

# Language Understanding in Maryland Virtual Patient

**Sergei Nirenburg**

**Stephen Beale**

**Marjorie McShane**

University of Maryland Baltimore County

{sergei, sbeale,  
marge}@umbc.edu

**Bruce Jarrell**

**George Fantry**

University of Maryland School of Medicine

BJarrell@som.umaryland.edu

GFantry@medicine.umaryland.edu

## Abstract

This paper discusses language understanding in the Maryland Virtual Patient environment. Language understanding is just one of many cognitive functions of the virtual patients in MVP, others including decision making about healthcare and lifestyle, and the experiencing and remembering of interoceptive events.

## 1 Introduction

Maryland Virtual Patient<sup>2</sup> (MVP) is an agent-oriented environment for automating certain facets of medical training. The environment contains a network of human and software agents, at whose core is a virtual patient – a knowledge-based model of a person with a disease. This model is implemented in a computer simulation. The virtual patient is a “double agent” that displays both physiological and cognitive function. Physiologically, it undergoes both normal and pathological processes in response to internal and external stimuli. Cognitively, it experiences symptoms, has lifestyle preferences, has memory (many of whose details fade with time), and communicates with the human user about its personal history and symptoms. Other software agents in the MVP environment include consulting physicians, lab technicians and a virtual mentor (tutor).

What makes virtual patient modeling feasible – considering that comprehensively modeling human physiology would be a boundless endeavor – is our task-oriented approach: we are

not trying to recreate the human organism in all its details, we are modeling it to the extent necessary to support its realistic autonomous functioning in applications aimed at training the diagnostic and treatment skills of medical personnel.

Trainees can use MVP to interview a virtual patient; 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; 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); and, if desired, receive mentoring from the automatic mentor.

The virtual patient (VP) simulation is grounded in an ontologically-defined model of human anatomy and physiology. Instances of virtual patients with particular diseases and particular physiological peculiarities are generated from core ontological knowledge about human physiology and anatomy by grafting a disease process onto a generic instance of a human. Disease processes themselves are described as complex events in the underlying ontology.

## 2 Reasoning by the Cognitive Agent

The cognitive side of the VP carries out reasoning in response to two types of input: interoception (the experiencing of physical stimuli, like symptoms) and language input. Specifically its functioning includes:

1. experiencing, interpreting and remembering symptoms
2. deciding to go see a doctor, initially and during treatment
3. understanding the doctor’s language input as well as its intent

---

© 2008. Licensed under the *Creative Commons Attribution-Noncommercial-Share Alike 3.0 Unported* license (<http://creativecommons.org/licenses/by-nc-sa/3.0/>). Some rights reserved.

<sup>2</sup> Patent pending.

4. deciding whether to ask knowledge-seeking questions about a test or intervention suggested by the doctor
5. deciding whether to agree to a test or intervention suggested by the doctor.
6. deciding on what specifically to say in response to the doctor's questions, recommendations, etc.

In this paper we concentrate on point 3. We point readers to other works about MVP (e.g., McShane et al. 2007) for a discussion of other aspects of MVP.

Five types of subdialogs are supported in MVP.

1. Requests for information and responses. These include (a) the physician asking the patient questions about symptoms and lifestyle, and (b) the patient asking questions about features of suggested interventions as well as other options.
2. Requests for action and responses – primarily the physician suggesting that the patient agree to have an intervention.
3. Domain descriptions provided by the user, the key points of which must be understood and remembered (“learned”) by the VP.
4. Scheduling follow-up appointments.
5. General dialog topics, like greetings, expressions of gratitude and other means for making the dialog more realistic in the user's eyes.

Our approach to treating dialog is unlike most other approaches in that all language-oriented reasoning is carried out on the basis of formal interpretations of text meaning. We call these interpretations *text meaning representations* or **TMRs**. Note that TMRs are written using the same ontologically grounded metalanguage as is used to represent interoception. In short, all knowledge and reasoning in our environment employs the same metalanguage, so whether a patient experiences new symptoms or learns information about its disease from the user, the new information will be stored the same way in the patient'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 produced. 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 even semantic (*Does the food get stuck when you swallow?* ~ *Do you have difficulty swallowing?*). Third, TMRs facilitate the detection of which aspects of meaning are central and which are of secondary importance. For example, the analyzer can determine which portions of input utterances merely convey politeness. To take an extreme example for illustration, the question “Do you have difficulty swallowing?” could be rendered by an overly polite physician as: “If you don't mind, I would really appreciate it if you would tell me whether you have any difficulty swallowing.”

When the VP receives language input, it uses its lexicon, ontology and a reasoning-enabled analyzer to create a TMR corresponding to the input. Next, it determines the intent of that input – e.g., through the recognition of indirect speech acts. After that it plans its response then generates its response. Here we talk about the first two stages of text processing: understanding the dialog turn and understanding its intent.

### 3 Understanding a Dialog Turn

The input to understanding a dialog turn is text input by the user. Background knowledge that must be leveraged is the knowledge stored in the lexicon, ontology and the patient's long-term memory of assertions, also called its fact repository. The output is a TMR. TMR production actually comprises two stages: the first stage, production of the *basic TMR*, involves disambiguation and the determination of semantic dependencies; the second stage, production of the *extended TMR*, adds the results of procedural semantic routines, like the resolution of reference.

For example, the following questions are all synonyms at the level of extended TMR, at least at the grain-size of description needed for our current application: *Have you been coughing?* *Do you find yourself coughing?* *Do you experience any coughing?* *Do you ever experience coughing?* *Do you have a cough?* *Any coughing?* *Coughing?* etc. All of these questions ask whether or not the patient has the symptom ontologically described as the event called COUGH. The extended TMR for this set of questions is:

**(REQUEST-INFO-1**  
 (THEME MODALITY-1.VALUE))  
**(MODALITY-1**  
 (TYPE EPISTEMIC)  
 (SCOPE ASPECT-1))

**(ASPECT-1**  
 (ITERATION MULTIPLE)  
 (SCOPE COUGH-1))  
**(COUGH-1**  
 (EXPERIENCER HUMAN-1)  
 (TIME  
 (FIND-INTERVAL (FIND-ANCHOR-TIME)  
 (FIND-INTERVAL-LENGTH BEFORE)))

This TMR is read as follows. The input creates an instance of REQUEST-INFO. The instance is numbered, like all TMR instances, to distinguish it from other instances of that concept. The THEME of REQUEST-INFO-1 – i.e., what is being asked about – is whether or not COUGH-1 has occurred repetitively; this is shown in the ASPECT-1 frame. The COUGH event itself has the VP, HUMAN-1, as the EXPERIENCER. The time of the COUGH event is calculated using a procedural semantic routine that seeks a certain time interval in the past (we leave out details of *which* period of time in order to avoid a lengthy tangent). Although this example is a bit complex – involving both aspect and modality – it provides some insight into the format and content of TMRs in our environment.

The text analyzer can automatically create this same TMR for all of the different inputs in large part thanks to the lexicon. Syntactic knowledge in lexicon entries in OntoSem is formulated using an extended form of Lexical Functional Grammar, with variables used to link entities in the syntactic structure (syn-struct) zone of an entry with those in the semantic structure (sem-struct) zone. Lexicon entries can also contain calls to procedural semantic routines (meaning-procedures). The caret means “the meaning of” a given variable. \$var0 is the head entry.

*Have you been coughing?* is a syntactic transformation of *Do you cough?*, which is understood directly by the analyzer as a question about *cough* (v.), which is mapped to the concept COUGH in the respective lexicon entry.

(cough-v1  
**(syn-struct**  
 ((subject ((root \$var1) (cat n)))  
 (root \$var0) (cat v)))  
**(sem-struct**  
 (COUGH (EXPERIENCER (value ^\$var1))))))

For the other paraphrases, “superfluous” words must be attributed null semantics. For example, *to find oneself verb-ing* is semantically same as *to verb*, the only real difference being stylistic. There is a lexical sense of *find* that attributes null

semantics to *find oneself* in the collocation *find oneself doing X*.

Examples in which question processing is folded into the lexicon entry are *Any + EVENT?* (*Any coughing?*) and *EVENT? (Coughing?)*. The lexicon entry that covers these is keyed on the question mark, since it is the only element that is always available in these turns of phrase (since “any” is optional). The sem-struct is headed by the concept REQUEST-INFO, whose THEME is the value of epistemic modality scoping over the event in question.

This brief overview is intended only to give a taste of the process of language understanding by virtual patients in MVP. This process is exactly the same as language understanding in other applications of our text processor, called OntoSem (see Nirenburg and Raskin 2004).

The eventualities of text understanding by the cognitive agent of the VP are: (a) successful understanding, (b) the VP’s belief that it understood, only to be corrected by the user, or (c) the failure of understanding, in which case the VP asks for clarification by the user.

#### 4 Understanding the Intent of a Dialog Turn

The extended TMR is our most complete model of the meaning of an utterance, but it does not include what is called indirect speech act processing – i.e., understanding intentions of the speaker when they are not overtly mentioned in the utterance. Well-known examples of the dichotomy between expressed meaning and intended meaning include *It’s cold in here* (which might be a statement/complaint or might be an indirect request for the interlocutor to do something about it, like close the window) and *Can you pass the salt?* (which might be a question about physical ability or an indirect request).

Our work on indirect speech acts includes long-term, fundamental theory building as well as short-term, immediately implementable solutions. At a fundamental level, speech act processing requires the speaker and the interlocutor to keep a full inventory of their beliefs about the other’s knowledge, their understanding of their own and the other’s plans and goals, both long-term and immediate, their understanding of what is and what is not within each person’s or agent’s power to do, and so on. More immediately, we have implemented a means of detecting indirect speech acts in the dialogs between VPs and us-

ers. Our approach, like all of our approaches to automatic reasoning, is grounded in TMRs.

There are three utterance types that the VP expects of the user, which correspond to three user plans: asking questions to learn information that will aid in diagnosis and treatment, explaining things to educate the VP, and giving advice to the VP about what it should do. At any point in the dialog when the user stops typing and expects a response from the VP, the VP must decide which of the plans the user is pursuing. Surface-level heuristics are not always definitive: e.g., *Would you agree to have a Heller myotomy?* is both a question and advice, and *I think that having a Heller myotomy is the best option* is both information and advice.

We prepare the VP to interpret indirect speech acts by creating TMR correspondences between the direct and the indirect meaning of certain types of utterances. Let us take as an example the doctor's offering advice on what to do. There are many ways the doctor can present advice, including the following, provided below with their respective TMRs. In all of these TMRs, HUMAN-1 is the doctor and HUMAN-2 is the patient (these TMRs are simplified for purposes of exposition; also note that all reference resolution has been carried out). INTERVENTION stands for any event that is ontologically an intervention – that is, a test or a medical procedure. Note that the lexicon directly supports the automatic generation of these TMRs.

### 1. I (would) advise/suggest/recommend (having) INTERVENTION

(ADVISE-1  
(THEME INTERVENTION-1)  
(AGENT HUMAN-1)  
(INTERVENTION-1  
(EXPERIENCER HUMAN-2))

### 2. I think you should have INTERVENTION

(MODALITY-1  
(TYPE BELIEF)  
(VALUE (> .7))  
(SCOPE MODALITY-2)  
(ATTRIBUTED-TO HUMAN-1))  
(MODALITY-2  
(TYPE OBLIGATIVE)  
(VALUE .8)  
(SCOPE INTERVENTION-1)  
(ATTRIBUTED-TO HUMAN-1))  
(INTERVENTION-1  
(EXPERIENCER HUMAN-2)))

### 3. I'd like to schedule you for <set you up for, set you up to have> INTERVENTION

(MODALITY-1  
(TYPE VOLITIVE)  
(SCOPE EVENT-1)  
(VALUE .8)  
(ATTRIBUTED-TO HUMAN-1))  
(SCHEDULE-EVENT-1  
(AGENT HUMAN-1)  
(THEME INTERVENTION-1)  
(BENEFICIARY HUMAN-2))  
(INTERVENTION-1  
(EXPERIENCER HUMAN-2))

The “core” meaning that the VP must glean from any of these TMRs is the meaning shown in (1): that the doctor is advising that the patient have the intervention. The correlations between the TMRs in (2) and (3) and this core TMR are established using a TMR-to-TMR translation function. The efficacy of this translation process depends on (a) preparing for the full inventory of possible *types* of input TMRs that correspond to the given meaning, and (b) being able to extract from more complex TMRs these basic kernels of meaning. We have already implemented part (a) in our current system. Part (b) requires more long-term effort, the problem essentially being that one needs to teach the system to zero in on what is important and ignore what is unimportant. For example, negation is very important: *I advise you to have INTERVENTION* is very different from *I do not advise you to have INTERVENTION*. However, *I think I would choose to advise you to have INTERVENTION* includes aspects of meaning (‘think’, ‘would choose’) that are really not important and should be simplified to the main meaning of the proposition. We consider research on this aspect of agent reasoning to be a long-term endeavor

### References

- McShane, Marjorie, Sergei Nirenburg, Stephen Beale, Bruce Jarrell and George Fantry. 2007. Knowledge-based modeling and simulation of diseases with highly differentiated clinical manifestations. *11th Conference on Artificial Intelligence in Medicine (AIME 07)*, Amsterdam, The Netherlands, July 7-11, 2007.
- Nirenburg, Sergei and Victor Raskin. 2004. *Ontological Semantics*. MIT Press.