An Interactive, Cognitive Simulation of Gastroesophageal Reflux Disease

Bruce JARRELL\textsuperscript{a}, MD, Sergei NIRENBURG\textsuperscript{b}, PhD, Marjorie MCSHANE\textsuperscript{b}, PhD, George FANTRY\textsuperscript{a}, MD, Stephen BEALE\textsuperscript{b}, PhD, David MALLOTT\textsuperscript{a}, MD, John RACZEK\textsuperscript{a}

\textsuperscript{a} University of Maryland School of Medicine
\textsuperscript{b} University of Maryland Baltimore County

Abstract. The Maryland Virtual Patient (MVP) Project seeks to create realistically functioning virtual humans endowed with automatic physiological and cognitive function that can be used in the training of medical personnel. Physiologically, the state of an MVP changes in response to internal pathophysiologic stimuli and external stimuli, the latter initiated by either the patient or the trainer. Cognitively, the MVP can communicate with trainees about current symptoms, lifestyle, history, adherence to prescribed treatments, etc. We will demonstrate simulation in the MVP environment using the example of patients suffering from gastroesophageal reflux disease (GERD).

Keywords. cognitive simulation, virtual patient, medical education

Introduction

Clinical decision making skills are developed through practice on live patients. We train our physicians using the mix of live patients available at the time of training and trust that the knowledge acquired by managing this cohort of patients will be sufficient to generalize to all patients. Even though this is a tried and true method, it has many drawbacks related to patient safety, lack of objective measures for competence, and the inconsistency of the learning experience with respect to types of diseases, variations in the presentation or course of the disease, and individual patient differences. We need more opportunities to expose our trainees to sufficient patient scenarios in order to foster mastery of the complex knowledge needed daily by practicing physicians.

Computer-based simulation is one way to address the shortcomings of current clinical training practices. For simulation to be effective, it must expose the student to virtual patients that demonstrate sophisticated, realistic behaviors; it must allow open-ended patient investigation by the student (learning through self-discovery); and it must provide each student with a population of patients suffering from a given disease, with each patient displaying clinically relevant variations on the disease theme. Such variations might involve the path or speed of disease progression, the profile and severity of symptoms, responses to treatments, and secondary diseases or disorders that affect treatment choices. If each student could independently manage the care of many such patients – especially in a context in which trial and error learning carried no risk – we hypothesize that the decision making skills of each student would develop faster.
than with traditional training methods alone. In the Maryland Virtual Patient (MVP)\(^1\,\,^2\) project we are developing a simulation and tutoring environment to test this hypothesis.

Before launching full-scale work on this project, exploratory observational exercises were conducted with medical students at the University of Maryland School of Medicine to understand the specifications for effective interaction with a simulated patient [1].\(^3\) In the exercises, the students managed several structured patients in electronic and manual simulations. All the exercises employed patient management problems used routinely in teaching and focused on high-level decision-making, such as the proposal and proof of an inference or the substantiation of an intervention. The most notable observations from this and a follow-up study of simulation for medical training were [1]:

- The simulation must accommodate trial and error patient management with multiple clinically plausible pathways to a solution.
- Changes in patient anatomy and physiology resulting from user action or disease processes over time must result in a consistent appropriate alteration of the state of the patient.
- The representation of time-related patient activities is critical for successful simulation, including allowing the user to “advance the clock” to the next phase of patient management.

In addition to these capabilities, the following are being incorporated into the MVP environment: chronic and acute disorders; simple and complex diseases; knowledge about well-understood and poorly-understood disease processes; knowledge spanning all levels, from gene to organism to population; complications of diseases and treatment modalities; and automatic tutoring. Among the most important conceptual aspects of MVP simulation is automaticity, which refers to the fact that the state of an MVP changes in a realistic way over time and in response to internal (physiological and pathological) and external (clinical and behavioral) stimuli.

**Elicitation and Encoding of Knowledge**

The MVP project centers on an ontology-based model of the physiological and cognitive processes affecting the virtual patient. We encode knowledge about biophysical functions that have clinical relevance in the maintenance of health, the production of disease, and the bidirectional transitions between these two states. When biomechanisms are known, they are modeled using causal chains. Where gaps exist in our knowledge of explicit biomechanisms, they are bridged in various ways – with non-biomechanistic knowledge from the literature, practical clinical knowledge, situational knowledge, observations, probabilistic methods, etc. This integration of implicit and explicit knowledge reflects precisely what a clinician employs when working with a patient. Further, the depth and granularity of this knowledge are determined by the demands of automatic function and realism. Thus, MVPs need not include every mechanism known to biology and clinical medicine.

\(^1\) Patent pending.
\(^2\) This research was supported by Department of Defense grant #17-03-2-001.
\(^3\) The exercises were conducted under IRB Exemption No. BJ-090103 for the project entitled “Computer Simulation as an Aid to Enhancing Medical Education.”
Diseases are modeled as changes in key property values over time. For each disease, a set number of conceptual stages is established, and typical values (or ranges of values) for each property are associated with each stage. Values at the start or end of each stage are recorded explicitly, with values between stages being interpolated. The disease model includes a combination of fixed and variable features. For example, although the number of stages for a given disease is fixed, the duration of each stage is variable; similarly, although the values for some physiological properties undergo fixed changes across patients, the values for other physiological properties are variable within a specified range. Therefore, on the one hand, each disease model is sufficiently constrained so that MVPs suffering from the disease must show appropriate physiological manifestations of it, while on the other hand, each disease model is sufficiently flexible to permit instances of MVPs to differ in clinically relevant ways, as selected by the author of each MVP instance.

Once an approach to modeling a given disease has been devised and all requisite details have been elicited, the disease-related events and their participants are encoded in ontologically-grounded scripts written in the metalanguage employed in the OntoSem environment. Scripts represent typical sequences of events and their causal and temporal relationships. In other words, they encode how individual events hold well-defined places in routine, typical sequences of events that happen in the world, with a well-specified set of objects filling different roles throughout that sequence. For example, if the event is swallowing, there is only one animate participant (the swallower), but many other objects play necessary roles: various nerves and muscles act as instruments of peristalsis; the swallowed bolus is the theme of peristalsis-driven motion events; the stomach is the final destination of the bolus, and so on. Scripts normally contain subscripted and can be more or less fine-grained depending on the goals of the given simulation. Within the MVP project we have developed both domain scripts and workflow scripts. Domain scripts describe basic physiology, disease progression and responses to treatments, whereas workflow scripts model the way an expert physician would handle a case, thus forming the knowledge substrate for automatic tutoring.

The OntoSem ontology differs from others not only in its inclusion of scripts, but also in its rich inventory of properties, both attributes and relations (most other ontologies, e.g., UMLS [3], are actually hierarchical word nets rather than knowledge-rich ontologies). As we expand the OntoSem general-purpose ontology into the medical domain, we are incorporating, where possible, the terminology used by the Foundational Model of Anatomy [4].

**Reasoning with Knowledge**

MVPs are modeled as “double agents” with both physiological and cognitive functions. Physiologically, the state of an MVP changes in response to internal pathophysiological stimuli and external stimuli, the latter initiated either by the patient or the trainee. Cognitively, the MVP can communicate with trainees about current symptoms, lifestyle, history, adherence to prescribed treatments, etc. Structured knowledge in the disease model acts as input to the simulation engine. The simulation

---

4 OntoSem is the implementation of the theory of Ontological Semantics, a theory originally developed for knowledge-rich text processing (Nirenburg and Rakin 2004).
can run in clinical mode, where patient symptoms and physiology are known only through questions and diagnostic tests, or in omniscient mode, in which all patient properties can be monitored throughout the simulation. As an example of automaticity in response to external interventions, students are permitted to prescribe any treatment available in the system at any time, with the MVP responding accordingly. If, for example, the student launches an inappropriate treatment, the MVP's state may or may not change, but certainly will not produce the intended result. Upon recognizing this undesirable result, the student can attempt to recover from the mistake, for example by withdrawing the treatment or introducing a different one. The effect of recovery attempts are interpreted relative to encoded knowledge and the current state of the MVP. The system does not exhaustively list all permutations of paths a trainee could take and all consequential responses of the MVP; instead, it relies on ontologically-grounded descriptions of basic physiology, disease processes, effects of treatments, and so on, so that the state of a given MVP at a given time will, quite literally, fall out of the underlying model.

Authoring Instances of MVPs

A cornerstone in creating a realistic MVP environment is providing for wide variation among instances of MVPs with a given disease. That is, the basic model of a disease includes all relevant tracks (i.e., paths of progression), and each track provides many choice points that differentiate cases. Among the many tasks carried out by the author of a disease model is the selection of properties to be tracked, their ranges of values, and the defaults for those values. Such a disease model is then concretized into a given patient instance by an instance author, who is typically a physician-teacher or disease specialist. The instance author determines the MVP's basic physiological properties, relevant lifestyle factors, the rate of progression of the disease, which path the disease takes at all possible furcations, the specific symptom profile at given times, and so on. This process has been reduced to an electronic multiple-choice questionnaire that takes a little time to complete. The simplicity of authoring patient instances derives from the care taken to create the basic model of the disease, including delineating exactly which property values are available for individual parameterization and which ones are fixed for all patients experiencing the given stage of a disease. As soon as the values (or defaults) for all relevant properties are chosen, the patient instance is available for use.

Results

Knowledge Elicitation. The MVP project places significant demands on authors of disease models to render complex, multi-scale functions in a form that can be implemented computationally. The knowledge elicitation process is a collaboration between the model author and a knowledge engineer, who mediates between the physician and the programmer. Physicians must distill their extensive and tightly coupled physiological and clinical knowledge into the most relevant subset, and express it in the most concrete of terms. Not infrequently, they are also called upon to hypothesize about the unknowable, like the state of a patient experiencing a pre-clinical stage of disease, or the state of a patient after an effective treatment that is never, in real life, followed up by objective tests. Such hypotheses reflect the mental models of given
experts, which might differ in subtle ways from those of other experts. However, such differences, we would suggest, have little bearing on the ultimate goal of this enterprise: to create MVPs whose behavior is sufficiently life-like to further specific teaching goals.

**Disease Simulation.** We chose to initially model esophageal disease because the esophagus is a relatively uncomplicated organ and because one of the symptoms of esophageal disease, chest pain, can cause significant diagnostic dilemmas with cardiac disease. Knowledge about the normal and abnormal anatomy and physiology of the esophagus was elicited from model authors and recorded. The two common mechanisms for gastroesophageal reflux disease (GERD) — a decreased Lower Esophageal Sphincter Pressure (LESP) and Transient LES Relaxation (TLESR) — and all relevant clinical forms of GERD were modeled. The latter included: non-erosive GERD; GERD with erosive esophagitis, stricture, Barrett’s metaplasia and adenocarcinoma; and proximal GERD. In addition, diseases potentially associated with GERD, including scleroderma, Zenker’s Diverticulum and achalasia, were modeled. For achalasia and scleroderma, which are diseases with a poorly understood esophageal pathophysiology, property values of the MVP change as a function of passing time, since the disease natural history can only be clinically observed rather than explained using causal chains. By contrast, for GERD, which has a well understood pathophysiology, the disease model is driven by causal chains that reflect current biomedical thinking.

Causal chain modeling is a particularly potent strategy that allows expanded opportunities for automatic function in virtual patients:

- A new disease can be generated as a side-effect of another disease: for example GERD is automatically initiated in any patient whose LESP drops below 10, which can occur due to scleroderma or after a successful surgical intervention for achalasia.
- The rate of progression of a disease can be automatically determined: for example, a patient with an LESP of 0 (after a successful Heller myotomy) will have a faster progression of GERD than a patient with an LESP of 9 (after a successful pneumatic dilation).
- The effects of interventions can be automatically determined: for example, whether GERD is progressing or healing is determined by the daily total time in acid reflux (TTAR). TTAR is determined by total time in reflux — which can be altered in some patients by changes in lifestyle, and by the acidity of the refluxed substance — which can be affected medication.

Upon testing, the MVP has functioned accurately, including reasoning effectively when responding to both expected and unexpected user actions.

**Discussion**

We have designed a simulation system that has demonstrated complex, automatic behavior. Thus far, it has a limited repertoire of esophageal physiology, pathophysiology, and clinical management for common esophageal diseases. In spite of the limited repertoire, we believe that MVPs represent a conceptual leap in the computer modeling of humans in the continuum of health and disease. MVPs display realistic function in simulations where they can be observed, interacted with, and
treated by students. Variability of selected parameters permits a wide variety of instances of virtual patients to be created from the same ontologically-grounded disease model. We have honed our approaches to knowledge elicitation, script writing and incorporating scripts into the simulation engine. We are now positioned to test our hypothesis that trial and error management of MVPs can teach students the basics of clinical medicine as well as, or better than, bedside teaching or small group teaching. In fact, there is evidence that learning by working through computer-based scenarios can be very effective: for example, 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 [5].

Two components are necessary for a trainee to learn clinical medicine: an inventory of patients showing clinically relevant variations of disease, and a tutor to guide the student (as necessary) and to validate that his or her success derives from accurate and sufficient knowledge. We have recently implemented the first version of the tutor for esophageal diseases.

Our work on tutoring has been informed by results from the CIRCSIM group, which has been pursuing automatic tutoring strategies for the diagnostics and treatment of the baroreceptor reflex. The current CIRCSIM-Tutor evolved from a system that offered students a dynamic mathematical model with no tutoring support into a system that offers tutoring without the dynamic mathematical model (results of certain scenarios were stored and are deemed sufficient for the given educational goals). The contrast with MVP is clear: for us, the autonomous functioning of MVPs is, and will remain, central, with tutoring being interpreted as a useful option alongside trial-and-error learning.

We are also currently working on incorporating natural language interaction into the system, with the current mode of interaction being menu-driven (that is, the trainee is presented with inventories of questions, diagnostic tests, treatments, hypotheses and diagnoses to choose from). The desire to incorporate natural language interaction into tutoring systems has been expressed by developers of many tutoring systems. Unlike others, however, our group has been working on knowledge-based natural language processing (NLP) for some twenty years. In fact, the OntoSem ontology, knowledge representation language, and many of the processors that are serving as a substrate for the MVP system were all originally developed for NLP applications. Therefore, we have confidence in our ability to incorporate natural language support into the MVP environment in the near term.

References