BIO-ALGORITHMS AND MED-SYSTEMS
JOURNAL EDITED BY MEDICAL COLLEGE – JAGIELLONIAN UNIVERSITY
Vol. 5, No. 9, 2009, pp. 57-63

MARYLAND VIRTUAL PATIENT: A KNOWLEDGE-BASED, LANGUAGE-ENABLED SIMULATION AND TRAINING SYSTEM

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

[1]*Univeristy of Maryland Baltimore Country, USA*
[2]*Univeristy of Maryland School of Medicine, USA*

Abstract. Maryland Virtual Patient (MVP) is system aimed at training medical personnel in certain aspects of clinical medicine. The user plays the role of attending physician and is tasked with diagnosing and treating virtual patients (VPs), with or without the help of a virtual tutor. Each VP is composed of both a realistically functioning physiological side and a reasoning – and language-enabled cognitive side. The former permits the VP to undergo the physiological states and changes associated with diseases, their treatments, and even unexpected external stimuli, such as clinically counterindicated interventions. The latter permits the VP to consciously experience and reason about its disease state, make decisions about its lifestyle and medical care, and discuss all of these with its attending physician (the user). This paper provides a brief overview of core aspects of MVP.

Keywords: virtual patient, medical simulation, automatic tutoring, knowledge-based systems

1. Overview

Maryland Virtual Patient (MVP) is a knowledge-based, language-enabled simulation and training system whose goal is to provide medical personnel with the opportunity to develop clinical decision-making skills by managing many highly differentiated virtual patients suffering from various diseases and combinations of diseases. The system seeks to offer a breadth of experience not attainable in a live clinical setting over a corresponding period of time.

MVP is modeled as a network of human and artificial agents, as shown in Figure 1.

The human agent, who is typically a medical practitioner or trainee seeking to improve his or her cognitive decision making skills, plays the role of the attending physician. High-level artificial agents include the virtual patient (VP), other medical personnel like lab technicians and specialist consultants, and an automatic tutor. Lower-level artificial agents include both domain-related processes, such as diseases, and control-oriented processes, such as event schedulers.

Users of MVP can interview a VP; order lab tests; receive the results of lab tests from technician agents; receive interpretations of lab tests from consulting physician agents; posit hypotheses, clinical diagnoses and definitive diagnoses; prescribe treatments, like medication and surgery; follow-up after those treatments to judge their efficacy; follow a patient's condition over an extended period of time, with the trainee having control over the speed of simulation (i.e., the clock); receive mentoring from the automatic tutor, if desired; and repeat the management of a given VP using different management strategies to compare their outcomes. The user can launch any intervention available in the system at any time during the simulation, be it clinically justified or not. In the latter case, if the user inadvertently worsens the VP's condition or initiates a new disease process, he must recover from the error in the continuing simulation.

The core artificial agent, the VP, is a knowledge-based model and simulation of a person suffering from one or more diseases. The VP is a "double agent" in that it models and simulates both the physiological and the cognitive functionality of a human (Nirenburg et al. 2008a). Physiologically, it undergoes both normal and pathological processes and responds realistically both to expected and to unexpected (e.g., by user error) internal and external stimuli. Cognitively, it experiences symptoms, has lifestyle preferences (a model of character traits), has dynamic memory and learning capabilities, has the ability to reason in a context-sensitive way, and can communicate with the human user about its personal history, symptoms and preferences for treatment. User-VP communication is carried out in unrestricted English.

Currently, MVP covers 7 diseases of the esophagus. We chose to initially model esophageal diseases because the esophagus is a relatively uncomplicated organ and because one of the symptoms of esophageal disease, chest pain, can

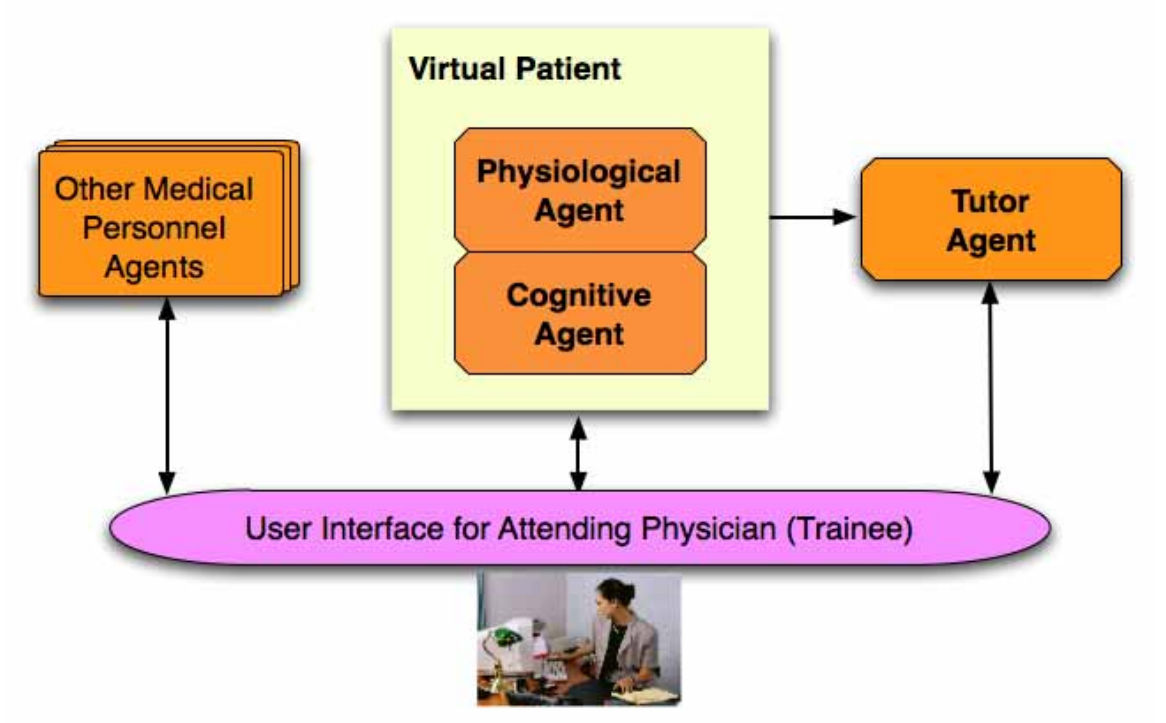

Figure 1. The network of human and artificial agents in MVP.

cause significant diagnostic dilemmas with cardiac disease. All modeling in MVP is done to the grain-size needed to support a realistic simulation. No attempt is made at achieving full coverage of all known medical processes since that endless and amorphous task would stand in the way of practical progress.

A question that frequently arises is, *How can one create unbiased models of aspects of clinical medicine considering that different physicians have different knowledge, experience and opinions?* The answer is that one cannot: a given simulation cannot simultaneously reflect more than one understanding of how things work, and a given tutor cannot simultaneously provide conflicting advice, at least if it intends to help rather than confound the trainee. However, a given *system can* house different models and different tutors, meaning that MVP could include variants 1-*n* of a given disease that reflect the mental models and tutoring advice of *n* different experts.

In the sections below we briefly describe a number of core aspects of MVP, pointing to references that describe them in more detail: the ontological substrate, physiological modeling, modeling best clinical practices, modeling the cognitive agent, and creating a society of VPs. We finish with some concluding thoughts.

2 The Ontological Substrate

The MVP simulation is grounded in an ontologically-defined model of human anatomy and physiology. The ontology, which was initially developed for, and is still used in, the OntoSem language processing environment, is a world model populated by concepts written in a language-independent metalanguage of description (Nirenburg and Raskin 2004). It currently contains around 8,600 concepts – equivalent in overall size to about 135,000 RDF triples – approximately 90% of which are general-purpose. The medical domain has been developed particularly well to support the needs of MVP.

Each object and event concept in the ontology is described by an average of 16 properties, some locally defined and others inherited. Key to the MVP simulation are ontologically recorded complex chains of events that are essentially scripts in the sense of Schank and Abelson (1977). Scripts describe how things typically happen in the world, with all of the relevant temporal and content-related variations on the theme. Examples of high level scripts relevant for MVP include the progression of a disease, the impact of treatments on disease progression, the first visit to a doctor, follow-up visits, and so on. Each high-level scripts breaks down into components scripts, which in turn can have component scripts, eventually ending with singleton events.

3 Physiological Modeling

The VP at the center of MVP is comprised of both normal and abnormal anatomical structures, and can experience both normal and pathological processes. For diseases of the esophagus, the core normal physiological process is swallowing, with its hundreds of component events – nerves firing, sphincters opening and closing, peristaltic muscle contractions moving like waves down the esophagus… Disease processes then affect specific aspects of these anatomical structures and processes.

When modeling a disease in MVP, knowledge engineers help physicians to distill their extensive and tightly coupled

physiological and clinical knowledge into the most relevant subset, express it in concrete terms, and hypothesize about unknown and currently unknowable things, like the nature of disease progression at the pre-clinical (pre-symptomatic) stage. The models thus created are dissimilar to anything physicians themselves have ever encountered and anything that is available in the medical literature. It turns out that even outside of any simulation, the models themselves are useful to trainees as a complementary learning resource, as shown by informal training sessions conducted at the University of Maryland School of Medicine.

One of the challenging aspects of medical modeling is that far from all biomechanisms are understood by the medical community. This necessitates a split strategy for modeling: modeling some processes in terms of biomechanistic causal chains, and others using what we refer to as knowledge "bridges", which derive from practical clinical knowledge, situational knowledge, observations and probabilistic methods.

We model using biomechanistic causal chains when the relevant biomechanisms are known and are relevant for our simulation. Let us take a *highly simplified*, *informally rendered* example of a causal chain related to gastroesophageal reflux disease (GERD) (Jarrell et al. 2007):

If a person's lower esophageal sphincter (the sphincter between the esophagus and the stomach) is hypotensive (too loose), acidic liquid from the stomach leaks up into the lower esophagus.

If liquid with a low pH is in contact with the esophagus for too long, the tissue that lines the esophagus can become inflamed.

If excessively low pH in the lower esophagus continues for long enough, the inflamed tissue can begin to erode.

If the erosive process continues for long enough, an ulcer can develop.

If the tissue lining the lower esophagus is inflamed, eroded or ulcerated, and if that tissue is exposed to an acidic substance, the person can experience heartburn.

If damage to the esophageal lining continues for too long, the person can develop esophageal cancer.

If the pH of the liquid in the stomach is raised (as by medication), the refluxing of liquid into the lower esophagus typically has no harmful effects and healing of the tissue occurs: an ulcer heals to an erosion, which heals to the inflammatory stage of the disease, and the inflammation ultimately goes away, leaving normal tissue. Cancer is not reversed by pH-raising medication.

Note that our rules say that things *can* happen because the esophageal lining of different patients reacts differently to acid exposure, with some reactions being far more common than others, of course.

Modeling using biomechanisms is preferable when possible not only because of its verisimilitude but also because it permits the VP to respond realistically even in the face of unexpected interventions (McShane et al. 2007a). For example, say a VP has the esophageal disease Zenker's diverticulum, which involves the upper esophagus and has nothing to do with the lower esophageal sphincter. And say the user mistakenly orders the surgical procedure Heller myotomy, which cuts the lower esophageal sphincter. This will automatically give the patient GERD, meaning that the user will not only have failed to effectively treat the Zenker's diverticulum, he will also have given the patient a secondary disease that must be managed for the rest of the patient's life. There is nothing in the model of Zenker's diverticulum that explicitly says what will happen if a patient suffering from the disease has its lower esophageal sphincter cut; there need not be – Heller myotomy is ontologically described as rendering the experiencer's lower esophageal sphincter extremely hypotensive, and an extremely hypotensive lower esophageal sphincter is ontologically described as giving rise to GERD in humans.

When biomechanisms are not known, or when they are not relevant for our simulation (typically being of a grain-size that will not play a role in the system), we bridge the gaps in the various ways mentioned above. For example, it is currently unknown why untreated GERD progresses to different levels of severity in different patients: some GERD patients never progress past the inflammation stage of the disease, others progress to only to erosion, still others progress to ulcer, and some develop cancer. Until some genetic or other determining factor is discovered, each VP in the system – no matter what its disease – is explicitly provided with a "GERD path" – just as it is provided with a height, weight, race and so on. The GERD path indicates which path the disease will follow if the VP should ever get GERD (McShane et al. 2007b).

Another example of a knowledge bridge is the use of temporal rather than causal chains to model diseases for which causal chains are unknown. One such disease is achalasia, a disease in which the lower esophageal sphincter, over time, becomes hypertensive (too tight), thus rendering swallowing increasingly difficult. This disease is modeled as being composed of 5 conceptual stages that reflect important clinical landmarks. Changes of relevant physiological and symptom-related property values through those conceptual stages, using functions that interpolate values for each moment in time, define the disease path. The changes in values for some properties over time is fixed across patients, while the values for others are variable within a specified range. This means that each disease model is sufficiently constrained so that all VPs suffering from the disease show appropriate manifestations of it (the disease is recognizable), but it is sufficiently flexible to permit different VP instances to present with differing clinical manifestations.

4 Modeling Best Clinical Practices

In addition to disease models, MVP includes models of best clinical practices that permit the automatic tutor to evaluate user moves and provide guidance throughout the management of a patient. These patient management scripts not only cover when it is appropriate to order a given test, procedure or medication – information that can be found in various published manuals – they also include crucial knowledge about the *process* of diagnosing and managing a patient, such as when it is more appropriate to make a generalized hypothesis ("this patient appears to have a motility disorder") than a specific hypothesis ("this patient appears to have the disease achalasia").

There are many capabilities that could be desirable of a virtual tutor, a representative inventory being described in CIRCSIM (Evens and Michael 2006). Developing a maximally robust tutor is not one of the near-term goals of our work, as

we are not pursuing predominantly pedagogical issues such as whether or not an automatic tutor should provide hints and if so, when and how. Our plan of tutor development consists of the following three stages.

The Stage I Tutor, which is already implemented, knows the "preconditions for best practice" for hypothesizing each disease, definitively diagnosing each disease, and ordering each test and procedure. This means that each time the user carries out a related move, the tutor can agree with it or not. Tutor responses can be of various types, depending on user settings. The tutor can dynamically show all fulfilled and unfulfilled preconditions for the move, show only unfulfilled preconditions, or simply respond that the move is not valid with no further information. Whether the tutor is enabled or disabled, all of the responses it made or would have made during the session are stored for post-session review. To take one example, if the tutor is enabled and the user decides to order a Heller myotomy but has not yet posited the diagnosis of achalasia, the tutor will block the move and – given the first tutor setting above – will say that the precondition for a Heller myotomy is a definitive diagnosis of achalasia. If the user then immediately attempts to posit a definitive diagnosis of achalasia but has not yet gathered all of the necessary test results to confirm that diagnosis, the tutor will respond, e.g., that although preconditions X and Y have been fulfilled, precondition Z has not… and so on, until the user carries out a sequence of moves, each of which is justified by the user's current state of knowledge about the VP.

The Stage II Tutor, which is under development, must be able to respond to the user's question "What should I do next?" This requires considerably more domain knowledge recorded as diagnosis and treatment scripts because the tutor must be able to select a single one of possibly many valid (according to Stage I knowledge) moves, and it must be able to explain that choice to the user.

The Stage III Tutor will permit the user to ask open-ended questions. Our long-term plans involve enabling the tutor not only to reason over the knowledge in our local knowledge bases but also to detect when outside knowledge is needed. In the latter case, the tutor must be able to find the relevant information on the Web or in some other available digital resource and return it as a targeted English response, not a pointer to some number of potentially relevant articles. This capability is actually not as far beyond the current state of the art as it might sound because of the strong language processing capabilities already available in the system (e.g., Nirenburg and Raskin 2004; Beale et al. 2004; McShane et al. 2008a,b).

5 Modeling the Cognitive Agent

The cognitive side of the VP currently models several aspects of cognitive processing:

- **interoception**, which is the perception of physiological phenomena, such as symptoms, and the interpretation and remembering of such phenomena
- **decision making**, including deciding when to go see a physician, both initially and during treatment; deciding whether to seek help in making decisions related to treatment by asking the user knowledge-seeking questions about a recommended test or intervention; and deciding whether to agree to a recommended test or intervention
- **natural language processing**, including language perception and understanding; this involves interpreting both the direct meaning of physician-user communication in natural language and its intent, deciding on what, specifically, to communicate to the user, and actually generating natural language utterances
- **learning**, which involves receiving from the user new knowledge about the world and the words and phrases used to describe it, and adding them to the ontology and lexicon, respectively (Nirenburg and Oates 2008; Nirenburg et al. 2008b)

We will briefly touch upon each of the above points with the goal of describing *what* the VP can do rather than *how* it does it. A sufficient discussion of the latter would require far more space.

Interoception. Interoception is the perception of physiological phenomena. It is a VP feature that has both physiological and cognitive aspects. The source of interoception is physiological phenomena, like symptoms of a disease, hunger and sleepiness. Here we focus on symptoms of a disease because our disease models to date have not required the tracking other kinds of interoception.

The VP experiences current symptoms of its disease and has memories of past symptoms so that useful comparisons can be made. For example, it might reason (of course, not using natural language), *Symptom X has gotten much worse, I had better go see my doctor sooner than our next scheduled appointment.*

Memories are stored using an ontologically grounded metalanguage that is identical to the one used to represent the meaning of language input. Of course, when memories about interoception are stored, there need be no translation into and from a natural language: the entire process occurs at the level of the metalanguage.

The experiencing of symptoms is individualized for each VP instance through the use of character traits and physiological features. Our current inventory of character traits includes trust (in the doctor's advice), suggestibility (how readily the VP agrees to the doctor's recommendations) and courage (how willing the VP is to undergo tests or procedures even if they are risky or have significant side effects). Our current inventory of physiological traits includes physiological resistance (e.g., how well the VP tolerates treatments), pain threshold (how much pain the VP can stand) and the ability to tolerate symptoms (how intense or frequent symptoms have to be before the VP seeks medical attention). This inventory will be significantly expanded in the future. When a given VP is created, values for these features are selected and affect the VP's reactions in the face of its disease(s). Of course, values for the physiological aspects of the disease(s) and the VP's response to interventions, should they be applied at various times, are selected for each individual VP during the process of patient authoring (cf. Section 6).

Decision-Making. The decision-making behavior of specific instances of virtual patients is parameterized using a model of personality traits and physical and mental states. It is informed by (a) the content of the VP's short-term memory, which is modeled as knowledge invoked specifically for making the decision at hand, and (b) the content of the VP's long-term memory, which is its recollection of its past states of health,

past communications and decisions, and general world knowledge.

A VP's decision-making is affected by the severity and duration of its symptoms, its knowledge of tests and procedures, the character traits trust, suggestibility and courage, and the physiological traits physiological resistance, pain threshold and the ability to tolerate symptoms.

VP reasoning is carried out through modeling the VP's goals and plans, thus broadly conforming to the belief-desire-intention (BDI) approach to developing intelligent agents (e.g., Bratman 1999).

When the VP starts to experience symptoms it can either do nothing, go the doctor, go to the emergency room or self-treat. To decide when to see a doctor for the first time, the VP compares its symptom severity with its ability to tolerate symptoms. Later in its treatment the VP also considers the date of its next scheduled appointment, whether or not its symptoms have spiked, what the doctor told it to expect, and so on.

When the doctor recommends a test or procedure, the VP must compare its knowledge of the test/procedure with its character traits, like courage and suggestibility. For example, if it knows nothing about the test/procedure and has little trust in the doctor, it will ask questions about the properties that interest it, like the expected pain and side effects; by contrast, if it has complete trust in the doctor and a high value for suggestibility, it will ask no knowledge-seeking questions and, instead, will agree to anything the doctor suggests.

When the patient has received all the information it wants about a test/procedure, it will decide whether or not to agree to it. It can also suggest other options that it happens to know about (based on its personal ontology), and the doctor can accept or reject such suggestions.

Language Processing. Our approach to treating language communication is unlike most other approaches in that all language-oriented reasoning is carried out on the basis of formal interpretations of the meaning of linguistic expressions. Our automatically generated, semantically-oriented text meaning representations (TMRs) are written using the same ontological knowledge substrate and the same ontologically grounded metalanguage as are used to represent physiological processes, interoception and agent goals and plans. In short, all knowledge and reasoning in our environment employ the same metalanguage, so whether a VP experiences new symptoms (through interoception) or learns information about its disease from the user (through language processing), the new information is be stored the same way in the VP's memory.

There are several advantages to orienting an agent's language processing around TMRs rather than text strings. First, TMRs are unambiguous, since linguistic ambiguity is resolved as the TMRs are being generated. Second, TMRs reduce to a single representation many types of linguistic paraphrase, be it lexical (*esophagus ~ food pipe)*, syntactic (*I will administer it to you ~ It will be administered to you by me*) or semantic (*Does the food get stuck when you swallow? ~ Do you have difficulty swallowing?*) (McShane et al 2008 a,b). Third, TMRs facilitate the detection of which aspects of meaning are central and which are of secondary importance. As regards paraphrase processing, in addition to having to resolve linguistic paraphrase, the VP must be able to resolve two other kinds of paraphrase: a) the reformulation of the representation of physiological events (e.g., symptoms) in "lay" ontological terms that can be understood and remembered by its cognitive agent and b) the representation of the meaning of verbal messages in terms compatible with how related content is stored in the cognitive agent's memory. Remember, the VP is typically not a medical professional, meaning that it must have a different ontology and a different lexicon than a physician would.

Learning. We just noted in passing that the VP's ontology and lexicon do not match those of a physician. Indeed, the physician's ontology will contain a vast subtree of medical information including objects, events and the properties that link them as well as script-based knowledge, which permits the physician to understand the progression of a disease, how to treat it under various circumstances, etc. The physician will have a correspondingly large technical and non-technical vocabulary (lexicon) that is linked to the respective ontological concepts and is used to analyze and generate language in the medical domain. The patient's knowledge base, by contrast, will typically include an impoverished medical subtree in the ontology and a relatively small number of medical terms in the lexicon – unless, of course, the VP happens to be a physician or even a specialist in the given domain.

In conducting an interview with a VP, the user must be able to express himself in different ways, using paraphrases selected according to the degree of medical knowledge the VP possesses. During appointments the physician will naturally teach the VP about various aspects of its condition: its name, the names of related drugs and procedures, the properties of drugs and procedures that the VP asks about or the user chooses to provide, the medical terms for words that formerly had to be paraphrased for the VP, and so on. For example, when the user tells the VP the name of its disease, that disease is added to the VP's ontological subtree of diseases and a new lexicon entry is created that maps to this ontological concept; and when the VP learns information about a test or procedure, it remembers it and no longer asks those questions about it – unless, of course, the VP forgets, in which case the user will need to offer a reminder.

6 Creating a Society of Virtual Patient Instances

A cornerstone for creating a realistic virtual patient training system is providing for wide variation among instances of VPs with a given disease. Our models of each disease include all relevant paths of progression, or "tracks", and each track provides many choice points that can differentiate individual, named cases. A graphical representation of this basic scheme is shown in Figure 2.

The overall model for the disease GERD is represented by the cross shape to the left. The clinically important prototypes that derive from it are shown in the middle with each prototype being represented by a different coloring: the "inflammation only" track has a black upper projection whereas the "inflammation to erosion" track has a progressively shaded lower projection. The named instances of virtual patients are shown to the right, each showing the main feature of its disease (the cross shape) and its track (the coloring of the upper or lower projection) as well as its own personal features, represented by the additional small colored shapes.

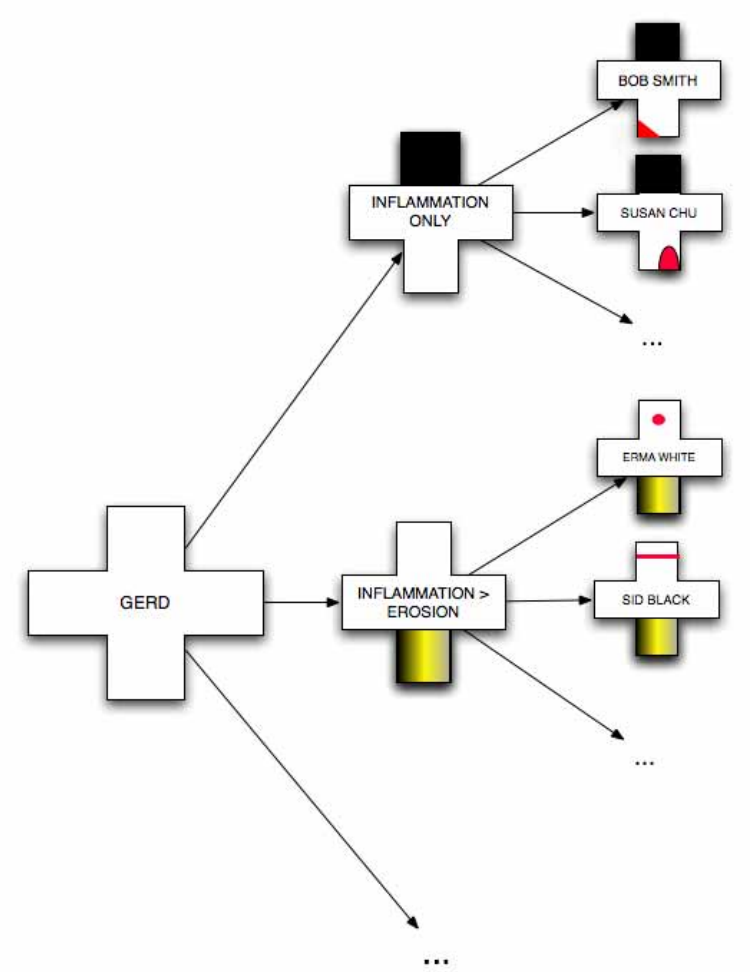

Figure 2. A graphical representation of a basic disease model, its tracks, and actual VP instances that represent those tracks.

A large population of VPs suffering from a given disease can quickly be assembled by teachers, disease specialists and even students and developers by filling out what we call a patient creation template, which is essentially a multiple-choice questionnaire. Each template shows the features central to the model and the various ways they can be initialized and change over time. For example, for GERD the relevant features include lower esophageal sphincter pressure, the state of the lining of the lower esophagus (inflamed, eroded, etc.), the frequency and duration of heartburn, the efficacy of various medications, and so on. The template permits the patient author to select parameter values within the permitted range and according to the permitted combinations. As soon as the values (or defaults) for all relevant properties are chosen and the patient data is saved, the patient is available for use. For details on creating a population of VPs and the ways in which the patient creation template reveals the conceptual model of the disease, see Jarrell et al. (2008).

7 Concluding Thoughts

As mentioned earlier, MVP is an implemented system at the prototype stage of development that currently covers 7 esophageal diseases. All of the functionalities discussed above are incorporated to varying degrees. We are currently working on making the language interaction more robust and will soon move to completing our in-progress model of heart disease.

MVP is a classical AI (artificial intelligence) system in that it strives to model human perception, reasoning and action capabilities and does so on the basis of encoded knowledge. It differs from much of classical AI practice in that it includes people as components in its architecture. If MVP were an expert system in the classical sense, *the system* would have been tasked to diagnose and treat patients rather than the other way around. Indeed, many systems in the medical domain, from Mycin (Buchanan and Shortliffe 1984) on up, had this as their main goal. Another difference from classical AI is the centrality of the descriptive component of the system: the VP's world is certainly not toy (i.e., it does not cover only an extremely small domain). Our emphasis is on acquiring knowledge that is sufficiently deep to support the complex reasoning, simulation and language processing required by the application. This is in contrast to many recent and current approaches – notably, in natural language processing – that stress broad coverage of data in contrast to the utilization of a depth of acquired knowledge.

We believe that the statements often heard nowadays about the demise of AI are ill conceived. The AI enterprise did

not fail. In fact, it has not yet been brought to the test. This is because the enterprise is much more complex than it was perceived to be even by many AI practitioners themselves. Despite the recent emphasis in the field on statistics-oriented methods, they should not be viewed as having superceded classical AI. In fact, these methods have contributed to the core task of knowledge acquisition that is a prerequisite to the success of the program of AI. Progress in learning, knowledge visualization and other ergonomic factors, the ease of access to vast collections of data on the Web and other developments make the original AI goals incrementally more attainable. Our work on the MVP corroborates this state of affairs. We believe that the development of a comprehensive MVP is feasible both scientifically and logistically.

References

1. Beale, S., Lavoie, B., McShane, M., Nirenburg, S., Korelsky, T.: Question answering using Ontological Semantics, Proceedings of the ACL-2004 Workshop on Text Meaning and Interpretation, Barcelona, 2004.
2. Bratman, M.: Faces of Intention, Cambridge University Press, 1999.
3. Buchanan, B., Shortliffe, E.: Rule-Based Expert Systems: The Mycin Experiments of the Stanford Heuristic Programming Project, Addison-Wesley, 1984.
4. Evens, M., Michael, J.: One-on-One Tutoring by Humans and Computers, New Jersey and London, Lawrence Erlbaum and Associates, 2006.
5. Jarrell, B., Nirenburg, S., McShane, M., Fantry, G., Beale, S., Mallott, D., Raczek, J.: An interactive, cognitive simulation of gastroesophageal reflux disease, Stud Health Technol Inform, 194-199, IOS Press, 2007.
6. Jarrell, B., Nirenburg, S., McShane, M., Fantry, G., Beale, S.: Revealing the conceptual substrate of biomedical cognitive models to the wider community, Stud Health Technol Inform, 281-286, IOS Press, 2008.
7. McShane, M., Fantry, G., Beale, S., Nirenburg, S., Jarrell, B.: Disease interaction in cognitive simulations for medical training, Proceedings of the MODSIM World Conference, Medical Track, Virginia Beach, Sept. 11-13, 2007 (2007a).
8. McShane, M., Nirenburg, S., Beale, S., Jarrell, B., Fantry, G.: Knowledge-based modeling and simulation of diseases with highly differentiated clinical manifestations, Proceedings of the 11th Conference on Artificial Intelligence in Medicine (AIME 07), Amsterdam, The Netherlands, July 7-11, 2007 (2007b).
9. McShane, M., Nirenburg, S., Beale, S.: Resolving paraphrases to support modeling language perception in an intelligent agent, Proceedings of the Symposium on Semantics in Systems for Text Processing (STEP 2008), Venice, Italy, 2008 (2008a).
10. McShane, M., Nirenburg, S., Beale, S.: Two kinds of paraphrase in modeling embodied cognitive agents, Proceedings of Biologically Inspired Cognitive Architectures, AAAI 2008 Fall Symposium, Washington, D.C., Nov. 7-9, 2008 (2008b).
11. Nirenburg, S., McShane, M., Beale, S.: A simulated physiological/cognitive "double agent", Proceedings of Naturally Inspired Cognitive Architectures, AAAI 2008 Fall Symposium, Washington, D.C., Nov. 7-9, 2008 (2008a).
12. Nirenburg, S., McShane, M., Beale, S., Jarrell, B.: Adaptivity in a multi-agent clinical simulation system, Proceedings of AKRR'08, International and Interdisciplinary Conference on Adaptive Knowledge Representation and Reasoning, Porvoo, Finland, Sept. 17-19, 2008 (2008b).
13. Nirenburg, S., Oates, T.: Learning by reading by learning to read, Proceedings of the International Conference on Semantic Computing (ICSC-07), Irvine, CA, 2007.
14. Nirenburg, S., Raskin, V.: Ontological Semantics, Cambridge Mass., MIT Press, 2004.
15. Schank, R., Abelson, R.: Scripts, Plans, Goals and Understanding, Hillsdale, NJ, Erlbaum, 1977.