Implementing tutoring strategies into a patient simulator for clinical reasoning learning

Froduald Kabanza[a,*], Guy Bisson[b], Annabelle Charneau[a], Taek-Sueng Jang[a]

University of Sherbrooke, Sherbrooke, Québec, J1K2R1 Canada
[a]Department of Computer Science
[b]Faculty of Medicine

KEYWORDS
Intelligent tutoring systems;
Clinical reasoning learning;
Patient simulation.

Summary

Objective: This paper describes an approach for developing intelligent tutoring systems (ITS) for teaching clinical reasoning.

Materials and methods: Our approach to ITS for clinical reasoning uses a novel hybrid knowledge representation for the pedagogic model, combining finite state machines to model different phases in the diagnostic process, production rules to model triggering conditions for feedback in different phases, temporal logic to express triggering conditions based upon past states of the student's problem solving trace, and finite state machines to model feedback dialogues between the student and TeachMed. The expert model is represented by an influence diagram capturing the relationship between evidence and hypotheses related to a clinical case.

Results: This approach is implemented into TeachMed, a patient simulator we are developing to support clinical reasoning learning for a problem-based learning medical curriculum at our institution; we demonstrate some scenarios of tutoring feedback generated using this approach.

Conclusion: Each of the knowledge representation formalisms that we use has already been proven successful in different applications of artificial intelligence and software engineering, but their integration into a coherent pedagogic model as we propose is unique. The examples we discuss illustrate the effectiveness of this approach, making it promising for the development of complex ITS, not only for clinical reasoning learning, but potentially for other domains as well.

[*] Corresponding author at: Department of Computer Science, University of Sherbrooke, Sherbrooke Québec J1K 2R1, Canada. Tel.: +1 819 82 8000; fax: + 1 819 821 8200.
Email address: kabanza@usherbrooke.ca

1. Introduction

Clinical reasoning describes the sum of problem-solving and reasoning required to diagnose a patient's disease, and decide on a management plan. Clinical reasoning learning (CRL) and clinical problem analysis provide a method for teaching students clinical reasoning [1-4]. To successfully "solve" a clinical problem, students rely on declarative knowledge they have learned and also on similar cases they have seen in clinical settings or in previous CRL sessions. Competence in clinical reasoning requires exposure to many cases, with adequate supervision and feedback. Recent changes in the healthcare sector have reduced opportunities for students to learn clinical reasoning under optimal conditions: patients in most university medical centers have multiple complex medical problems, skilled clinical instructors are becoming more scarce, and a shift towards ambulatory care for patients has decreased the number of hospitalized patients.

All these considerations prompted us to develop a patient simulator, called TeachMed, that teaches medical students to diagnose a simulated patient by collecting evidence (taking the history, performing a physical exam and ordering labs), and formulating hypotheses about the probable disease or pathology based upon collected evidence.

The main innovative feature of the current version of TeachMed is its pedagogic module, that is, the module responsible for providing tutoring feedback to the student during the problem-solving process. Pedagogic knowledge is modeled using a hybrid finite state machine (FSM), in which states model the key phases of a problem-based learning process for clinical reasoning. The hybrid nature of the FSM is reflected by the production rules that are attached to states to detect situations necessitating pedagogic intervention and to activate a dialogue with the student to help him. The dialogue is in turn implemented by another kind of state machine.

To provide feedback, TeachMed relies on a model of an expert capable of solving the case, represented as an influence diagram (ID) [5]. Given the evidence collected by the student at the current stage and the hypothesis he is considering, the ID is used to determine what would be the current step from the point of view of the expert. Logic propositions about the student's current step and the expert's current step trigger preconditions of tutoring feedback rules to capture situations requiring tutoring intervention and to initiate a tutoring dialogue.

Situations requiring intervention are not just situations in which the student has made a mistake. Sometimes it is preferable to let students learn from their mistakes. It may also be desirable to provide additional information to the student, depending on his progress through a case. For example, after a student has sought some particular piece of evidence, obtained the result, and related the collected evidence to the right hypothesis, it may be beneficial to explain that a different test result would have yielded a different hypothesis. Hence, preconditions in tutoring feedback rules involve factors other than just tracking mistakes. They are part of an entire pedagogic strategy, which can be more or less complex, depending on the clinical case and the underlying teaching objective.

Some strategies require providing help based on past solving steps and previous tutoring interventions. We use modal temporal operators [6] in the preconditions of feedback rules to express such requirements about the past. In some cases, this is more elegant, more modular and more efficient than using preconditions based on an explicitly recorded trace of the student's solving steps.

As far as we know, such an integration of IDs, FSMs, production rules and temporal logic is unique. Each of these modeling formalisms has been proven to be independently useful in various subfields of artificial intelligence and software engineering, but individually. We demonstrate that it not only makes sense to combine them, but this also leads to a natural and efficient pedagogic model for intelligent tutoring systems (ITS) which teach complex cognitive tasks such as clinical reasoning. We believe the approach can be useful in other ITS domains as well. This paper expands in many ways the ideas that we first introduced in [17]. The different components of the pedagogic model are better articulated and they are extended with temporal logic specifications. More details are also provided on how the system works.

2. Related work

Commercial tools such as DxR [7] and VIPS [8] simulate patients on which students can practice by asking questions, performing a physical exam, or ordering labs. The difference is that these tools provide no immediate feedback. In fact, students have to wait until their teacher has looked at their solution traces, as is the case with traditional homework activities.

ITS attempt to implement teaching capabilities that are close to the skills of human teachers. Typically the tutoring system includes (1) a student model which maintains a representation of the student's behaviors, attitudes, subject mastery, and or levels of individual skills, (2) an expert model – encompassing the expert knowledge required to solve a case, and (3) a pedagogic model - which provides instruction that changes in response to the state of the student model [9].

TeachMed is an inquiry-based learning ITS, comparable to medical diagnosis systems such as RASHI [10] and Bioworld [11]. However, the expert models in these previous systems do not support uncertainty reasoning. GUIDON was one of the earliest ITS in medicine [12], aimed at the diagnosis of bacterial infections. Given a clinical case on which to teach, GUIDON used MYCIN (a production-rule based expert system) to generate a solution graph. GUIDON would then check the student's actions (e.g., questions asked) against the solution graph (i.e., intuitively against the steps the expert would have taken), to intervene whenever the student's actions were deemed suboptimal, or whenever the student asked for help. In contrast, TeachMed uses an expert model based on IDs rather than production rules; production rules in TeachMed are only involved in the pedagogic model and are much simpler. Another difference is with the pedagogic model in GUIDON, which helped a student towards any of the solutions found by MYCIN (and hence encoded in the solution graph), but made no attempt in guiding him towards optimal solutions. In contrast, TeachMed is able to infer, for example, the best next action for collecting evidence and advise the student accordingly.

SlideTutor [13-14] is an ITS for teaching visual classification problem-solving. Its expert model is based on production rules, but contrary to GUIDON, it uses a more abstract knowledge representation. The solution graph is generated dynamically, by using problem solving methods that instantiate the general expert domain knowledge on the fly, taking into account the current problem solving state and specific information related to the case. In contrast to TeachMed, SlideTutor does not emphasize the value of evidence, and its expert model does not support reasoning under uncertainty.

COMET [15] is an ITS with an expert model based on Bayesian networks. It thus accounts for reasoning under uncertainty, but it focuses more on hypothesis formulation than on evidence gathering. For instance it does not model cost and utility of evidence gathering actions. A specific objective of CRL is learning to evaluate the merit of different evidence gathering actions with respect to hypotheses. COMET is also geared towards group learning, whereas TeachMed is intended for individuals.

Adele [16] is an ITS for medical diagnosis, that also uses Bayesian networks to model the expert. Although the approach does not follow a traditional representation of an ID [5] - it is quite reminiscent of the use of IDs in TeachMed. The key difference is in TeachMed's pedagogic model, which uses a more general, hybrid representation. This provides more modularity and flexibility for the implementation of the pedagogic model.

The purpose of this project is to develop an ITS which closely models an existing teaching method for a complex medical reasoning task. In the next section, we describe the educational goals and methods that we sought to replicate in TeachMed, based upon an examination of videos of CRL sessions provided by the curriculum committee, and hands-on experience by one of the co-authors.

3. Educational setting – existing CRL courses

CRL courses are mandatory for clerkship year students (fourth year) at our institution, in every basic medical specialty (internal, surgery, pediatrics, psychiatry, etc.). Students participate in two sessions per week for two hours per session.

Goals of CRL. At the fourth year, students have accumulated significant declarative knowledge. CRL courses aim at helping them structure this knowledge, by applying it to real or virtual patient cases. Thus the reasoning process a student exhibits when solving the case is more important than his final diagnosis. CRL courses in other places have similar objectives [20-23].

Sequences of CRL. A CRL session is conducted live in small groups of eight students under the supervision of a teacher [18-19]. One student is picked to play the role of a patient; another plays the role of the physician and the remaining students are observers who actively participate in the discussion (Figure 1). Queries to the patient are of three categories: interview (e.g., motive of the consultation, symptoms, how frequently they occur, and so on); physical exam (e.g., sensing reflexes), and laboratory/imaging tests (e.g., urine tests, blood tests, x-rays or MRI). Answers to queries are given by the patient student, following a script prepared by the teacher, according to pedagogical objectives fixed by the curriculum committee. Whenever the query has no predefined answer in the script, the teacher intervenes to provide the answer.

Hypothesis triggering and evaluation. The physician student writes on the blackboard evidences and hypotheses, so the teacher can follow his reasoning path. The teacher often tries to make the student focus on one hypothesis at a time, and pursue it until exhausting all evidence that can significantly influence its likelihood. The working hypothesis is often the most likely one or the least likely one. In the first case, the evidence gathering process is driven by the desire to confirm the hypothesis, whereas in the second case, the student is trying to discard it. But there are exceptions to this rule, such as when the actions for gathering the evidence

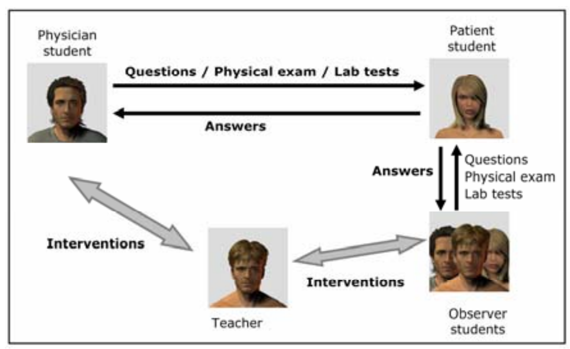

Figure 1. CRL session.

for the least or most likely hypothesis are too costly, so it may be more cost-effective to focus first on another hypothesis. At several points, the student has to recapitulate his findings, and consider whether he should re-order his hypotheses, discard those that have become less likely or add new ones. This phase is called “hypothesis formulation” or “problem formulation”. The process of evidence collection, evidence gathering, hypotheses generation, and hypotheses formulation, is iterated until the student is able to narrow his list of hypotheses to the one or two most probable diagnoses. Once a diagnosis is obtained, there is a short reasoning process about the treatment and management plan for the patient.

4. CRL pedagogic approach

Mixed-initiative feedback. The student or the teacher can break out of the problem-solving process at any time to initiate feedback. In particular, the student may request clinical knowledge recall (e.g., “what are the symptoms of this disease?”), or confirmation on uncertain knowledge (e.g.; “could it be the case that a patient having this symptom can also have this one?”). The teacher may intervene when it appears that the student is stuck or is going astray from the correct reasoning path.

Hypothesis-generation feedback. Failure by the student to generate a hypothesis for which relevant evidence has been gathered often leads the teacher to initiate a dialogue aimed at helping the student recognize the hypothesis he is missing. Incorrect links between gathered evidence and generated hypotheses also lead to teacher interventions. The teacher may also test the student’s level of confidence by questioning his findings.

Evidence-gathering feedback. Queries by the student to the patient that do not influence the working hypothesis may also lead the teacher to initiate a dialogue to check whether the student has changed his working hypothesis but forgot to mention it, or whether the student is assuming an incorrect relationship between evidence and the working hypothesis. The teacher may also intervene to help the student ask cost-effective queries. For instance, the expected level of influence of evidence obtained from physical exams or lab tests on the working hypothesis may be weighed against the degree of discomfort for the patient or invasiveness.

Working-hypothesis feedback. One criterion for selecting the working hypothesis is the cost of collecting evidence. The expected cost of not collecting evidence is another criterion (e.g., not ordering an MRI when there is a risk of missing a life threatening disease). On the other hand, for well organized and coherent investigation, it is not good to zigzag between hypotheses (e.g., asking questions to the patient related to one hypothesis, switching to another then coming back to the previous), hence the student is generally encouraged to stick to one hypothesis as far as possible, taking into account the other criteria and the necessity to arrive at a diagnosis quickly. The teacher may intervene when it appears that the selected working hypothesis is not appropriate.

Problem-formulation feedback. The teacher also guides the student on when and how to formulate hypotheses. If the student looks confused, they may be reminded that this is an appropriate moment for a problem formulation. During the problem formulation phase, the teacher verifies whether the student is following a coherent and well-organized investigation: (1) the student is able to explain the links between evidence and hypotheses; (2) the sequence of working hypotheses makes sense; and (3) specialized hypotheses are generated after their more general parents in the hypothesis hierarchy (e.g., salpingitis, which is an infection of the Fallopian tubes, should not be mentioned before genecology infection). The teacher may suggest the right sequence of reasoning steps, such as the order in which to consider working hypotheses. The feedback can also be a conditional plan about evidence collection steps, for instance by saying that if the next evidence suggests a particular kind of pathology, then we can think of next performing a particular aspect of the physical exam, then depending on the outcome, ordering a particular lab test.

Follow-up feedback. The teacher also helps the students generalize their knowledge by presenting variant cases related to the one under study. For instance, after the student has correctly interpreted evidence gathered from a lab test, the teacher may discuss cases for which the same lab test results would have yielded a different interpretation or course of action. The teacher mentions whether these are cases he experienced himself, cases experienced by colleagues, or cases from the literature. The teacher often summarizes such a discussion using a generalized statement like “each time you see a patient of this age, having this symptom, and with antecedent of this, you must immediately think of that pathology or that one, before considering anything else.” He may also provide links to best practices for the case under study.

5. TeachMed pedagogic approach

The current version of TeachMed is tailored for fourth-year students who are beginning to practice clinical reasoning. Feedback is initiated by TeachMed and covers:

- Generating the appropriate hypotheses;
- Ordering hypotheses;
- Using the appropriate evidence gathering queries for the current working hypothesis;
- Selecting the appropriate working hypothesis;
- Formulating the problem.

The other forms of feedback mentioned above are for future extensions.

6. System architecture

Figure 2 describes the system architecture of TeachMed. Its main components are a student interface, a patient model, a student model, an expert model, a student-model tracer, and a pedagogic model (also called the tutor). Each of these components is described in more detail in the following sections.

7. Student interface

The student interface consists of user interfaces through which the student can select a patient (Figure 3), interview him by asking questions (Figures 4-5), perform a physical exam (Figure 6), order lab tests (Figure 7), and access his health record. Questions are entered in free text, which serves as input for an ontology based search from a list of questions. Matching questions are displayed to the student for him to pick only one to ask (Figure 4). If no question matches, the whole list of questions (including some irrelevant to the case, to test the perspicacity of the student) are displayed, arranged by category, again so the student picks one of them to ask (Figure 5).

The interface provides functionalities for the student to edit the evidence table and the hypothesis table, to positively or negatively link evidence to hypotheses, and to highlight the working hypothesis. Thus the student is able to interact with the patient, collecting evidence, going back and forth between the interview, exam and lab modules, updating the evidence table and hypotheses table, until he feels ready to formulate a diagnosis. The patient's problem statement can be brief as illustrated in Figure 3, or more elaborate including contextual information (e.g., there is an outbreak of flu). A teacher can prepare a case that models the clinical evolution of a previous case, expecting students to use their prior knowledge.

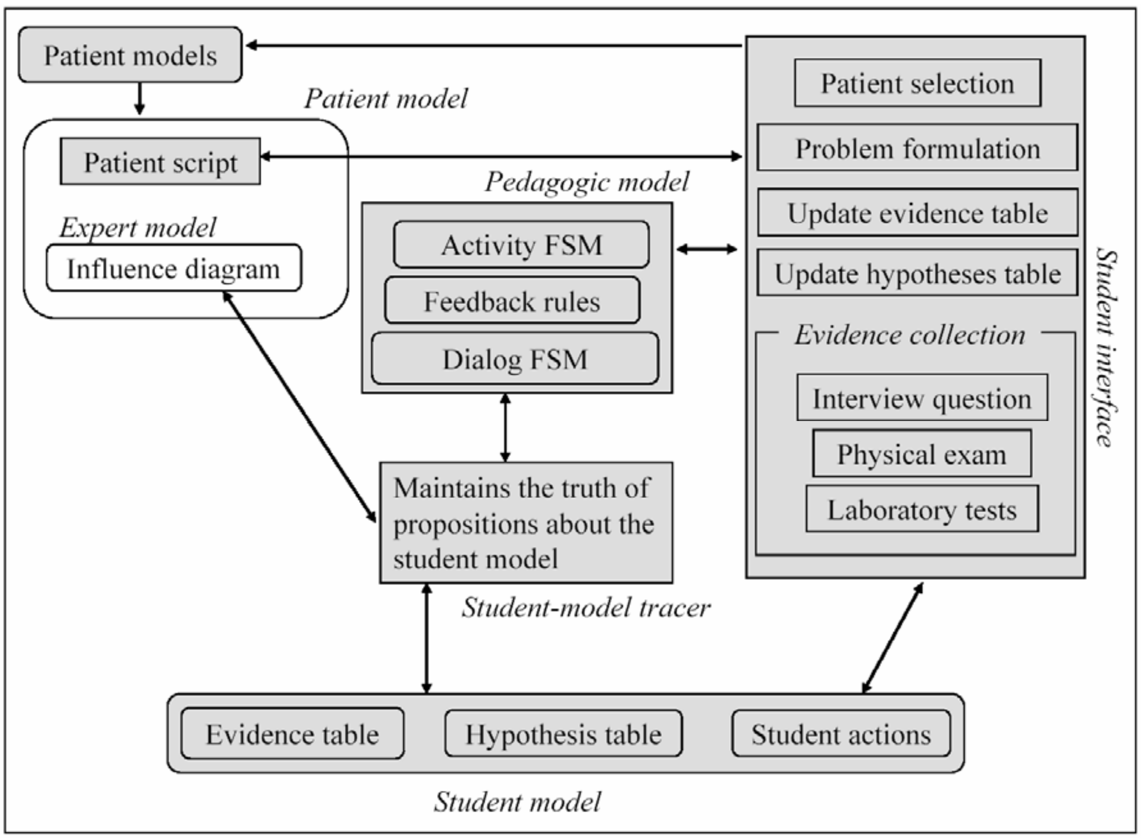

Figure 2. TeachMed's architecture. FSM = finite state machine.

Figure 3. Patient selection.

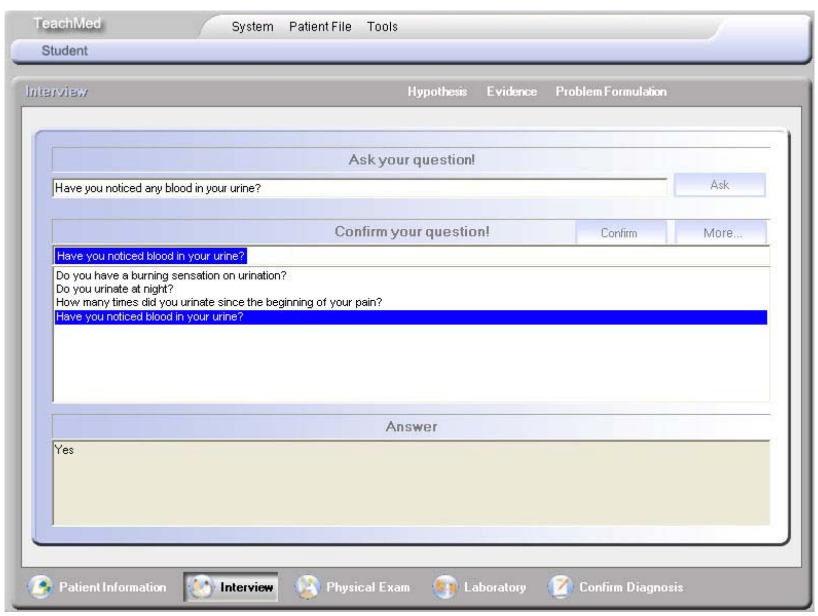

Figure 4. Interview query.

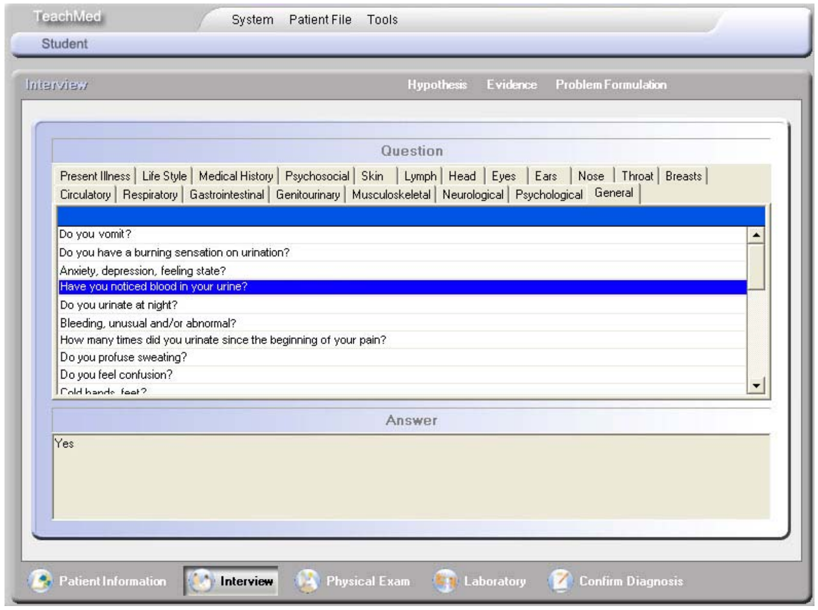

Figure 5. Interview query.

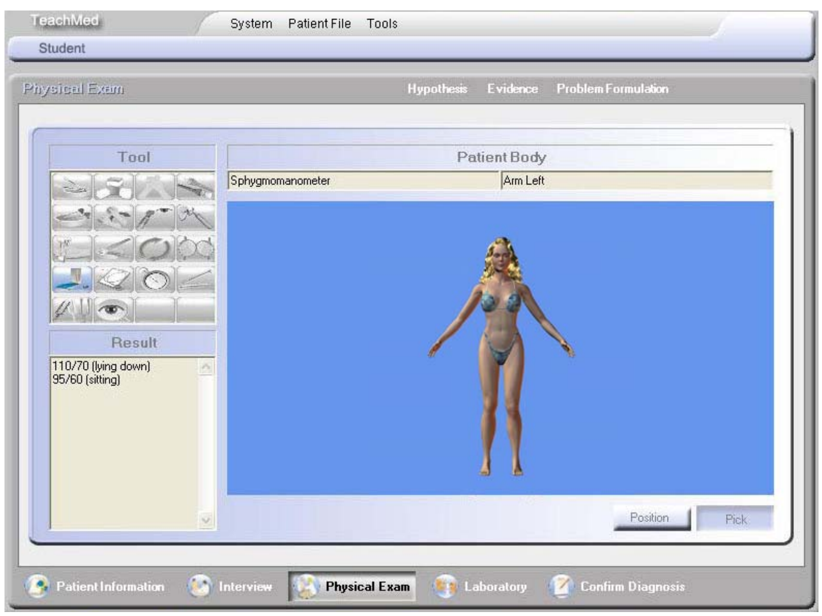

Figure 6. Physical exam query.

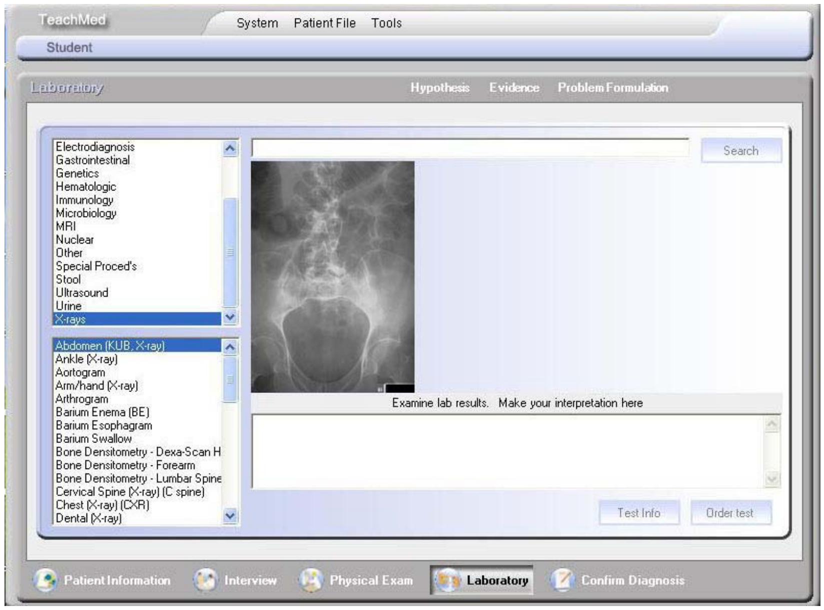

Figure 7. Laboratory test query.

8. Patient model

The patient model consists of a script recording answers to queries that may be asked by the student (interview questions, physical exams and lab tests) and an expert model. The expert model in turn consists of an ID [5].

The use of an ID in TeachMed is reminiscent of medical decision support systems such as Pathfinder/Intellipath [24]. It specifies the causal relationships between symptoms and diseases for a given case, as well as actions related to collecting evidence. Hence we can determine what would be the next diagnostic step from the expert point of view (e.g., the most likely hypothesis given current evidence, or the next best evidence gathering step), then compare with the action taken by the student to provide feedback depending on how far the student's actions are from the expert's. However, as we are in a tutoring setting rather than a decision support system, there are differences between TeachMed and Pathfinder. In particular, Pathfinder systematically recommends a list of hypothesis (i.e., the differential diagnosis) ordered by likelihood given the evidence gathered so far, and a set of evidences to observe that are most cost-effective for narrowing the list of hypotheses. In contrast, TeachMed does not systematically correct the student when he does not gather the most cost-effective evidence or order the list of hypotheses optimally. Rather, feedback is triggered depending on how far the student's actions are from the optimal ones and on the pedagogic strategy (e.g., we may let the student err for some time to see whether he can recover from the situation by himself). Intuitively, PathFinder recommends optimal hypothesis-generation and evidence-gathering actions, whereas TeachMed tries to

confine the student within a space around optimal actions. This confinement is implicit in the tutoring rules that provide guidance to the student, and that are built on top of the ID. Feedback rules can let the student make a few mistakes (e.g., selecting a suboptimal working hypothesis), but then later advise him to what degree he made coherent decisions after the mistake (e.g., evidence gathering was consistent with the suboptimal working hypothesis).
Figure 7 shows a fragment of an ID we use for the case of a patient with an abdominal pain. The fragment illustrates only some nodes with few interview queries. The complete model for this case contains 146 nodes.

9. Student model

The student model consists of the evidence table, the hypothesis table, and a record of the actions performed by the student during his investigation towards a diagnosis that are relevant for feedback. The actions are not limited to evidence gathering queries, but also include actions such as accessing the evidence table, hypotheses table, hypotheses formulation section, and switching between evidence gathering sections (i.e., interview, physical exam, or lab).

10. Student-model tracer

The student-model tracer maintains the truth values of propositions expressing facts about the current status of the student model, with respect to the expert model and the patient model. Examples include:

- query-related-to-working-hypothesis (qrwh). Returns true if the evidence established by the query modifies the probability of the working hypothesis (positively or negatively), given the current evidence.
- query-related-to-some-hypothesis (qrsh). Returns true if the evidence established by the query modifies the probability of some hypothesis (positively or negatively), given the current evidence.
- redundant-query (rq). Returns true if the evidence established by the query does not modify the probability of the working hypothesis (positively or negatively), given the current evidence, but modifies it without any evidence. Intuitively, there is collected evidence (either with the same query or others) that make the current one redundant.
- section-suboptimal-query (sq). Returns true if there exists a query with a higher expected value of information with respect to the working hypothesis than the query currently asked by the student; the more efficient question has to be in the current investigation section (interview, exam or lab).
- query-timed-out() (qt). This proposition is true when the student has not accomplished any action since the last query, for a period of time set by the author (the parameter is different for the three types of query sections: interview, exam or lab).
- section-timed-out() (st). This proposition is true when the student has remained in the current section (interview, exam, lab, problem formulation, evidence table editing, or hypothesis table editing) longer than a time-out parameter set by the author for this section.

The above examples are simple ones we use here for concise scenarios. For full-fledged scenarios, more propositions are involved, in the following categories:

- Recognizing evidence. Given a query asked by the student on the patient, we use the ID to infer the evidence the student is trying to establish.
- Recognizing the working hypothesis. The student is supposed to keep his working hypothesis updated in the hypothesis table. However, as we explained in Section 4, observations from class videos show the student occasionally forgets to record his working hypothesis. To detect such situations, given a query asked by the student, again we use the ID to determine the evidence the student is trying to collect, and to infer the hypotheses this evidence influences. If they do not match the working hypothesis, this suggests the student has either changed his working hypothesis (without updating it), or is collecting evidence that is not related to the working hypothesis. Hypotheses influenced by evidence are determined by dependency analysis [5].
- Determining next best query. Given the current working hypothesis, query costs and utilities, we use the ID to determine the next query that most influences the current working hypothesis.
- Determining most valuable working hypothesis. We use the ID to determine the next best query (i.e., the one maximizing the expected value of information); then we determine the most valuable working hypothesis as the one mostly affected by the next best query (i.e., whose change in probability is higher, whether increasing or decreasing).
- Keeping track of sections. These propositions keep track of the current section the student is in (interview, exam, lab, problem formulation, evidence table editing, or hypothesis table editing) and changes to section (i.e., the student is clicking on the button to move from the current section to a new one).

The above propositions are easily built using approximate inference and dependency analysis algorithms for Bayesian networks [5]. In the worst case, these algorithms have a complexity that is exponential in the size of the network, but in practice they run very quickly depending on the structure of the network. For example, they run in linear time on single-connected networks [5].

11. Pedagogic model

The pedagogic model is used by TeachMed to confine the student into a reasoning space that is implicitly defined by a set of tutoring feedback rules specified by the author. The pedagogic model is specified in four steps.

11.1. Internal variables

The first step is to specify internal variables, on which the preconditions of feedback rules are partly based. For instance, it is possible to define clock variables keeping track of the time elapsed since their initialization.

11.2. Activity FSM

The second step is to define an activity FSM, specifying the potential flow of actions for a student interacting with TeachMed.

The activity FSM is a graph with two types of states: wait and feedback. In a wait state, it is monitoring the type of action performed by the student, until one matches an outgoing transition. In a feedback state, TeachMed is deciding whether to give feedback to the student (i.e., analyzing preconditions of tutoring feedback rules attached to the feedback state); then feedback is given if necessary (i.e., if a rule with a precondition evaluating to true is found); finally a symbol is produced as output, moving along the transition matching the symbol.

There must be a transition for every possible output of the feedback state. Our convention is to produce the symbol “t” for “else” or “otherwise” cases. It is an anomaly if two transitions outgoing from a state have the same label. In this case, the behavior of the FSM can be unpredictable since TeachMed follows the first enabled transition it finds, and the author is not aware in which order they are examined.

Figure 8 shows a simple activity FSM, with only one wait state (W), waiting for a student action: switch to evidence table (set), switch to hypothesis table (sht), switch to hypothesis formulation (shf), switch to interview question section (siq), switch to physical exam section (spe), or switch to lab test section (slt). Wait states are drawn with circles and feedback states are drawn with rectangles. Entry states are indicated by an arrow with no origin state.

Upon perceiving any of the actions, the FSM moves into a corresponding feedback state: for a lab test query (LT), interview query (IQ) , physical exam query (PE), for an update to hypothesis table, for an update to evidence table, or for a hypothesis formulation.

Another way to look at W is as modeling an ongoing process waiting for something interesting to happen (i.e., events on the outgoing transitions), whereas a feedback state is an internal process making some calculation, producing a symbol as output, and then moving to a state along a transition matching the symbol.

Activity FSM can be, of course, more complex than the previous example. For instance, it is possible to have an activity FSM that waits on a sequence of actions (i.e., on an observable temporal behavior of the student), before giving any feedback, such as waiting on a query, then moving to a W on some other query, then to a state where feedback will be given about this sequence of queries. The activity FSMs we use typically contain from 6 to 20 states.

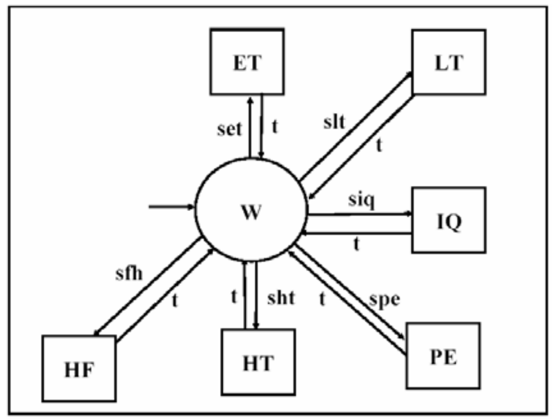

Figure 8. A simple activity finite state maching. ET = feedback for an update to evidence table; LT = feedback for a lab test query; W = wait state; IQ = feedback for an interview query; HF = feedback for a hypothesis formula; HT = feedback for an update to hypo hypothesis table; PE – feedback for a physical exam query; set = switch to evidence table; sht = switch to hypothesis table; shf = switch to hypothesis formulation; siq = switch to interview question section; spe = switch to physical exam section; slt = switch to lab test section; t = true.

11.3. Feedback rules

The second step in specifying a pedagogic model is to provide a set of <u>feedback</u> <u>rules</u> for each feedback state in the activity FSM.

A <u>feedback</u> <u>rule</u> has a precondition that is a conjunction of propositions and <u>temporal</u>

logic statements and a consequent that is a dialogue FSM or an instruction updating some variables of the pedagogic model (e.g., reset a clock variable). Propositions in the preconditions of feedback rules are those maintained by the model tracer or any Boolean function on variables in the pedagogic model.

To illustrate, the following set of feedback rules is an intersection of the feedback rules in states LT, IQ, and PE, that is, they are used for any type of evidence collection action. The precondition and consequent are separate by ":". The negation symbol is noted "!" and the conjunction symbol "&". We use the propositions introduced in Section 10, in their shorthand notations.

1. qt : hesitate()
2. rq : hesitate()
3. !qt & ! rq & !qrwh & qrsh : changed-wh()
4. !qt & ! rq & !qrwh & ! qrsh : incoherent()
5. !qt & ! rq & qrwh & st : formulate-h()
6. !qt & ! rq & qrwh & !st : answer()

The first rule specifies that if the current query is timed out, then TeachMed triggers a dialogue hesitate(), aimed at asking the student whether he is stuck. The second rule is triggered when the student is asking a redundant question and also leads to a hesitation dialogue. With the third rule, the query is not timed out and not redundant; it is not related to the working hypothesis; but it is related to some hypothesis for the clinical case (i.e., a hypothesis in the ID specifying the expert model). In this case, the student may have changed his working hypothesis, but forgotten to update the hypothesis table. The dialogue changed-wh is aimed at checking the situation and assisting the student accordingly. With the fourth rule, the query is not related to any hypothesis, so it is assumed the student is making an incoherent investigation, and the triggered dialogue will assist him. In the fifth rule, the query is related to the working hypothesis, but is timed out for the period allowed to stay in the current section. The triggered dialogue will provide the answer and will then suggest that the student reformulate the hypotheses. Finally, with the sixth rule, the query is related to the working hypothesis and is still within the time limit of the current inquiry section, so the answer is given without any further notice.

The rules we use vary from 1 to 30 rules per state, and have longer preconditions with varying complexity. One source of the complexity is related to adapting feedback to the progression of the student. We know that the quality of the initial hypothesis formulated is important for subsequent reasoning sub-processes, since it will determine how the subsequent phases will be conducted [3]. Conversely, a flawed initial hypothesis formulation could lead the student's investigation in the wrong direction. Thus tutoring feedback has to be provided in the early stages of the student's reasoning process, depending on the level of coaching one wants to enforce. This is achieved by having clock variables keeping track of the time elapsed, counters keeping track of the queries asked by the students and updates made to the tables, and using propositions about these variables in preconditions of feedback rules.

Another source of complexity is related to changing the working hypothesis and testing relevance of queries to the working hypothesis. We have seen that conditions other than change in confidence level have to be taken into account to evaluate the merit of working hypothesis candidates and the evidence that influences them (Section 4). These conditions are conveyed in preconditions of feedback rules in the pedagogic model that are more complex than in the above scenarios.

Preconditions of feedback rules can also use past linear temporal logic (PLTL) formulas [6]. A past temporal logic formula is obtained from a propositional formula by applying the modal operators L (last state), P (previously), A (always in the past) and S (since) to the logic. Formulas are combined using the usual conjunctive, disjunctive and negation connectives, but in addition we can have subformulas prefixed to a temporal connective.

Let's consider a reasoning trace for a student as a trace of the student model and the pedagogic variables, sampled every second (an empirical parameter). Each time tick constitutes a step in the reasoning trace. Given any formulas f and g:

- L(f)at the current step means that f holds at the last step.
- P(f) means that f holds at some previous steps.
- A(f) means that f holds at all previous steps
- S(f,g) means that f holds since g (i.e., g holds on all steps after the latest step at which g holds).

Note that temporal operators can be nested, that is, f could involve any of the temporal connectives. Hence the above informal semantic rules have to be interpreted recursively. This gives an interesting expressive power to PLTL. In fact, PLTL can express any property expressible by a finite state automaton.

To illustrate, we could replace the second feedback rule in the previous example with the following one, expressing that we trigger a hesitation dialogue only after two redundant queries:

```
rq & L(P(rd)) :  fhesitate()
```

If we want three consecutive redundant queries, then we could use

```
rq & L(P(rd) & L (P(rd))):  fhesitate()
```

More complex situations can be handled using more elaborate formulas.

The above semantic rules may suggest that we need to keep an explicit trace of the propositions involved in feedback rules. Fortunately not, since this would be inefficient. Instead, we use a technique called formula progression, originally defined for future linear temporal logic (FLTL) formulas, and consists of labelling states on the fly with the formulas they satisfy [25]. We adapted the technique to PLTL by keeping track, in a variable of the pedagogic model, of the set of subformulas (formulas involved in the preconditions of feedback rules) that are true. At every second we update the set according to the semantic rules of PLTL.

Consequently, at every stage, we have a set containing all subformulas that are true. Hence we can evaluate PLTL by a simple look into the set, just as propositions are evaluated by calling their defining functions (which are either functions of the model tracer or Boolean functions on variables of the pedagogic model).

11.4. Dialogue FSM

The third and final step in specifying a pedagogic model is to describe the dialogue FSMs that are triggered by the feedback rules.

A dialogue FSM is a graph with two types of states: display state and internal state. The structure is reminiscent of that of an activity FSM, but they serve completely different purposes. In a display state, TeachMed displays a multiple-choice message to the student and waits for him to answer by one choice, then moves to the corresponding transition; messages involve place holders (delimited by “<” and “>” in the example below) to be filled by values of corresponding variables at the time they are displayed. In an internal state, TeachMed is making some computations, producing a symbol as output, and moving to a state along a transition matching the output.

Figure 9 shows a dialogue for a student perceived to be changing his hypothesis without explicitly updating his hypothesis table. Display states are drawn with a rectangle containing part of the displayed message. Internal states are drawn in ovals. Entry states are indicated by an arrow with no origin state. The FSM implements changed-wh, which is triggered by the third rule in the example of Section 11.3. This dialogue is triggered when the student has just asked a query that is not related to his working hypothesis. The initial state of the dialogue asks the student whether his query is related to the working hypothesis. If the answer is “yes”, the next state will test the student’s confidence by asking him if he is sure. If he confirms, this is recorded in a variable as an over-confidence (he is persisting in error), otherwise as a lack of confidence (he is not sure of his investigation). In both cases, the next step is to activate a dialogue handling the case of a query not related to the working hypothesis; some transitions in this last dialogue depend on the value of the “confidence” variable from the previous dialogue.

If the student replies “no” on the initial state, the corresponding next state will ask him whether he has finished with the working hypothesis. If he says “no”, he is requested to finish it; that is, the tutoring feedback implemented by this dialogue makes the student examine all evidence related to the working hypothesis before considering a new one. If the student replies “yes”, his confidence is checked by asking him whether he is sure; a confidence variable is updated depending on the answer; then the FSM checks whether in fact there remains evidence related to the working hypothesis (i.e., the “More questions?” state). If there are, this means the student has switched the working hypothesis too soon, so a dialogue for ignoring evidence is activated. Otherwise, the system determines that the student was justified in switching the working hypothesis, but has simply forgotten to update this information. He is asked to do so.

There are several such dialogues in TeachMed. To illustrate, we have dialogues for incorrect updates of the evidence and/or hypothesis table, triggered when the student makes entries that are not inferred by the ID, or omits adding links between evidence and hypothesis that are in the ID.

A dialogue FSM can be quite complex to specify, considering that the purpose is not to provide the answer to the student immediately, but to check first whether the perceived situation is indeed correct, to engage in a dialogue where the student’s confidence will be tested (e.g., by the “are you sure?” question, but this can get more complex), and then to encourage the student to identify the error himself.

It is worth noting that the specification of dialogue FSMs and their trigger feedback rules does not depend on a specific clinical case. It is dictated by the pedagogic strategy one wants to implement. Only the expert model (i.e., its ID) and the patient model depend on specific cases. The pedagogic model takes the specific case into consideration only (1) through the evaluation of propositions that depend on the ID or patient model and that are in the precondition of feedback rules and (2) through place holder variables in dialogue FSM. Authors can of course adopt default pedagogic models, or adapt them.

Combining FSMs and PLTL provides some flexibility. For instance, to provide feedback after a sequence of actions, one can model these sequences as transitions in the activity FSM and attach rules conveying the intended feedback in states ending these sequences. Alternatively, one can use a simpler activity FSM and then recognize the sequences by using temporal logic formulas in the preconditions of feedback rules in appropriate states. Since temporal logic specifications can be more succinct and understandable than FSM, depending on behaviors, and vice-versa, their combination provides additional flexibility.

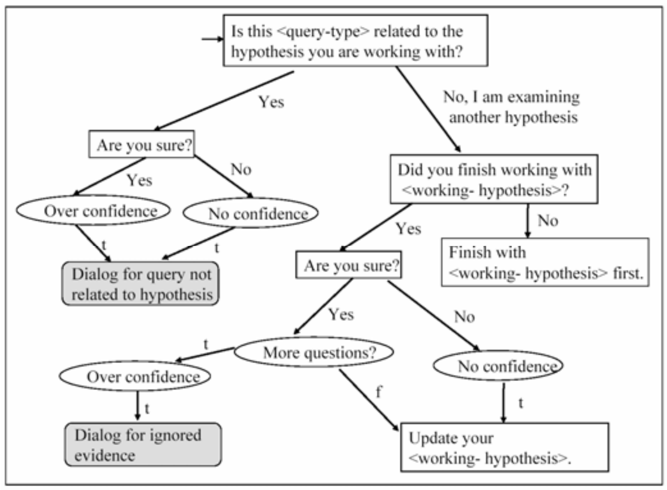

Figure 9. A dialogue finite state machine. t = true; f = false.

12. Examples of feedback in TeachMed

The following scenarios are from the abdominal pain case (Figure 10). The student's actions are preceded by "ST" followed by the type of action. Here we illustrate the student's actions that are evidence collection queries, changes to the working hypothesis, or just answers to TeachMed's Tutor. TeachMed's output is preceded by "TM" followed by "Patient" if this is the patient answering a query from the student or "Tutor" if this is pedagogic feedback to the student.

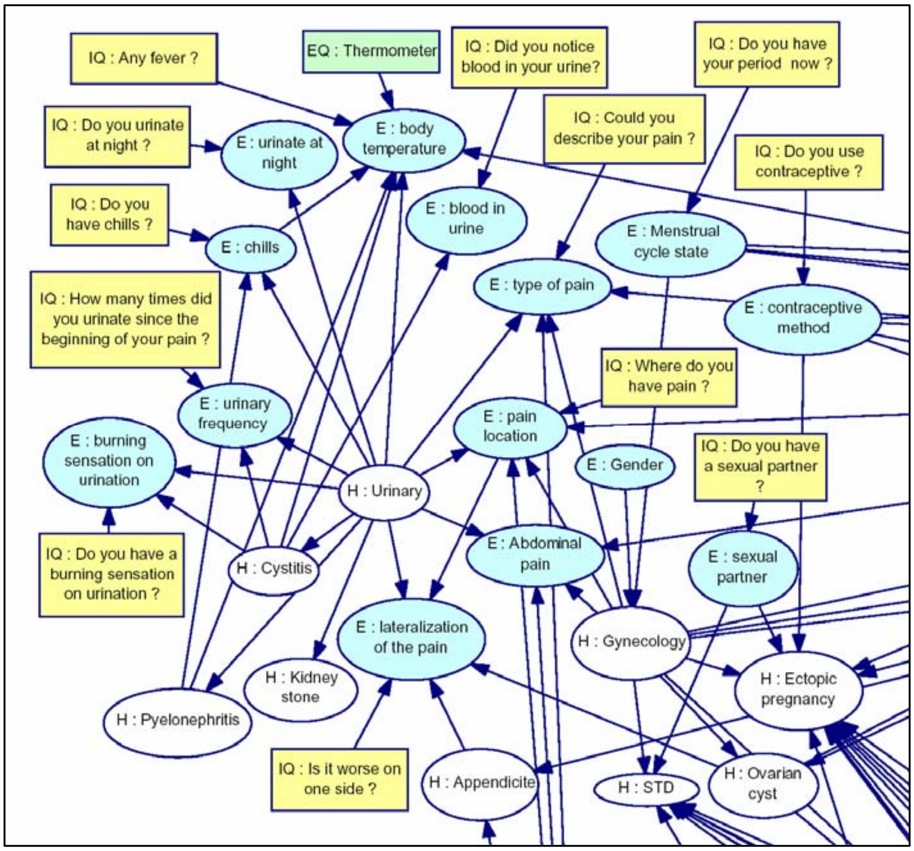

Figure 10. A partial view of the influence diagram for the abdominal case. IQ = interview query; EQ = evidence query; E = evidence; H = hypothesis.

12.1. Feedback when forgetting to record a new working hypothesis

The first scenario illustrates tutoring feedback when the student is attempting to switch from the current working hypothesis to a new one for good reasons (he has exhaustively collected evidence that brings valuable information to the working hypothesis); the only problem is that he forgot to update his working hypothesis.

Collected Evidence: Acute lower abdominal pain
Working Hypothesis: Urinary infection

1. Student Query: Any fever?
   TM Patient: I don't know.
2. ST Query: Could you describe your pain? Is it like a cramp, a burning sensation?
   TM Patient: It's not a cramp but it's very painful.
3. ST Query: Is it worse on one side?
   TM Patient: No.
4. ST Query: How many times did you urinate since the beginning of your pain?
   TM Patient: I don't know. More often I think.
5. ST Query: Do you have a burning sensation on urination?
   TM Patient: No.

6. ST Query: Do you have a sexual partner?
7. TM Tutor: Is this question relevant to the hypothesis you are working with? (Yes / No, I'm examining another hypothesis.)
ST: No, I'm examining another hypothesis.
8. TM Tutor: Did you finish working with urinary infection? (Yes/No)
ST: Yes.
8. TM Tutor: Are you sure? (Yes/No)
ST: Yes.
9. TM Tutor: Change your working hypothesis before proceeding.

To understand how the above dialogue is generated, consider the ID of Figure 10, the activity FSM in Figure 8, the feedback rules in Section 11.3, and the dialogue FSM in Figure 9.

After getting the problem statement, the student is thinking about urinary infection (his working hypothesis). Until step 5, the student has been making queries that are related to the working hypothesis, the situation always matching feedback rule 6. On step 6, the query is not related to the working hypothesis (urinary infection), but is related to another hypothesis (sexually transmitted disease, (STD)). Hence feedback rule 3 is matched, triggering the dialogue FSM in Figure 9. Note the message displayed on step 7. The place-holder "<query-type>" is replaced by "question" because the value of this variable is "question" (meaning "interview question"). The displayed message is a concatenation of the message in the entry state with labels on outgoing transitions.

The remainder of the dialogue can be followed easily from Figure 9, given the choices made by the student when answering multiple choice questions displayed by the dialogue process.

12.2. Feedback when trying to change the working hypothesis too early

The second scenario illustrates feedback to a student who is switching from the current working hypothesis to a new one too early (the current hypothesis still has valuable evidence).

Collected Evidences: Acute lower abdominal pain
Working Hypothesis: Urinary infection

1. ST Query: Could you describe your pain? Is it like a cramp, a burning sensation?
TM Patient: It's not a cramp but it's very painful.
2. ST Query: How many times did you urinate since the beginning of your pain?
TM Patient: I don't know. More often I think
3. ST Query: Do you have a burning sensation on urination?
TM Patient: No.
4. ST Query: Do you have a sexual partner?
TM Tutor: Is this question relevant to the hypothesis you are working with? (Yes/No, I'm examining another hypothesis.)
5. ST: No, I'm examining another hypothesis.
6. TM Tutor: Did you finish working with urinary infection? (Yes/No)
ST: Yes.
7. TM Tutor: Are you sure? (Yes/No)
ST: Yes.
8. TM Tutor: What do you think about the body temperature? (It isn't related to urinary infection./It is related to a urinary infection./I don't know.)
ST: It is related to a urinary infection.
9. TM Tutor: Ask the relevant question.
10. ST Query: Any fever?
TM Patient: I don't know.
11. ST Changing Working Hypothesis to STD
12. TM Tutor: Have you finished with urinary infection? (Yes/No)
ST: Yes.
13. TM Tutor: Are you sure? (Yes/No)
ST: Yes.

14. TM Tutor: What do you think about lateralization of the pain? (It isn't related to urinary infection. / It is related to a urinary infection. / I don't know.)
ST: I don't know.
15. TM Tutor: Lateralization of the pain influences negatively urinary infection as your working hypothesis.
Ask the necessary question.
16. ST Query: Is it worse on one side?
TM Patient: No.

We are still assuming the ID of Figure 10, activity FSM in Figure 8, the feedback rules in Section 11.3, and the dialogue FSM in Figure 9. On step 4, this student asked the question "Do you have a sexual partner?" He has not yet asked the question about fever and location of the pain. Again, the question on step 4 is not related to the working hypothesis, but to some hypothesis in the ID; once more this triggers feedback rule 3, moving to dialogue FSM in Figure 9. On step 6, when asked whether he has finished with his working hypothesis, the student replies "yes". On step 7, TeachMed tests his confidence, and the student maintains a "yes". Between step 7 and step 8, the dialogue FSM is now in state "More questions". In this state, TeachMed examines the evidence table and the working hypothesis (urinary infection), and the ID to determine whether there is evidence not queried yet that could influence the working hypothesis. In this case there are some (body temperature and lateralization of pain), so this leads to a dialogue on unconsidered evidence, which will drive the remaining part of the dialogue from step 8.

12.3. Feedback when generating a specialized hypothesis before its more general parent

Fourth-year students in CRL are often asked to consider specialized hypotheses only after their more general parents. We have tutoring feedback checking situations in which this guideline is violated and corresponding dialogue FSMs that respond. In this case, the student is adding a hypothesis into his table, but some of its parents in the hypothesis hierarchy are not yet present. Note that the hypothesis table is often updated (e.g., to add new hypotheses or links to evidence) without necessarily changing the working hypothesis. Indeed, as the student is collecting evidence on the current working hypothesis, some of it may point to other hypotheses, so he can add them to the table to consider them later if necessary.

In the following scenario, the student is working on urinary infection. Since the pain is not lateralized, the student is diminishing his confidence in urinary infection, but still working further on it, and thinking that it may be salpingitis (an infection of the Fallopian tubes). So he adds the salpingitis hypothesis to the table. This would be a legitimate action for experienced students, but not for beginners who are required to organize their investigation in details, explicitly stating the hierarchy of hypotheses among other things. In this case, the student mentions salpingitis before its parent hypotheses, which are, according to the ID fragment in Figure 11: gynecology infection and gynecology.

Collected Evidence: Young woman, acute abdominal pain, not lateralized, temperature (38.5C)
Working Hypothesis: Urinary infection

1. ST Adding hypothesis: Salpingitis
TM Tutor: Could you specify your hypothesis hierarchy before proposing that hypothesis? (Yes/No)
ST: Yes.
2. TM Tutor: Formulate your hypothesis.
ST Adding hypothesis: Gynecology
ST adding positive link: Gynecology – young woman

ST Adding positive link: Gynecology– abdominal pain
ST Adding hypothesis: Salpingitis
3. TM Tutor: Can you specify your hypothesis hierarchy before proposing that hypothesis? (Yes/No)
ST: No
4. TM Tutor: How about fever? Can you give me a hypothesis more general than Sapingitis related to this evidence? (Yes/No)
ST: Yes
5. TM Tutor: Formulate your hypothesis.
ST: Added hypothesis: Gynecologic Infection
ST: Added positive link: Gynecology – Young woman
ST: Added positive link: Gynecology – Abdominal pain not lateralized
ST: Added positive link: Gynecology – Temperature (38.5C)
ST: Added hypothesis: Salpingitis

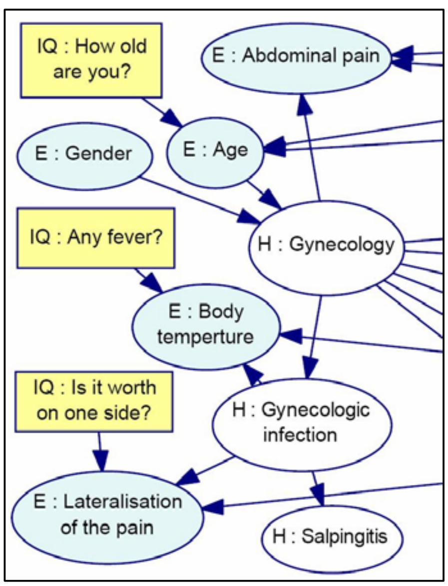

Figure 11. Another partial view of the influence diagram for the abdominal pain case. IQ = interview query; E = evidence; H = hypothesis.

Hints are given to the student about the missing hypothesis by pointing to evidence nodes that influence the hypothesis. For instance, on step 4, the evidence body temperature is used to give a hint about gynecology infection.

13. Implementation

TeachMed is programmed in C# as an internet client-server application. We are currently working on three cases: Abdominal Pain Case, Acute Diarrhea Case and Chest Pain. The ID is implemented using Genie/Smile Bayesian network library [26].

Cases do not currently share clinical knowledge elements. Each case was authored separately by a professional programmer under the supervision of an expert clinician. The programmer created an initial ID structure from a handwritten script given by the expert (this is a script currently used for live CRL sessions for these cases). Then the structure was extended by adding missing nodes following hypotheses, evidence and queries heard from videos of live clinical learning sessions. Working with the expert, the network was iteratively refined, adding subjective probabilities and utilities and fine tuning it to make it behave as closely as possible to the script specifications. Finally, pedagogic FSMs were specified, starting from the general pedagogic state machines explained in this paper then refining them based on the teacher’s interventions observed on the specific video cases.

Feedback rules are compiled into a binary decision tree (handling PLTL formulas as if they were propositional formulas, i.e., a PLTL formula is a node in the decision tree, given that its truth evaluation is a simple lookup into the set of progressed formulas as explained before). Figure 12 shows the decision tree for the feedback rules in Section 11.3.

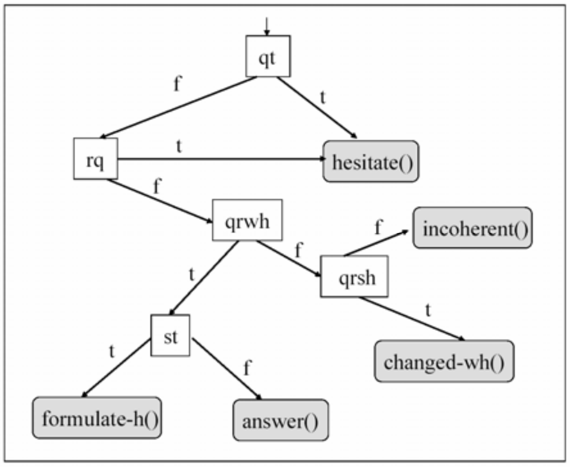

Figure 12. A decision tree for feedback rules. qt = query-timed-out; rq = redundant-query; qrwh = query-related-to-working-hypothesis; qrsh = query-related-to-some-hypothesis st = section-timed-out; t = true; f = false.

14. Future work

Our institution is planning to start using TeachMed to complement live CRL simulations in 2006, to give students more oportunities to practice their clinical reasoning skills autonomously. We expect this to be done with almost no additional incremental cost per student given that they will be learning autonomously. Our evaluation protocol will aim at measuring TeachMed's impact following our previous methodology [19] that is based on the script concordance test, recognized to evaluate clinical reasoning processes that are judged to be essential in a real life setting [23].

Ideally we would like to be able to implement a pedagogic model as efficient as the human teachers in live CRL sessions. The best human teachers are those who have achieved mastery, have strong communication skills, and are able to understand the student's solving steps. To achieve this, our expert model has to be accurate and complete, and allow for very fast inference. Also, our student model tracer must be more advanced than currently implemented in TeachMed.

One difficulty with the use of IDs is in adjusting the probability and action utility parameters. Another difficulty is with having an efficient graph model, without loops, or with as few loops as possible, to improve its performance. These limitations constitute some of the key challenges for our future development of an authoring environment. In fact, Bayesian networks have proven to be extremely difficult to use for non-programmers in other ITS applications for similar reasons [27].

We observed that skill level significantly alters the approach taken by the student in the system. One of our future extensions will be to have a more complex student model, that adapts tutoring help to the level of experience. The expert and pedagogic models will also be extended to take into account uncertainty in the interpretation of collected evidence. With the current implementation the answer received from a query is used to set the corresponding evidence to one of its states depending on the answer. The extension of the pedagogic model will cover other kinds of feedback that are not yet handled, such as planning evidence gathering steps and relating findings for the case under study to other similar cases (see Section 4).

15. Conclusion

Clinical reasoning is a difficult and complex process to model. From a pedagogic perspective, the cognitive reasoning process conducted by the students is more important to evaluate than the final diagnosis [20-23]. It is therefore crucial that patient cases be sufficiently complex to drive students down different reasoning paths. At the same time, the system must be able to evaluate and guide reasoning processes as students traverse the problem-space. For this reason, the TeachMed approach integrates a set of formalisms that can provide a coherent environment for creating cases and tutoring students. IDs, production rules and FSMs, allow for a modular pedagogic knowledge representation. Temporal logic in feedback rules specify trigger conditions based upon previous observations of the student's actions and previous feedback, without explicitly recording a trace or switch in context. Consequently, the author of the pedagogic model can provide an abstract representation of the terms of temporal conditions that enable the feedback.

Acknowledgements

We would like to thank Sylvain Clavette and Yannick Ma for their contributions in the implementation of TeachMed. The Curriculum Committee of the school of medicine has been providing feedback, most particularly Martine Chamberland, Serge Langevin and René Hivon. The anonymous reviewers and the editor gave us very helpful comments that significantly contributed to the presentation of the paper. We are also grateful to The University of Sherbrooke and The Canadian Natural Science and Engineering Research Council (NSERC) for financial support.

References

[1] Gruppen, L. and Frohna, A.: *Clinical Reasoning. International handbook of research in medical education.* Kluwer Academic (2000) 205–230.

[2] Corcos, J.: *Clinical reasoning learning sessions.* In Ann Acad Med, Vol. 30, No. 4, (2001)353-355.

[3] Charlin, B., Tardif, J. and Boshuizen, H.: *Scripts and medical diagnostic knowledge: theory and applications for clinical reasoning instruction and research.* Academic medicine, vol 75., No. 2, (2000)182-190.

[4] Custers, E., Stuyt, P. and De Vries Robbé P.: *Clinical problem analysis (CPA) : a systematic approach to teaching complex medical problem solving.* Academic Medicine, Vol. 75, No. 3, (2000)291-297.

[5] Neapolitan, R. *Learning Bayesian Networks.* Prentice Hall, (2004).

[6] Manna Z. and Pnueli A.: *The temporal Logic of reactive and concurrent systems: Specification.* Springer Verlag, 1992.

[7] DxR Clinician: http://www.dxrgroup.com/ (accessed August 19, 2005).

[8] VIPS: Virtual Internet Patient Simulator: https://www.swissvips.ch/fr/index.htm (Accessed: August 19, 2005).

[9] Wenger E. *Artificial intelligence and tutoring systems -- computational and cognitive approaches to the communication of knowledge.* Morgan Kaufmann Publishers, (1987).

[10] Woolf, B P., Marshall, D., Mattingly, M., Lewis, J. Wright, S., Jellison, M. Murray, T. *Tracking student propositions in an inquiry system.* Artificial intelligence and education (AIED), IOS Press, (2003) 21-28.

[11] Lajoie, S. P., Lavigne, N. C., Guerrera, C. & Munsie, S. *Constructing knowledge in the context of BioWorld.* Instructional Science, 29:2 (2001), 155-186.

[12] Clancey W. *Knowledge-based tutoring––the GUIDON program.* MIT Press, 1987.

[13] Crowley R. and Medvedeva O.: *An intelligent tutoring system for visual classification problem solving.* To appear in Journal of artificial intelligence in medicine, (2005).

[14] Crowley R., Medvedeva, O, and Jukic, D.: *SlideTutor – A model-tracing Intelligent Tutoring System for teaching microscopic diagnosis.* Proc. of 11th international conference on artificial intelligence in education (AIED), IOS Press, (2003)157-164.

[15] Suebnukarn, S. and Haddawy, P. *Collaborative intelligent tutoring system for medical problem-based learning.* Proc. of the ninth conference on intelligent user interfaces, (2004)14-21.

[16] Ganeshan, R., Johnson, L., Shaw, E. and Wood, B.: *Tutoring diagnostic problem solving.* Proc. of conference on intelligent tutoring systems. Lecture Notes in Computer Science, Vol. 1839, (2000) 33-42

[17] Kabanza, F., Bisson, G. *Clinical reasoning learning with simulated patients.* In : Proc. of artificial intelligence in medicine (AIME), Lecture Notes in Artificial Intelligence, Vol. 3581, (2005)385-394.

[18] Chamberland, M: *Les Séances d'Apprentissage du Raisonnement Clinique (ARC) : un exemple d'activité pédagogique contextualisée adaptée aux stages cliniques en Médecine.* Ann Med Interne, Vol. 149, (1998)479-484

[19] Chamberland, M., Hivon, R., Tardif and J., Bédard, D. : *Évolution du raisonnement clinique au cours d'un stage d'externat : une etude exploratoire.* In : Pédagogie Médicale, Vol. 2, (2001)9-17.

[20] Holmboe, S., Hawkins, E.: *Methods for evaluating clinical competence of residents in internal medicine.* Ann Intern Med, Vol. 29, (1998)42-48

[21] Norman, R., Smith, M., Powles, C., Rooney, J., Henry, L. and Dodd, E.: *Factors underlying performance on written tests of knowledge.* Medical Education, Vol. 21, (1987)297-304

[22] Bordage, G.: *Why Did I Miss the Diagnosis? Some Cognitive Explanations and Educational Implications.* Academic Medecine, Vol. 66, (1999):138-143.

[23] Charlin, B., Gagnon, R., Sibert, L. and Van Der Vleuten C.: *Le test de concordance de script : un instrument d'évaluation du raisonnement clinique.* In : Pédagogie Médicale, Vol. 3, (2002)135-144.

[24] Heckerman, D., Horvitz, E. and Nathwani, B.: *Towards Normative expert systems: Part I.* The Pathfinder Project. Methods of Information in Medecine (1992): 90-105.

[25] Kabanza, F., Barbeau M., and St-Denis R: *Planning control rules for reactive agents.* Artificial Intelligence, Vol. 95, No. 1, Elsevier, (1997):67-113.

[26] Druzdzel, M.: SMILE: *Structural Modeling, Inference, and Learning Engine and GeNIE: A Development Environment for Graphical Decision-Theoretic Models.* Proc. of the Eleventh Conference on Innovative Applications of Artificial Intelligence, (1999) 902-903.