ADAPTIVITY IN A MULTI-AGENT CLINICAL SIMULATION SYSTEM

Sergei Nirenburg[1], Marjorie McShane[1], Stephen Beale[1] and Bruce Jarrell[2]

[1]University of Maryland Baltimore County
Baltimore, MD, USA
{sergei, marge, sbeale}@umbc.edu
[2]University of Maryland School of Medicine
Baltimore, MD, USA
bjarell@som.umaryland.edu

ABSTRACT

We describe several of the adaptive features of a multi-agent environment for training the cognitive decision-making capabilities of medical personnel. First, we give a very brief sketch of the agents in our system and the network in which they are immersed. Next, we illustrate the adaptive character of our system by discussing three areas of adaptivity: a) dynamic changes to the physiology of a virtual patient due to (unpredictable) interventions; b) dynamic enhancement of the virtual patient's knowledge base through learning by asking or being told; and c) contextually adapting the automatic tutor's responses to users' actions.

1. INTRODUCTION

The goal of our work is to create a machine-tractable and human interpretable biomedical and clinical knowledge and simulation environment that a) covers clinical expertise as well as human anatomy, physiology, pathology; b) supports the functioning of a society of artificial intelligent agents; and c) exhibits both the ability to function in situations that have not been pre-scripted and to learn. An inherent characteristic of any system built according to the above principles of modeling is its adaptive character.

The first application system in our knowledge environment – called Maryland Virtual Patient (MVP) – models a team of medical professionals diagnosing and treating a patient. It is an agent-based system in more than one sense, as it involves the simulation of high-level, roughly anthropomorphic agents as well as lower-level processes interpreted as agents. These latter include both domain-related processes (such as diseases) and control-oriented processes (such as event schedulers). Thus, lower-level agents can be viewed as components of high-level ones (a detailed nomenclature of agent types in MVP is presented in [1]).

The high-level agents in the system include the human agent and a number of artificial agents. The human agent, who is typically a medical practitioner or student seeking to improve his cognitive decision making skills, plays the role of the attending physician. The artificial agents in the system include: a virtual patient (VP), lab technicians, specialist consultants and an automatic tutor. The user can diagnose and treat any number of virtual patients selected from the extensive library of patients created (asynchronously with the functioning of the system) by expert physicians and instructors. At the moment of writing, the system covers diseases of the esophagus.

The paper is organized as follows. First, we give a brief sketch of our agent network, focusing on the VP and the tutor, who are the adaptive agents in the current version of the system. The VP is what we call a "double agent", having both cognitive and physiological aspects. Next, we illustrate the adaptive behavior of the VP and the tutor by discussing tasks that underscore it: a) dynamic changes to the physiology of the VP due to (unpredictable) interventions; b) dynamic changes to the VP's cognitive side through learning by asking or by being told; and c) adapting the tutor responses to the tutor's knowledge about what the user does and does not know or has and has not done. Finally, we compare our work with other contributions in the field.

2. THE MULTI-AGENT NETWORK

A session with our system starts when the instructor a) selects a virtual patient from a library of patients and b) launches the simulation. Once the simulation goes past the pre-clinical (asymptomatic) stage, the virtual patient decides at some point to present to the physician. This initiates the interactions among the network of agents.

First, the human user conducts an interview with the virtual patient and hypothesizes some disease or disorder. Based on that hypothesis, he typically runs some tests in an attempt to confirm the hypothesis. Once the hypothesis is confirmed, a course of treatment is prescribed and follow-up visits are scheduled. The interview-hypothesis/diagnosis-test-treatment cycle repeats as necessary.

The control and data flow in our current implementation is illustrated in Figure 1. The semantics of the links in the diagram is as follows. When a user requests a test or intervention, or asks the patient a question, or posits a hypothesis or diagnosis, a corresponding message is sent to the tutor (Arrow 1). The tutor's natural language analyzer extracts the meaning of the message and passes it

to the tutor's reasoning module (Arrow 2). The tutor's reasoning module determines whether the action requested by user is appropriate and generates a response (Arrow 3). The reasoning module uses knowledge from a dedicated tutoring knowledge base (centrally including expert knowledge about "best clinical practices") as well as knowledge available to the user at the time of the request – namely, whatever elements of the VP's profile have thus far been "revealed" based on the user's prior actions (questions, tests, etc.). The tutor's natural language generator transforms the response into text form and sends it to the user (Arrow 4). If the requested action was not appropriate, the message from the tutor will say so and will provide an explanation at the selected level of detail. Note that the tutor's reactions can be shown or hidden depending on the preferences of the user and/or teacher. Even if hidden, the responses are saved in a log that can be reviewed after the simulation.

In all of the following descriptions, we assume the action was either approved by the tutor or the tutor was turned off, thus not blocking ill advised actions.

If the action was positing a hypothesis or diagnosis, the latter is recorded and the system awaits further action. If the action was a request for an intervention, the request is transmitted to the Virtual Patient physiological simulation agent and carried out (Arrow 5a). If the action was a question to the Virtual Patient, the question is sent directly to the VP's cognitive agent, which uses its natural language analysis and generation modules, in cooperation with its reasoning module, to respond (Arrow 5b). If the action was a request for labwork, the request is sent directly to a lab technician or a consultant (arrow 5c). Any interactions between lab technicians/consultants and the VP (arrow 5d) occur independently of the user.

Responses from the VP (from both the **cognitive** and the **physiological** agents that comprise it) are recorded in the subset of the virtual patient's profile that is "revealed" to the user (Arrow 6). It is as if the user were playing the game "Battleship" with the VP.

In the version under development, the user will be able to ask the tutor for help directly, sending queries of the type "What should I do now?" (Arrow 7). The ensuing processes are similar to the ones described by Arrows 1-4 above. The tutor's natural language analyzer will extract the meaning of the message and pass it to the tutor's reasoning module (Arrow 8); its reasoning module will determine one or more appropriate responses (in this case, courses of action) and will send this content to the natural language generator (Arrow 9); and finally, the generator will transform the response into text form and send it to the user (Arrow 10).

3. THE DOUBLE AGENT AND THE TUTOR

The virtual patient is the most complex agent in the system. This is because we model both its body and its mind. In other words, the virtual patient is, for us, a "double" physiological and cognitive agent. The physiological agent is a simulation of physiological and pathological processes. The cognitive agent combines the capabilities of perception (specifically, interoception and language understanding), goal- and plan-based reasoning and action (decision-making and language generation).

The operation of the physiological agent is not directly controlled, though it can be influenced, by decisions and choices of the cognitive agent. Examples of such influences are lifestyle preferences, like diet and regular exercise. The operation of the cognitive agent can, in turn, be influenced by the physiological agent. Indeed, the cognitive agent's choice of goals, and the

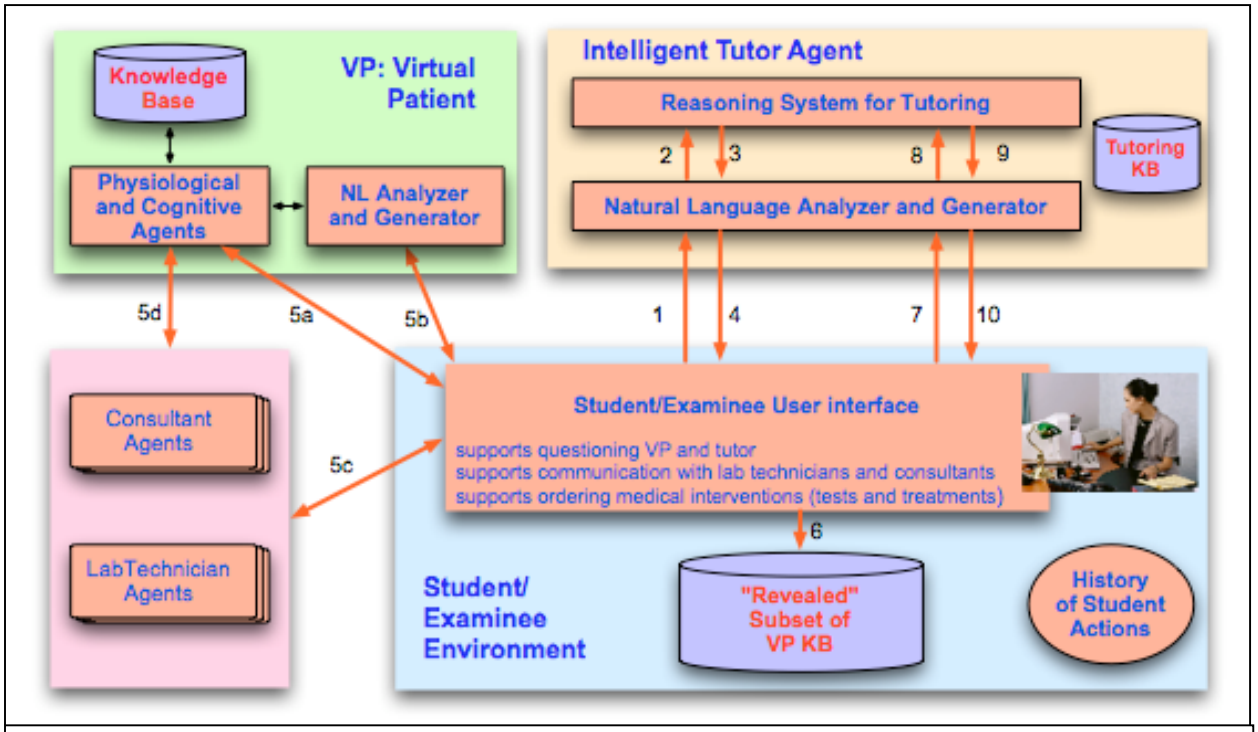

Figure 1. Applying the multi-agent clinical simulation environment to medical training.

choice of plans for their attainment, will be influenced by both the physiological agent's physical state (e.g., disease, fatigue) and its mental state (e.g., stress).

The physiological agent models the body as a collection of anatomical objects and physiological processes, including both normal and pathological ones (diseases). Whenever possible, disease processes are modeled as causal chains of component events and, implicitly, states. These causal chains are encoded as complex events – i.e., scripts – in the system's static knowledge, specifically, in its underlying ontology. However, in many cases, medicine does not at this time possess sufficient knowledge about the biochemical mechanisms of disease progression to allow for the construction of completely causal scripts. This means that disease scripts must often contain a combination causal chains and empirical knowledge about the progression of a disease – what we refer to as clinically derived "bridges". In the current implementation, when it is not possible to encode a causal chain, the progression of the disease is divided into clinically relevant conceptual stages, and a set of value ranges of relevant physiological properties is encoded for the beginning and end of each stage. During simulation, values for interim time points are established through interpolation.

In our knowledge acquisition work we have found that expert clinicians like to express their knowledge in terms of probabilities, e.g., "X% of patients develop stage N of disease D within M months of inception." Though the use of value ranges instead of single values reflects this state of affairs, in our current application system the probabilities do not actually play a central role. This is because the system is supposed to train medical personnel, and this is best done when the instructor can create an inventory of virtual patients carrying a particular disease such that the patients display the full spectrum of disease manifestations, symptom profiles and responses to treatments, not only the most common ones. Presenting users with a carefully crafted set of such patient instances permits all of the necessary learning points to be targeted without the need to spend years of real-world training waiting for the less common patients to present. A case in point: in the evaluation of the SHERLOCK II system, which teaches electronics troubleshooting, it was reported that technicians learned more from using this system for 24 hours than from 4 years of work in the field [8].

Once the progression of a disease reaches the symptomatic (i.e., clinical) stage, the simulation reaches the cognitive side of the double agent. Through interoceptive perception, the cognitive agent becomes aware of symptoms such as pain or difficulty swallowing. Note that the experiencing of symptoms varies widely across patients and, accordingly, cannot be directly linked to given physiological states. However, a fixed inventory of symptoms is associated with each disease and expected ranges of values for each symptom can be asserted for each stage of the disease.

While the physiological agent has been built as a result of the formal encoding of expert knowledge about the progression of diseases and responses to treatments, the knowledge encoded for the use of the tutor agent concentrates on formalizing what expert physicians understand to be best clinical practices – for example, the preconditions that should occur for particular physician actions to be warranted and the best ways of scheduling these actions. Our work on the tutoring agent to-date has concentrated on these contentful issues much more than on the specific pedagogical choices (so-called "tutoring moves") that are at the center of interest in many intelligent tutoring projects in medicine and other areas (e.g., [8]). At present, we have implemented only a small subset of choices available to a tutor. For example, different tutor settings allow the presence or absence of explanations for why certain suggested user actions were blocked. While we will continue to enhance the repertoire of choices in tutoring, our main goals are the adequate simulation of human physiology and the decision-making capabilities of the VP, the encoding of relevant best clinical practices, and the support of realistic natural language dialog between the user and the virtual patient.

4. ADAPTING TO INTERVENTIONS

One highly dynamic aspect of our VPs is the way their physiological state and, accordingly, symptom profile, adapts to both internal and external stimuli. Internal stimuli are the factors, sometimes of unknown provenance, that cause physiological changes to occur if a disease is left untreated. Progressions of such changes are recorded in ontological scripts that are parameterized to permit great variety among patients. However, these scripts are deterministic, meaning that if one launches a specific patient (who has specific inherent characteristics, symptom thresholds, etc.) and does not interfere at all, the course of that patient's disease will always be the same. However, in an interactive environment, the point is to interact: to intervene and see what happens, thus allowing the user to learn by doing, possibly making mistakes and learning both to avoid them and to recover from them. In MVP, any available interaction can be launched at any time, permitting a very large number of variations in a patient's disease path and outcome – at least for diseases that are treatable. The system automatically adapts to processing any of those variations, so that there is no need to write specific scenarios for each of the many possible eventualities that can occur over the course of diagnosing and treating a patient.

To illustrate the adaptive nature of the VP in the face of external interventions, let us consider gastroesophageal reflux disease, or GERD. This disease initiates if the pressure of a person's lower esophageal sphincter (LES) drops below the level where it can act as a sufficient barrier between the stomach and the esophagus. When the LES is deficient in this way – i.e., hypotensive – excessive acidic stomach contents can reflux into the esophagus, detrimentally affecting its lining.

Although a full description of our model of GERD lies beyond the scope of this paper (see [3-4] for details), a few aspects of modeling must be presented for orientation. GERD has widely different manifestations among patients: some patients, if untreated, never progress past

painful but benign esophageal inflammation, whereas others progress to esophageal cancer or other post-inflammatory stages. We capture this distinction by assigning each patient a basic "genetic" predisposition to GERD damage. For hand-authored GERD patients (who are created before a simulation run and are stored in the virtual patient library) this predisposition is assigned manually. But such preparation is not always possible because there are certain conditions under which a patient with some other disease becomes a GERD patient dynamically, during the run of a simulation. In this case the assignment of "genetic" predispositions is done using a probabilistically informed automatic strategy whose decision space (here, the possible paths of GERD if untreated) is ontologically determined.

The speed of progression of GERD depends in large part upon the amount of acid exposure of the esophagus: an extremely hypotensive LES permits more acid exposure than a mildly hypotensive LES and leads to a faster progression of the disease – if the patient is predisposed to the latter stages of the disease to begin with. If an effective medication is prescribed and taken regularly, daily acid exposure drops below the level needed to sustain a disease state, which causes the beginning of the healing process. If medication is taken irregularly, disease states and healing states fluctuate accordingly.

In some cases, the cause of a patient's hypotensive LES is unknown; in other cases it is a side-effect of another disease (e.g., scleroderma esophagus can induce loss of muscle tone in the LES); and in still other cases it can be directly caused by certain actions of a user or the virtual patient itself. It is this last case that is of particular interest here, since it is not known at the start of any simulation *whether* such actions will be taken by the user or the virtual patient, and, if they are taken, *in which order* they will taken and *when*. As such, the simulation must adapt realistically on the fly if and when such interventions occur.

One way in which a physician/surgeon can cause a hypotensive LES is by performing a Heller myotomy, which is a surgical procedure that cuts the LES. This procedure is recommended for patients suffering from the disease achalasia, which renders the LES hypertensive and makes it progressively harder, and eventually impossible, to swallow. A known side effect of a successful Heller myotomy is the initiation of GERD: however, since GERD is readily treatable whereas the inability to swallow is deadly, the benefits of this procedure outweigh its complications.[1] Performing a Heller myotomy is good clinical practice for patients with achalasia; however, in our environment *any* patient can be given a Heller myotomy – even if it is a serious mistake. If, for example, we give a Heller myotomy to a GERD patient with a mildly hypotensive LES (that has a basal pressure of 9 mmHg), the procedure will lower his LES pressure to between 0 and 2 mmHg (randomly selected from this ontologically specified range). According to the GERD model, the rate of disease progression will automatically increase, and the range of medications that will effectively treat the GERD will be more limited. In short, we can launch any expected or unexpected intervention on any patient at any time, and the system will respond in a realistic way based on ontologically recorded knowledge about how procedures can affect patient physiology.

Another type of outside "intervention" is an intervention by the patient itself. One aspect of the basic GERD model is that patients with a mildly hypotensive LES (around 9 mmHg) can reverse their disease progression by stopping GERD-irritating habits, like drinking coffee. As such, a virtual patient with a borderline-hypotensive LES who is sensitive to caffeine but has never drunk coffee before could become a GERD patient if it suddenly discovered the joys of coffee. Similarly, a patient who had GERD but modified its habits such that it went away, could reinitiate the disease by going back to its old habits. This is an example of how the functioning of the cognitive agent in our double agent can affect its physiology. We are currently working on increasing the scope of decisions available to the VP – like deciding to start drinking large quantities of coffee. The next section discusses another important aspect of cognitive agent adaptivity in our approach.

5. LEARNING NEW CONCEPTS

A core component of MVP is the module supporting dialog between the VP and the user. One of the important aspects of creating natural-sounding dialog is to allow different VPs to have different ways of expressing themselves, as well as different degrees of knowledge about medicine and medical terminology. Users should be as adept at conducting an interview with an uninformed patient (in which case many paraphrases of medical terms might be needed) as with a fellow physician. Another important aspect of conducting an interview is educating the VP about its condition, its medication regime, needed lifestyle improvements, options for treatments, and so on. The result of all of this communication has to be learning on the part of the VP: after all, if the user chooses to explain to the VP that the feeling of having something stuck in one's throat is called a "globus sensation", we would expect the VP to remember that 2 minutes later and, perhaps, at the next visit as well. At a minimum, the VP will eventually need to remember the name of its disease, its medications, and so on.

In developing the natural language processing (NLP) support for MVP, dynamically adding to the VPs knowledge repository is of primary concern, and is a good example of an adaptive system module. That is, not only will the patient learn about its condition and treatment through verbal interactions, it will be able to put this knowledge to use in decision-making, one of the key functionalities of the cognitive agent. For reasons of space, in this paper we use a sample interaction to give an informal, content-oriented description of this adaptive process – a more formal analysis would have required a description of the various static and dynamic knowledge resources underlying the system and, specifically, its

[1] In addition, anti-reflux procedures can be performed at the time of Heller myotomy. These are not incorporated into the current version of the simulation engine.

natural language processing component (this information can be found in [2], among others). The adaptive strategies in language processing described below are currently under development.

Suppose that during a patient interview the user asks the VP if the latter ever experiences regurgitation, and suppose that the VP does not understand the term *regurgitation* (that is, the entry for this word is absent from the VP's semantic lexicon).

Background: Each agent in the system is supplied with its own version of the knowledge resources available to the system. This means, for example, that the tutor knows much more about diseases and clinical practices than the VP. The latter's lexicon and ontology are deliberately filtered to reflect an average lay person's knowledge of medicine. During patient authoring, the author selects the level of medical knowledge of the patient, and the lexicon and ontology supplied to the patient are populated accordingly.

The VP will ask for clarification by issuing a dialog turn such as: "What is regurgitation?" The goal of this subdialog is to learn the new term.

First let us consider the eventuality when the human user responds by suggesting a synonym for regurgitation. If the synonym is in the patient's lexicon, then the VP first learns the new lexicon entry, whose semantics will be the same as for the known synonym, and then responds to the original question using this semantic interpretation. As a result of this process, the VP's lexical stock is increased, so that the next time the formerly unknown word is used in a dialog, there will be no need for the clarification subdialog. Of course, the user might provide a synonym that is not in the patient's lexicon, in which case the patient may opt to continue the clarification subdialog.

Instead of a synonym, the user may provide a description of what regurgitation is. On receiving this input, the patient's goal is to match the description to a concept in its ontology. If such a concept is found, then a new lexicon entry for the unknown word (in this case, regurgitation) is created, and the matching concept is used in the entry's semantic description. If no concept is a close enough match, then the VP must learn a new concept.

Suppose the user supplies the following definition of regurgitation: "The return of partially digested food from the stomach to the mouth."[2] And suppose the VP knows all the words in the above explanation.[3] The language analyzer processes this input and comes up with the text meaning representation shown in Table 1 (only relevant information is presented; ontological concepts are shown in small caps):

RETURN-1 (which IS-A MOTION-EVENT)

THEME	INGESTIBLE-105
SOURCE	STOMACH-1
DESTINATION	MOUTH-1

Table 1. A text-meaning representation.

The text meaning representation in Table 1 contains numbered instances of ontological concepts as heads of frames and values of properties. But when comparing this structure with the concepts in the patient's ontology, we disregard instance numbers, in effect treating this text meaning representation as a candidate ontological concept. If a sufficiently close match is found in the VP's current ontology, then the putative new ontological concept is discarded and the already existing best match is used to describe the semantics of the unknown word. If the best match is not considered close enough, then the candidate concept is "promoted" to a regular concept in the ontology. In our example, the search in VP's "lay person" ontology for the best match of the above text meaning representation is the concept VOMIT, a subset of whose properties is shown in Table 2:[4]

	VOMIT
IS-A	ANIMAL-SYMPTOM, MOTION-EVENT
THEME	INGESTIBLE
SOURCE	STOMACH
DESTINATION	MOUTH
VELOCITY	> .8

Table 2. The concept VOMIT in a lay person's ontology.

In other words, the event most closely associated with food coming up that the patient knows about is vomiting. If the two concepts are judged sufficiently similar, the concept VOMIT will be used to describe the semantics of regurgitation. If not, the candidate concept will be included in the ontology and given the name REGURGITATION. If the results of clustering are uncertain, the VP may opt for a continuation of the clarification subdialog by passing the responsibility for this decision to the user, that is, by asking, e.g., "Do you mean vomiting?" If the user agrees that these are sufficiently close, that settles the issue. But if the user considers it important to distinguish between vomiting and regurgitating, then he will respond to the effect that regurgitation is like vomiting but not as forceful. In this case, the learning module in the VP will add the concept REGURGITATION to the ontology; this new concept will be similar to VOMIT but have the additional property that its VELOCITY is lower (the property VELOCITY will be licensed for VOMIT on account of its being an ontological descendant of MOTION-EVENT, for which the property of VELOCITY is defined).

It is important to stress that the "maximum coverage" ontology that our system can use already has the concept REGURGITATION, along with its sibling VOMIT. These concepts are children of BACKWARDS-MOTION-OF-INGESTED-SUBSTANCE, which itself is a child of both

[2] This definition is taken from *The American Heritage Science Dictionary*.

[3] This is actually the case with our language processing resources – the complete (unfiltered) English lexicon in our system covers over 30,000 word senses, and the ontology used to explain these senses consists of over 9,000 concepts, each of which has on average 16 properties defined for it.

[4] The process briefly described here is a special case of learning ontologies and lexicons by reading text and analyzing it using our OntoSem environment for meaning extraction. This work is described in more detail in [9].

ANIMAL-SYMPTOM and MOTION-EVENT, exploiting multiple inheritance. The experiencer of all of these events is a MEDICAL-PATIENT, but the events differ with respect to four properties, as shown in Table 3, illustrating a subset of the knowledge in the "maximum-coverage ontology."

	REGURGITATE	VOMIT
IS-A	BACKWARDS-MOTION-OF-INGESTED-SUBSTANCE	
THEME	BOLUS	BOLUS
SOURCE	ESOPHAGUS STOMACH	STOMACH
DESTINATION	ESOPHAGUS THROAT MOUTH	MOUTH
VELOCITY	< .2	> .8
INSTRUMENT	-	MUSCLE-LAYER

Table 3. Excerpts from the concepts REGURGITATE and VOMIT in the maximum coverage ontology.

The above underscores our commitment to adaptivity in the system. Indeed, we could have made the VP omniscient, at least in the domain of the diseases that it carries. Instead, we chose to model a much more realistic situation without "cheating" by allowing all agents to operate with full access to all knowledge at all times.

6. ADAPTIVE TUTORING FEEDBACK

An automatic tutor can perform a variety of useful tasks. One important role of a tutor is alerting the user to errors and describing unfulfilled preconditions for the action about to be taken. In order for the virtual tutor to make such judgments, it must combine context-independent knowledge with context-specific reasoning.

The context-independent knowledge of our virtual tutor covers, non-exhaustively:

1. **high-level best clinical practices**: e.g., that, in non-critical situations, the physician should interview the patient before all else; that he should posit a working hypothesis or diagnosis before ordering tests and procedures; that each test and procedure should be ordered only if the patient meets certain objective criteria warranting it, etc.;
2. **diseases**: their signs and symptoms over time; the chief complaints they give rise to; other diseases with overlapping signs and symptoms; variations in disease manifestation across patients; what constitutes sufficient evidence to hypothesize, clinically diagnose or definitively diagnose each disease;
3. **diagnostic tests**: the specific criteria that should be met before ordering them; what they test for and what kinds of results they return; potential complications and their frequency;
4. **interventions**: the specific criteria that should be met before ordering them; their projected outcome; potential side-effects and the frequency of those side-effects.

Context-specific reasoning combines this static knowledge with knowledge about the patient and the user that the tutor compiles during the course of the given simulation run. Such dynamically created knowledge includes: (a) the current state and past history of the patient, as elicited by the user through patient interviews, (b) each of the past actions of the user (questions asked, tests ordered, interventions performed, hypotheses and diagnoses posited), and (c) the content of any past interactions between the tutor and the user (the user asking questions and the tutor answering; the tutor intervening to stop the user from making an ill-advised move, etc.). Note that the tutor does not have omniscient knowledge of the patient's physiology since no physician can work from that unrealistic starting point – it knows about the patient exactly what the user knows about the patient. The difference between the tutor and the user in this respect is that the tutor embodies the knowledge of best clinical procedures, as possessed by expert clinicians and encoded by knowledge engineers in the knowledge resources of the system.

The tutor can use its combined static knowledge and reasoning capabilities to function in a number of tutoring modalities, including:

1. providing step-by-step commentary about whether each move is clinically appropriate and why;
2. stopping the user before he carries out clinically inappropriate moves;
3. showing the user how to fulfill the unfulfilled preconditions for a given move.

These three tutoring modalities, as well as a number of variations on the theme, are available in the first release of MVP. Under development is the tutor's ability to suggest the best next move for the user to make. Let us consider one scenario which highlights the adaptive nature of the tutor's reasoning capabilities:

1. The patient presents with the chief complaint "occasional difficulty swallowing."
2. The user interviews the patient, finding out that the only other symptom is occasional mild chest pain.
3. The user hypothesizes the disease 'achalasia.'
4. The user orders an esophagogastroduodenoscopy (EGD). Negative results for the EGD are returned by the lab technician and specialist agents.
5. The user then orders a barium swallow. It reveals a slight narrowing at junction of the stomach and the esophagus (the GE junction).

During the simulation, whether or not the step-by-step commentary function is enabled, the tutor evaluates each move the user makes, saving the evaluations in a log that can be reviewed later by the user and/or teacher. The tutor would approve of step 2 as long as sufficient questions about related diseases have been asked. For example, since the patient complains of chest pain, questions about other symptoms of heart disease should be asked, since chest pain is a primary symptom of heart disease. The inventory of symptoms that should be asked about is *dynamically generated* based on patient responses to each symptom question: the more symptoms the patient

has, the more associated symptoms must be asked about, to rule out similarly presenting diseases.

The tutor would approve of step 3 but would suggest that a better hypothesis would be "motility disorder," which is a superclass of achalasia, since there is no evidence at this point in the proceedings suggesting which specific motility disorder this actually is. The tutor evaluates whether a hypothesis is reasonable on the basis of the filler of the ontological property SUFFICIENT-EVIDENCE-TO-HYPOTHESIZE, which is defined for each disease and class of diseases. If the above property has the value DIFFICULTY-SWALLOWING listed, this is sufficient grounds to hypothesize a MOTILITY-DISORDER, and is also sufficient grounds to hypothesize its child, ACHALASIA, but hypothesizing the more general disorder is better clinical practice in the absence of further evidence – a generalization known by the tutor.

The tutor would approve of step 4 on the basis of the clinical practice of *first ruling out any alarm signals* (i.e., potential immediate causes of danger). Difficulty swallowing can suggest a tumor, which could be cancerous, so the first action should be to rule that out. The need to rule out cancer in the presence of the symptom DIFFICULTY-SWALLOWING is recorded using the property TRIGGER-ALERT in the ontological description of DIFFICULTY-SWALLOWING, as follows:

DIFFICULTY-SWALLOWING
  TRIGGER-ALERT TUMOR (LOCATION ESOPHAGUS)

Fillers of the property TRIGGER-ALERT should be pursued first, even if the related condition is unlikely and does not represent the current working hypothesis. Ontologically recorded knowledge about EGD includes the fact that it can detect tumors. It also includes the fact that DIFFICULTY-SWALLOWING, by itself, is a sufficient condition to order an EGD. As a result, the tutor determines that the user has acted correctly in ordering an EGD as the first study.

Whereas the EGD was ordered in step 4 in order to rule out a potential problem (tumor), the test ordered at step 5 is intended to *provide evidence to confirm the current hypothesis.* Ordering tests to confirm a hypothesis represents clinically correct behavior, since for many diseases (including achalasia) a diagnosis must be made before any treatment can be launched.

In the version of the system under development, the user will be able to ask the tutor what to do next. It is not surprising that the he would ask for help after the results of the barium swallow were received because his hypothesis was not confirmed by the test ordered: the barium swallow would have had to have shown a finding known as "bird's beak" at the GE junction – rather than just a slight narrowing – if achalasia were to be diagnosed. *What should I do now?*, the user asks the tutor.

When the user asks *What should I do now?*, the tutor will review the path taken so far and search for any points at which other moves might have been taken. Formally, this means comparing the preconditions for all the available actions with the knowledge available at the time of each move. In this case, the user acted similarly to the way the tutor would have acted except for the following: (a) as described above, the tutor would have posited a motility disorder, not achalasia; and (b) the tutor might have sent the patient home after receiving the results of the EGD, since the patient was in no danger and its symptoms were very mild (whether to go ahead with the barium swallow or wait and see how the symptoms progress is a judgment call; an experienced diagnostician might have guessed that there would be insufficient evidence to diagnose any disease at this point, based on the patient interview). The tutor would point out to the user the slight deviations from how it, the tutor, would have handled the case, and would tell the user that at this time the best thing to do is to send the patient home, with a recall in a few months or if the symptoms became significantly more pronounced.

This option of "wait and see" is one aspect of the system that both makes it extremely open-ended and trains users to do something that they are known to find uncomfortable – namely, not take immediate action. Most training environments for medical personnel tell them that they must do something immediately and give them a choice of what to do (cf. Section 7). However, physicians frequently must simply wait to see how a disease plays out, reassuring the patient and sending him home. In the scenario above, this is exactly the advice the tutor would give the user. The point, however, is that all of the tutor's decisions are based on comparing its static knowledge with the dynamically changing evidence available about the state of the patient and the knowledge of the user.

7. RELATED WORK

We know of no simulation or tutoring environments that closely resemble the one described here, so comparisons with related work are necessarily rather remote. We orient the review around given features that each system shares with MVP.

Realistic Simulation. One type of computer-aided training involves technical task trainers. [5] These focus on training a specific technical step, with little or no cognitive simulation. Like our environment, they aim to be sophisticated and lifelike.

Cognitive Training. Among the computer methods to train decision making skills are systems based on decision trees that embody diagnostic and treatment algorithms at the case level. These include no biomechanistic processes, and the user is limited to selecting one of the pre-scripted options at fixed points in case. [6] Although these are far more "canned" than the ones in MVP, they too seek to train cognitive capabilities.

[5] For example, manikins to teach the care of infants and adults have been developed by Laerdal, Inc. ("SimBaby", http://www.laerdal.com/) and Meti, Inc. ("The Human Patient Simulator", http://www.meti.com/), respectively.

[6] MedCases, Inc. (http://www.medcases.com) is an e-learning company that develops patient interaction scenarios for continuing medical education.

Hybrid Models. A well-known simulation project is the Virtual Soldier (http://www.virtualsoldier.net/), which simulates the human thorax in the context of penetrating trauma. It combines the Foundational Model of Anatomy with stochastic physiological knowledge at the cell, tissue, body and population levels. Although Virtual Soldier differs from our work in several ways – as by focusing on the short-term treatment of trauma rather than the long-term treatment of patients with ever-changing disease states – the approach integrates various types of knowledge, as does ours.

Intelligent Behavior in Changing Circumstances. Like the well-known expert systems (e.g., Mycin [5]), MVP shows intelligent behavior in ever changing circumstances. However, unlike traditional expert systems, our system is grounded in simulation, stresses language-based interaction, uses the same knowledge bases for both simulation and interaction, and permits free-form interventions – all of which involve innovative uses of adaptive computing.

Large Population of Patients. Sumner and Hagen [6] report a system that is closer in purpose to our work than the traditional adaptive systems. It models a medical certification examination that involves a simulated patient whom the examinee can interview. Importantly, their system, like ours, permits a large population of simulated patients to be created such that users are not all tested on the same patient. However, there are significant differences between Sumner and Hagan's system and ours. For example, while the knowledge in their system covers "health states" of a virtual patient and causal and temporal connections among them (with their associated properties and symptoms), this knowledge does not yet support a realistic simulation of the virtual patient's physiological processes. The probabilistic nature of much of the operation of this system (warranted by the need of creating a "secure" testing environment) suggests the use of Bayesian networks as the underlying representational mechanism, which adds complexity to both knowledge acquisition and processing. Some of the other differences between this system and ours are discussed in [1].

Medical Tutoring. The CIRCSIM project [8] concentrates on tutoring in a medical domain and involves natural language dialog – just like MVP. However, CIRCSIM-Tutor currently does not incorporate simulation, and it covers only one specific medical condition, the baroreceptor reflex – the body's rapid response system for dealing with changes in blood pressure.

Knowledge-Based Dialog. Susan McRoy [7] at the University of Wisconsin, Milwaukee has been developing a dialog system in the framework of a tutoring environment for medical students. She shares our belief in the need for knowledge for language processing but focuses on dialogue issues without a detailed specification of physiological and pathological states.

In sum, whereas other systems and approaches have aspects in common with our system, the overlap is typically with respect to only one feature.

8. FINAL THOUGHTS

The first release of MVP was informally tested and positively evaluated by medical students at the University of Maryland School of Medicine as well as gastroenterologists who were not part of the development team. We are currently working on incorporating our natural language capabilities (which were not invoked in the first release), expanding the reasoning capacity of the cognitive side of the VP and the tutor, and extending the depth and breadth of disease descriptions to provide for the full range of clinical manifestations of each disease covered.

9. REFERENCES

[1] S. Nirenburg, M. McShane, S. Beale, B. Jarrell, B. Johnson, "Diagnosing and treating virtual patients using multi-level heterogeneous networks of human and software agents," Submitted to *Artificial Intelligence in Medicine.*

[2] S. Nirenburg and V. Raskin, *Ontological Semantics*, MIT Press, 2004.

[3] M. McShane, G. Fantry, S. Beale, S. Nirenburg, B. Jarrell, "Disease Interaction in Cognitive Simulations for Medical Training," in Proc. MODSIM World Conference, Medical Track, Virginia Beach, Sept. 11-13, 2007.

[4] M. McShane, S. Nirenburg, S. Beale, B. Jarrell, G. Fantry, "Knowledge-based modeling and simulation of diseases with highly differentiated clinical manifestations," Proc. 11th Conference on Artificial Intelligence in Medicine (AIME 07), Amsterdam, The Netherlands, July 7-11, 2007.

[5] E. H. Shortliffe, "The computer and clinical decision-making: Good advice is not enough," in *IEEE Engineering in Medicine and Biology Magazine* 1(1), pp. 16-18.

[6] W. Sumner and M. Hagen, "Computer Architecture and Process of Patient Generation," Evolution and Simulation for Computer Based Testing System Using Bayesian Networks As a Scripting Language. US Patent 7,024,399, 2006.

[7] S. McRoy, S. A. Syed, S. M. Haller, "Uniform knowledge representation for language processing in the B2 system," *Journal of Natural Language Engineering*, 3(3), 1997.

[8] M. Evens and J. Michael, *One-on-One Tutoring by Humans and Computers*. New Jersey and London: Lawrence Erlbaum and Associates, Publishers, 2006.

[9] S. Nirenburg and T. Oates, "Learning by Reading by Learning To Read," in Proc. ICSC-07. Irvine, CA. September, 2007.