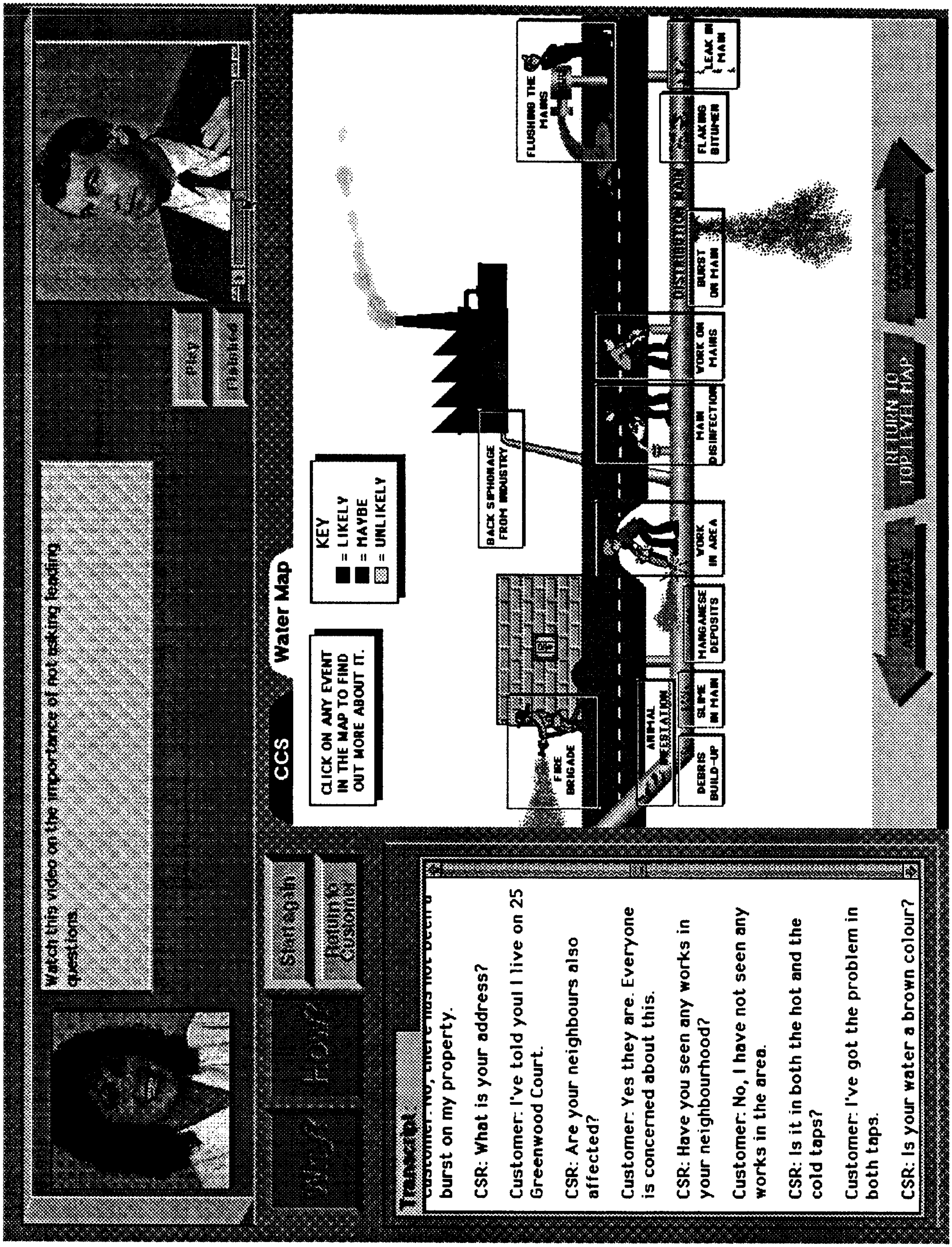
Watch this video on the importance of not asking leading questions.
Play
Finished
CCS
Water Map
CLICK ON ANY EVENT IN THE MAP TO FIND OUT MORE ABOUT IT.
KEY
= LIKELY
= MAYBE
= UNLIKELY
FIRE BRIGADE
BACK SIPHONAGE FROM INDUSTRY
FLUSHING THE MAINS
LEAK IN MAIN
FLAKING BITUMEN
BURST ON MAIN
WORK ON MAINS
MAIN DISINFECTION
WORK IN AREA
MANGANESE DEPOSITS
SLIME IN MAIN
DEBRIS BUILD-UP
RETURN TO TOP LEVEL MAP
Transcript
burst on my property.
CSR: What is your address?
Customer: I've told you! I live on 25 Greenwood Court.
CSR: Are your neighbours also affected?
Customer: Yes they are. Everyone is concerned about this.
CSR: Have you seen any works in your neighbourhood?
Customer: No, I have not seen any works in the area.
CSR: Is it in both the hot and the cold taps?
Customer: I've got the problem in both taps.
CSR: Is your water a brown colour?

Figure 6

References

Anderson, J.R. (1988). The Expert Module. In M. C. Polson & J. J. Richardson (Eds.), *Foundations of Intelligent Tutoring Systems.* Lawrence Erlbaum Associates, Hillsdale, NJ.

Burke, R.D. (1993). *Representation, Storage and Retrieval of Stories in a Social Simulation.* (Ph.D. Thesis). ILS Technical Report #50, Northwestern University.

Carbonell, J.R. (1970) *Mixed-Initiative Man-Computer Instructional Dialogues.* Doctoral dissertation, Massachusetts Institute of Technology, Cambridge, MA.

Collins, A., John S. Brown & Susan E. Newman (1989). "Cognitive Apprenticeship: Teaching the Crafts of Reading, Writing, and Mathematics." In L. B. Resnick, (ed.), *Knowing, learning and instruction: Essays in honor of Robert Glaser.* Lawrence Erlbaum Associates, Hillsdale, NJ.

Ferguson, W., Bareiss, R., Birnbaum, L., and Osgood, R. (1992) ASK Systems: An Approach to the Realization of Story-Based Teachers, *The Journal of the Learning Sciences,* 2(1), pp. 95-134. Lawrence Erlbaum Associates, Hillsdale, NJ.

Fitzgerald, W. (1994a) "Indexed Concept Parsing for Interactive Tutors." *Working Notes of the 1994 AAAI Workshop on Active Parsing.* Palo Alto, CA.

Fitzgerald, W. (1994b). "Evaluating parsers based on the fulfillment of practical goals." Submitted to the 1994 AAAI Workshop on Knowledge Representation for Natural Language Processing, New Orleans, LA.

Jona, K. and Alex Kass (1993) "The Teaching Executive: Facilitating Development Of Educational Software Through The Reuse Of Teaching Knowledge." Presented at the 10th Annual International Conference on Technology and Education. Cambridge, MA.

Kass, A., R.Burke, E. Blevis, M. Williamson (1992). The GuSS Project: Integrating Instruction and Practice Through Guided Social Simulation. ILS Technical Report #34. Northwestern University, Evanston, IL.

Kass, A., R. Burke, E. Blevis, M. Williamson (1994a). "Constructing Learning Environments for Complex Social Skills." To appear in *The Journal of the Learning Sciences,* 3(4). Lawrence Erlbaum Associates, Hillsdale, NJ.

Kass, A., S. Dooley, F. Luksa, and C. Conroy (1994b). "Using Broadcast Journalism to Motivate Hypermedia Exploration." To appear in *Educational Multimedia and Hypermedia Annual, 1994.* Association for the Advancement of Computing in Education, Charlottesville, VA.

Klahr, D., K. Dunbar, & A. Fay (1990). Designing good experiments to test bad hypotheses. In J. Shrager & P. Langley (eds.), *Computational models of scientific discovery* 355-402. Morgan Kaufmann, San Mateo, CA

Lave, Jean & Etienne Wenger (1991). *Situated Learning: Legitimate peripheral participation.* Cambridge University Press. New York, NY.

Papert, Seymour (1980). *Mindstorms: Children, Computers and Powerful Ideas.* Basic Books. New York, NY.

Schank, Roger C. & Menachem Y. Jona (1991). Empowering the Student: New Perspectives on the Design of Teaching Systems. *Journal of the Learning Sciences,* 1(1). New York, NY.

Schank, Roger C. (1982). *Dynamic Memory: A Theory of Learning in Computers and People.* Cambridge University Press. New York, NY.

Schauble, Leona, Robert Glaser, Kalyani Raghavan, & M. Reiner (1991). Causal models and experimentation strategies in scientific reasoning. *Journal of the Learning Sciences,* 1(2) 201-238. Lawrence Erlbaum Associates, Hillsdale, NJ.

Shrager, Jeff (1990). Commonsense perception and the psychology of theory formation. In J. Shrager & P. Langley (eds.), *Computational models of scientific discovery and theory formation* (pp. 437-470). Morgan Kaufmann. San Mateo, CA.

Funke, Joachim (1991). Solving Complex Problems: Exploration and Control of Complex Systems. In R. J. Sternberg and P. A. Frensch (eds.), *Complex Problem Solving: Principles and Mechanisms* (pp. 185-222). Lawrence Erlbaum Associates. Hillsdale, NJ.

Williams, Susan M. (1993). Putting Case-Based Instruction into Context: Examples from Legal, Business, and Medical Education. *Journal of the Learning Sciences..* Lawrence Erlbaum Associates, Hillsdale, NJ.