

# SlideTutor: A model-tracing Intelligent Tutoring System for teaching microscopic diagnosis

REBECCA CROWLEY

*Center for Pathology Informatics  
Center for Biomedical Informatics  
University of Pittsburgh School of Medicine  
UPMC-Shadyside Cancer Pavilion 307  
5230 Centre Avenue  
Pittsburgh, PA 15232  
(412) 623-1752  
[crowleyrs@msx.upmc.edu](mailto:crowleyrs@msx.upmc.edu)*

OLGA MEDVEDEVA

*Center for Pathology Informatics  
University of Pittsburgh School of Medicine  
UPMC-Shadyside Cancer Pavilion 315  
5230 Centre Avenue  
Pittsburgh, PA 15232  
(412) 623-1248  
[medvedevaop@msx.upmc.edu](mailto:medvedevaop@msx.upmc.edu)*

DRAZEN JUKIC

*Departments of Dermatology and Pathology  
University of Pittsburgh School of Medicine  
UPMC-Shadyside  
5230 Centre Avenue  
Pittsburgh, PA 15232  
(412) 623-2337  
[jukicdm@msx.upmc.edu](mailto:jukicdm@msx.upmc.edu)*

Keywords: Intelligent Tutoring and Scaffolding, Modeling and Representation, Ontologies, Multimedia Interfaces

## Abstract

We report on initial system development of SlideTutor - a web-deployed, image-based, model-tracing Intelligent Tutoring System for teaching microscopic diagnosis. The system is based on our previous work describing the development of expertise in this complex visual diagnostic task. SlideTutor is designed to provide individualized coaching to students as they search, and interpret virtual pathology slides. The system models three important sets of cognitive skills: slide search and lesion detection, visual feature identification, and diagnostic inference. We describe the design and development of the model-tracer for this system, including system architecture, knowledge representation, methods for feedback, and student interface.

# SlideTutor: A model-tracing Intelligent Tutoring System for teaching microscopic diagnosis

REBECCA CROWLEY, OLGA MEDVEDEVA and DRAZEN JUKIC  
*Center for Pathology Informatics, and Departments of Pathology and Dermatology*  
*University of Pittsburgh School of Medicine*  
[crowleyrs@msx.upmc.edu](mailto:crowleyrs@msx.upmc.edu)

**Abstract.** We report on initial system development of SlideTutor - a web-deployed, image-based, model-tracing Intelligent Tutoring System for teaching microscopic diagnosis. The system is based on our previous work describing the development of expertise in this complex visual diagnostic task. SlideTutor is designed to provide individualized coaching to students as they search, and interpret virtual pathology slides. The system models three important sets of cognitive skills: slide search and lesion detection, visual feature identification, and diagnostic inference. We describe the design and development of the model-tracer for this system, including system architecture, knowledge representation, methods for feedback, and student interface.

## 1. Introduction

Diagnostic pathology is the sub-discipline of medicine that provides a disease classification of tissues obtained during biopsies and operations. Pathologists classify disease based on a complex set of visual features apparent under the microscope, supplemented by additional procedures such as special stains, and molecular markers. The assignment of a pathologic diagnosis is a critical aspect of patient care. Pathologic classification is reported to the referring physician and determines therapy and prognosis.

Pathology residencies typically last 5 years. Many trainees opt for subsequent sub-specialty fellowships. Long residencies and sub-specialty fellowships are needed because there are a vast number of patterns to be learned and because many patterns are infrequent, requiring an extended training interval to accrue sufficient exposure. Resident and Fellow pathologists are trained in the apprenticeship style – learning through frequent, intensive, one-on-one interactions with expert pathologists, over a microscope. But new pressures in Health Care and Pathology are increasingly limiting the time available for teaching and learning basic skills in Surgical Pathology and sub-specialty areas. We are investigating the potential use of an ITS in this domain to help bridge this gap.

## 2. Background

Although this project represents the first ITS developed in Pathology, several tutoring systems have been developed in Radiology [1-3], another image-intensive medical sub-specialty. Development of tutoring systems in these two domains poses similar requirements

and challenges including: (1) the need to combine training in both the perceptual and problem-solving aspects of the task, (2) the importance of a comprehensive, carefully selected, controlled and annotated case set, and (3) the challenge of representing domain knowledge related to image interpretation. However, there are significant differences between Radiology and Pathology – with important consequences for ITS development. For example:

- Pathologists look through a microscope, while Radiologists use the naked eye. The microscope can be emulated in an ITS with a Virtual Microscope interface. This interface can be exploited to (1) provide useful information to the model-tracer about the state of the problem, and (2) to structure the delivery of visual hints.
- Pathologic classification is significantly more complex, but also generally more deterministic than radiologic classification. Pathologists have an expressive, shared vocabulary for findings, and many areas of the domain have crisp, standardized classification-criteria. The complexity increases the burden of knowledge acquisition. But the well-defined, diagnostic criteria make a rule-and-frame approach to knowledge representation feasible.

In addition to search and perceptual skills, diagnostic pathology requires skills in classification problem-solving. Accumulated evidence must be associated with the most appropriate classification and sub-classification. Individual visual features may trigger hypothesis, which may be further processed by searching for additional features. Multiple, mutually exclusive, hypotheses may be resolved by recalling and searching for features that best distinguish among them. The GUIDON project [4-6] extensively explored the development of knowledge-based tutors for teaching classification problem solving. GUIDON used MYCIN's rule set to teach medical students to reason about causative organisms in infectious meningitis and bacteremia given a patient's history, physical examination, and laboratory results. Later, Clancey reconfigured MYCIN to create NEOMYCIN, a knowledge representation specifically designed for use in GUIDON 2 [7]. Key insights from the GUIDON project relevant to development of all medical tutors include: (1) the need to represent implicit knowledge, (2) the importance of supporting the abstraction-refinement process of heuristic classification, and (3) the challenges of creating a knowledge representation of sufficient size, complexity and validity to support student learning of real medical tasks.

### **3. Previous Work – A Cognitive Task Analysis of Microscopic Diagnosis**

We have previously used information-processing methods to explore differences in the visual diagnostic processes of twenty-eight novice, intermediate, and expert pathologists [8,9]. Subjects were asked to examine and interpret a set of eight slides while we collected think-aloud verbal protocols and captured on digital video the actual visual data examined with the microscope. We coded video/think-alouds for cognitive processes and errors. Process coding captured aspects of data examination, data explanation, data interpretation and hypothesis testing, and other processes, such as meta-reasoning. Error coding captured critical incidents such as failure to get the lesion under the objective, failure to notice the lesion even though it was present in the field of view, incorrect identification of a visual feature, and incorrect inference from evidence to hypothesis.

The major findings from this study have been previously published [9]: (1) Novices make frequent errors in physically searching the slide (2) Novices frequently cannot distinguish the area of the slide that is diagnostically relevant (3) Both intermediates and novices are error

prone in the identification of visual evidence. (4) Intermediates generally use an explicit strategy of searching for visual cues and reasoning from these cues towards a diagnostic conclusion. Novices lack the knowledge base to do so. (5) Intermediates are error prone in the diagnostic reasoning steps, resulting in diagnostic errors even when they are otherwise accurate in detecting the lesion. Consequently, our system is designed to:

- **Monitor and provide feedback on searching skills.** For example, SlideTutor (i) prevents students from interpreting parts of the slide that are non-diagnostic, (ii) requires that students see the diagnostic area before interpretation can proceed.
- **Assist students in learning to assign the correct term to the specific visual feature.** SlideTutor requires students to identify and refine key visual features that support their diagnosis, corrects identification errors and offers progressive hints when students cannot identify the key features by: (i) moving the viewer, (ii) annotating the image, (iii) providing clues to salient features.
- **Help students learn AND apply the steps for reasoning through a case.** SlideTutor scaffolds the novice's limited knowledge base by providing a graphical goal structure to help students learn the next step, corrects errors in search, identification and inference, and supports efforts to identify and find features that distinguish among hypotheses.

#### 4. System Architecture

SlideTutor is being developed in the Java and Jess programming languages. Our model-tracer utilizes Jess – a Java production rule system (<http://herzberg.ca.sandia.gov/jess/>) [10], and Protégé-2000 (<http://protege.stanford.edu/index.html>) [11,12], an ontology editing environment. Our Virtual Microscope system extends an existing software package for web-based image panning and zooming (<http://www.xippix.com>). The tutor engine is located on our server at the Center for Pathology Informatics. An instance of the tutoring engine is created for each student who logs on to the system.

SlideTutor is accessed from the SlideTutor Project home page (<http://slidetutor.upmc.edu>) where users may download the student interface client with Java Web Start. An icon for the tutor is created on the desktop. On subsequent start-up, the newest version of the interface is automatically downloaded to the client machine, eliminating the need for upgrades. Java Web Start offers an excellent compromise between web-based and standalone systems. It provides web-deployment, true client-server architecture, freedom from browser limitations, and minimal client-management. Figure 1 shows the basic components of the system. Our expert module is a computable rule-and-frame based expert model that dynamically generates the set of valid next steps for a given case and problem state. Student modeling system and Pedagogic model are planned but not yet implemented (shown with gray fill). Agent-based communications between components are based on Java Agent Services (JAS), which incorporates emerging standards of the Foundation for Intelligent Physical Agents (FIPA).

#### 5. Knowledge Representation

SlideTutor is currently being instantiated in the sub-domain of Inflammatory Diseases of Skin. Classification in this domain encompasses approximately 500 diseases, 3000 visual features, and the relationships among them. Classification is commonly expressed as a set of

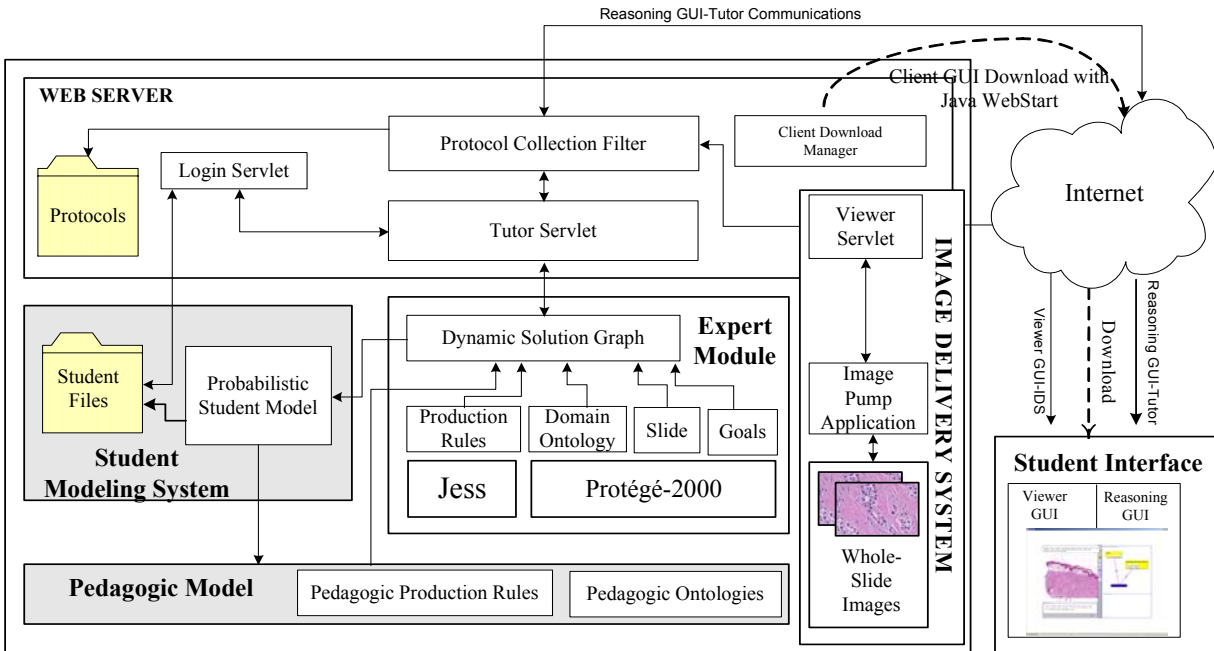


Figure 1. SlideTutor System Design

diagnostic algorithms, roughly akin to decision trees. The solution space is large - we estimate the need to represent approximately one million potential unique cognitive steps to model this single sub-domain. This scale is beyond what is feasible with rules alone, the approach taken by most other medical ITS. Furthermore, an important objective is to create a Knowledge Representation of sufficient abstraction to allow us to easily swap in and out knowledge bases for other domains, even non-medical domains. For these reasons, we created our productions as abstract problem solving methods that dynamically create a set of valid-next-steps after each student action from three sets of working memory elements (frames): (1) the slide representation for a given case, (2) the domain ontology (knowledge-base) and (3) the goal set (domain task structure). The slide representation is a set of frames that describes the location and content of each lesion and feature for a given virtual slide. Each image is associated with a unique slide representation instance, generated with the Slide Authoring Tool during the authoring process. The domain ontology is a set of frames that encodes features, feature attributes, feature values, diseases, tests, and the relationships among them - implicitly representing the structure of the diagnostic algorithms. One domain ontology is used for all cases within its range. The domain ontology is instantiated during the knowledge acquisition process. The goal set is a representation of the abstracted goals that apply in any given domain task. Instantiation of the goal set occurs at run-time, dependent on the case and student, and will ultimately be subject to decisions of the pedagogic model.

**Domain Ontology.** The class concepts and relationships of the domain ontology are general, and should apply widely throughout any domain in which classification of feature based, but the instances are specific to the domain of microscopic diagnosis of Inflammatory Diseases of Skin. The ontology is currently instantiated for one of the eleven diagnostic algorithms – Subepidermal Vesicular Dermatitis. A goal of future work is to extend the model to represent disease entities in all eleven algorithms. The structure of the domain ontology is shown in Figure 2. Diseases are hierarchically represented. For example, the disease Bullous

Pemphigoid is represented as a concrete subclass of Subepidermal Vesicular Dermatitides, which is an abstract subclass of Inflammatory Diseases, which is an abstract subclass of Skin Diseases. Any disease may belong to more than one class (multiple inheritance). Diseases have any number of FEATURE\_SPECIFICATION instances representing the distinct histologic patterns that are to be learned. The classes FEATURE\_SPECIFICATION and DISEASE are related by a pair of inverse slots (has\_feature\_specification and has\_disease) to reduce errors in knowledge acquisition. The redundancy of this inverse slot also permits greater simplicity and efficiency of production rules. Instances of the class FEATURE\_SPECIFICATION include one or more instances of the class FEATURE\_ATTRIBUTE\_VALUE\_SET which is in turn made up of one or more instances of class FEATURE and instances of class ATTRIBUTE\_VALUE\_SET. Instances of FEATURE are intended to represent distinct perceptual primitives of visual entities (such as blister) that form the “atoms” of visual feature recognition. Instances of ATTRIBUTE\_VALUE\_SET are intended to represent the additional cognitive steps for refining these features (such as the distinction of a blister’s location relative to the epidermis as sub-epidermal or intra-epidermal). Individual instances of FEATURE\_SPECIFICATION describe the range of features, attributes, and values that any given case may match against and still be considered to match against a particular instance of DISEASE. Our representation closely follows ontologic classes described by the IBROW group, as a first layer for the Semantic Web [13].

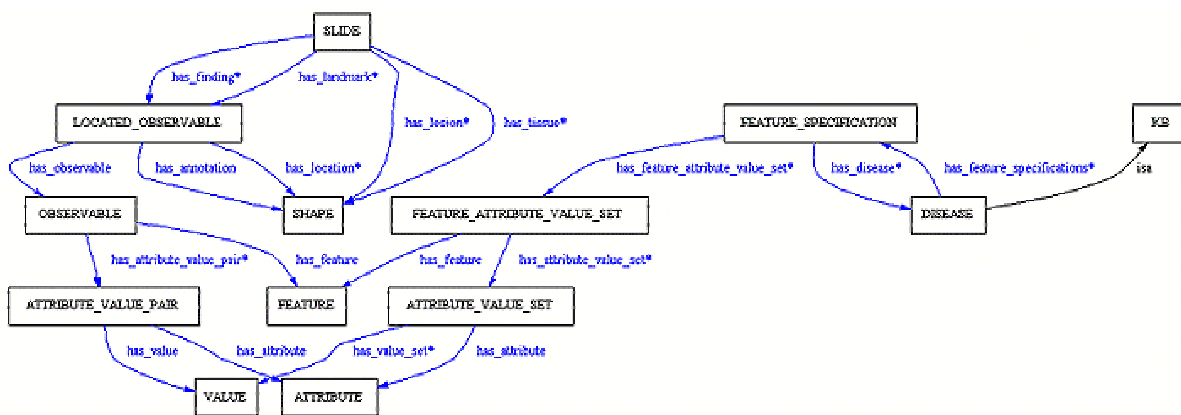


Figure 2. Class diagram of SlideTutor domain ontology

Our domain ontology and all other working memory elements are constructed using Protégé-2000. The flexibility, expressiveness, and ease of integrating Protégé ontologies with Java applications and Jess rule sets provide a powerful platform for building our tutor. Protégé-2000 plug-ins offer useful additional functionality for debugging our system, for encoding frames to standard formats such as XML, and for visualizing ontologic relationships.

**Slide Representation.** Each instance of SLIDE represents the information encoded in a single slide. ATTRIBUTE, VALUE and FEATURE are the same classes used in the Knowledge-Base. Instances of FEATURE and unique sets of attributes and values are represented as instances of class OBSERVABLE. In contrast to FEATURE\_SPECIFICATION instances, instances of the class OBSERVABLE do not model attribute ranges, but rather values specific to the given case (through ATTRIBUTE\_VALUE\_PAIR). LOCATED\_OBSERVABLE represents the presence of an OBSERVABLE instance in a particular location and a SLIDE

instance is built up of any number of LOCATED\_OBSERVABLE instances that encode the symbolic meaning and location of the lesion, normal landmarks, and diagnostic features.

**Goal Set.** The goal set represents goals to be achieved including (1) finding a focal lesion, (2) identifying visual features, (3) modifying features with attributes and values, (4) asserting a hypothesis, (5) linking evidence to hypothesis, (6) setting a goal to find a feature that distinguishes between multiple hypotheses, (7) making a diagnosis, (8) searching for prognostic features, (9) performing additional diagnostic tests. Goals contain priorities that will ultimately be set by the pedagogic model.

**Abstract Problem Solving Methods.** At the start of a problem, the system retrieves information about feature-types and locations from the slide representation. Production rules then fire to create an initial graph of valid-next-steps by incorporating domain knowledge encoded in the domain ontology, case knowledge encoded in the slide representation, and knowledge about the domain task encoded in the goal set. After each student action, additional valid-next-steps are added to the solution graph of possible paths through the problem space. Student actions for subsequent steps (action-selection-input triplets) are matched against valid-next-steps in the dynamic solution graph. Hints are delivered by returning context specific information from the goal with the highest priority. Bug messages are delivered by returning context specific information relevant to individual errors. Currently, the pedagogic constraints model default assumptions. In the future, pedagogic constraints will vary with the state of the student model.

## 6. Methods for Feedback

SlideTutor currently uses methods analogous to other “Cognitive Tutors” to produce (1) hints when the student requests help, and (2) bug messages when the student errs. Hint and bug messages are template text into which context specific variables are inserted. Hint and bug messages are themselves frames, and therefore can be easily altered by a future pedagogic model. In addition to text-based feedback, SlideTutor provides “visual hints” by moving the viewer to an area of interest for the student, and by highlighting particular areas and features of the image. Text and visual cues work together. For example, SlideTutor can highlight a particular feature, and then offer a general hint about the kind of feature, based on knowledge about features in SlideTutor’s domain ontology. For example, the tutor might highlight an area of “neutrophilic infiltrate” and indicate to the student that there is a specific kind of “inflammatory infiltrate” that has special significance. The use of “parent” terms guides students towards the critical features that must be evaluated. SlideTutor will also be able to use the existing case set to retrieve and display correct examples of features and hypotheses that have been incorrectly advanced.

## 7. Student Interface

Microscopic diagnosis is a complex medical diagnostic task, which requires search, perceptual, and problem-solving cognitive skills. Consequently, our interface incorporates two components – a Virtual Microscope interface built on commercial technology ([www.xippix.com](http://www.xippix.com)) and a diagnostic reasoning interface of our own design. The Virtual Microscope interface allows students to interactively pan and zoom in a huge digitized image that captures an entire slide at high resolution. Student actions in the Virtual Microscope interface are evaluated against the dynamic solution graph. SlideTutor can therefore (1) determine the coordinates and magnification of the current field and determine which if any

features are present in that view, (2) change viewer position to demonstrate significant features, (3) detect student selection/pointing in the image and compare these locations to the location of features in it's slide representation, and (4) draw on the image to highlight particular features.

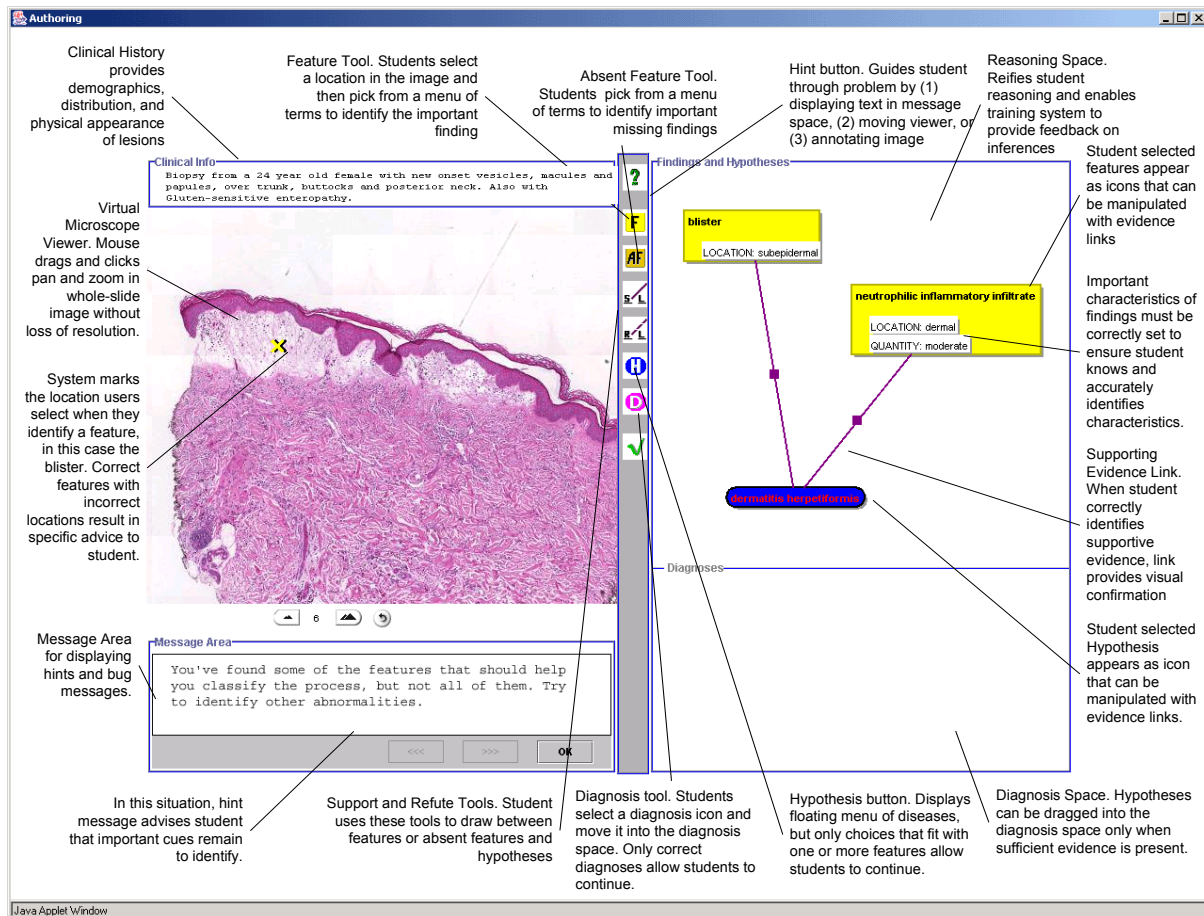


Figure 3. SlideTutor 'node and arc' student interface

Diagnostic reasoning is an abstract process for which no formal notation exists. Therefore an important goal of this project is to develop interfaces that support a deep understanding of the diagnostic process, and efficient learning, without being seen as an undue burden by students. The first of several planned interfaces is shown in Figure 1. In this 'node and arc' interface, students select features (evidence), as well as attributes and values that further refine these features, create hypotheses, and connect evidence and hypotheses with support and refute links. This interface is an adaptation of the graphical interface used in other scientific reasoning environments [14]. SlideTutor detects student identification actions, hypothesis assertions and linking actions, and provides text and/or image annotation as feedback. Valid inferences appear in the diagram, but invalid inferences result in corrective feedback. A second 'algorithmic interface' design uses a tree-based representation to help students learn the most efficient way to navigate through the algorithms toward a set of potential diagnoses. Other interfaces are planned.

## 8. Future Work

We have developed the foundations for a model-tracing ITS in Pathology. Before formative evaluation can begin, we must complete development of an initial set of authored slide instances. In the next year, we will also continue to instantiate our domain ontology, and begin development of the student modeling and pedagogic modeling sub-systems. A key goal is to produce a version of the system that will be used in formative, parametric evaluations, scheduled to begin in late 2003. Specifically, our empirical studies will focus on three questions of general interest to medical tutoring systems: (1) What is the effect of ‘immediate feedback’ versus ‘post-case critiquing’ on learning gains? (2) What options for student modeling offer the best balance between predictive power and speed of evaluation, given the large size of knowledge-bases in medical domains? and (3) Which interfaces and interface metaphors for reifying diagnostic reasoning are most accepted by students, and which are associated with the most efficient learning of domain knowledge?

### Acknowledgements

Funding for this project has been provided by grants from the National Library of Medicine 5-(Training Grant T15-LM07059), Pathology Postdoctoral Research Training Program, and Competitive Medical Research Fund of the Office of Research, University of Pittsburgh Medical School. We thank Valerie Monaco, Vincent Alevan, and Ken Koedinger for their helpful input in the early stages of this work.

### References

- [1] Azevedo R, and Lajoie SP. The cognitive basis for the design of a mammography interpretation tutor. *International Journal of Artificial Intelligence in Education*. 9:32-44, 1998.
- [2] Sharples M, Jeffery NP, du Boulay B, Teather BA, Teather D, and du Boulay, G.H. Structured computer-based training in the interpretation of neuroradiological images. *International Journal of Medical Informatics*: 60: 263-280, 2000.
- [3] Rogers E. VIA-RAD: A blackboard-based system for diagnostic radiology. *Artificial Intelligence in Medicine* 7:343-360, 1995.
- [4] Clancey WJ. *Guidon*. *J Computer-based Instruction* 10:8-14, 1983.
- [5] Clancey WJ. *Knowledge-Based Tutoring - The GUIDON Program*. Cambridge, MA: MIT Press, 1987
- [6] Clancey WJ. Heuristic Classification. *Artificial Intelligence* 27:289-350, 1993.
- [7] Clancey WJ and Letsinger R. NEOMYCIN: reconfiguring a rule-based expert system for application to teaching. *Proceedings of the Seventh Intl Joint Conf on AI, Vancouver, BC*. 1981; 829-835.
- [8] Crowley RS, Naus GJ, Stewart J, and Friedman CP. Development of Visual Diagnostic Expertise in Pathology – An Information Processing Study. *J Am Med Inform Assoc* 10(1):39-51, 2003.
- [9] Crowley RS, Naus GJ, and Friedman CP. Development of visual diagnostic expertise in Pathology. *Proc AMIA Symp* 2001:125-129.
- [10] Friedman-Hill, EJ, “JESS, The Java Expert System Shell”, SAND--98-8206, November 1997.
- [11] Musen MA, Ferguson RW, Grosso WE, Noy NF, Crubezy M, and Gennari JH. Component-based support for building knowledge-acquisition systems. *Conference on Intelligent Information Processing (IIP 2000) of the International Federation for Information Processing World Computer Congress (WCC 2000), Beijing*. 2000.
- [12] Grosso WE, Eriksson H, Ferguson RW, Gennari JH, Tu SW, and Musen MA. Knowledge modeling at the millenium (The Design and Evolution of Protégé-2000). *Stanford Medical Informatics Technical Report - SMI-1999-0801*, 1999.
- [13] Motta E, and Lu W. A library of components for Classification Problem solving. 2000 Pacific Rim Knowledge Acquisition Workshop, Sydney, Australia, December 11-13, 2000.
- [14] Suthers D. Effects of Alternate representations of evidential relations on collaborative learning discourse. In C. M. Hoadley and J. Roschelle (Eds.), *Proceedings of the Computer Support for Collaborative Learning (CSCL) Conference Palo Alto, CA: Stanford University*. 1999; 611-620.