A Cognitive Architecture for Simulating Bodies and Minds

Marjorie McShane, Sergei Nirenburg, Stephen Beale
Department of Computer Science and Electrical Engineering, University of Maryland Baltimore County, Baltimore, MD, 21136, USA
{marge,sergei,sbeale}@cs.umbc.edu

Roberta Catizone
Onyx Consulting, Inc.
roberta@dcs.shef.ac.uk

Keywords: cognitive architecture, artificial agents, simulation, clinical advisor

Abstract: This paper presents an overview of a cognitive architecture that supports the creation and deployment of intelligent agents capable of simulating human-like abilities. The agents, which have a simulated mind and, if applicable, a simulated body as well, are intended to operate as members of multi-agent teams featuring both artificial and human agents. The agent architecture and its underlying knowledge resources and processors are being developed in a sufficiently generic way to support a variety of applications. In this paper we briefly describe the architecture and two applications being configured within it: the Maryland Virtual Patient (MVP) system for training medical personnel and the CLinician's ADvisor (CLAD). We organize the discussion around four aspects of agent modelling and how they are utilized in the two applications: physiological simulation, modelling an agent's knowledge and learning, decision-making and language processing.

1 INTRODUCTION

In this paper we present a cognitive architecture that permits us to build agents that emulate a number of high-level, human-like capabilities that support a variety of useful applications. The core capabilities of the agents include the following: they are designed to operate in a hybrid network of human and artificial agents; they emulate human information processing capabilities by modeling conscious perception and action, which includes reasoning and decision making; they can communicate with people using natural language; they can incorporate a physiological model, making them what we call "double agents" with simulated bodies as well as simulated minds; they can be endowed with personality traits, preferences and psychological states that affect their perceived or subconscious preferences in decision-making; their means of perception include language understanding and interoception, which is the experiencing of sensations from one's body; their underlying principles, knowledge resources and processors are broad-coverage rather than geared at a particular application, which makes them – after a modicum of inevitable knowledge base refinement – portable to a variety of domains and application configurations; and the perception and action algorithms used by the agents are supported by and, in turn, augment an agent's memory of event, state and object instances to complement its ontological knowledge of event, state and object types. What makes modeling such multi-faceted agents feasible is that all aspects of agent functioning, from physiological simulation to decision-making to communicating in natural language to learning new information, are supported by the same knowledge substrate encoded in a single metalanguage.

As a testbed for the development of our extended cognitive architecture ("extended" because cognitive architectures typically do not include physiological simulation), we have been modeling agents in the medical domain. Our first application, Maryland Virtual Patient (MVP; Jarrell et al. 2008, McShane et al. 2007a,b, 2008, Nirenburg et al. 2008), is an environment for training medical personnel in patient management – diagnostics, treatment and care over time. Our second application, built on the same knowledge and processing substrate, is a CLinician's ADdvisor (CLAD), which is intended to assist practicing clinicians by reducing their cognitive load. This paper focuses on the common knowledge needs and functionalities for agents in both applications, and provides evidence that taking a knowledge-based approach to building agent-based systems has significant payoffs, both within a given application and in porting to other applications.

The paper is organized as follows. After a brief overview of MVP and CLAD, we turn to the main discussion – a comparison of how four specific aspects of agent modeling are realized in our two applications, and how the R&D results from the first application are being reused in the second. We conclude the paper with comparisons with other approaches and systems.

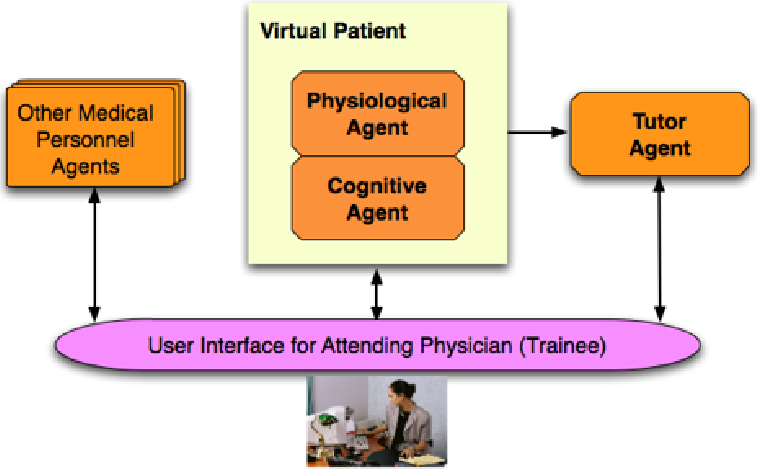

Virtual Patient
Physiological Agent
Cognitive Agent
Other Medical Personnel Agents
Tutor Agent
User Interface for Attending Physician (Trainee)

Figure 1. The network of agents in MVP.

2 A BRIEF OVERVIEW OF MVP AND CLAD

Maryland Virtual Patient (MVP) is a cognitive simulation and training system whose goal is to provide medical practitioners with the opportunity to develop clinical decision-making skills by managing many highly differentiated virtual patients (VPs). MVP is modeled as a network of human and artificial agents, as shown in Figure 1.

The human agent, who is typically a physician seeking to improve his cognitive decision making skills, plays the role of the attending physician. 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. 2008). 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.

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; 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 by treating the new condition he has inadvertently caused.

A prototype MVP system has been implemented, and the system continues to be developed.

CLAD, a CLinician's ADvisor, is intended to decrease the cognitive load of clinicians caused by the vast amount of information available, and to improve overall patient outcome through providing high-value decision-making assistance. CLAD will assist clinicians by providing advice (along with its justification), answering questions, providing prognoses, carrying out administrative tasks (e.g., finding out if a given procedure is covered by the patient's insurance company), and so on. As shown in Figure 2, CLAD can receive new information in two ways, through language recorded in the chart and by listening in to the conversation between the doctor and the patient. Currently, the first of these is implemented and the second is under development.

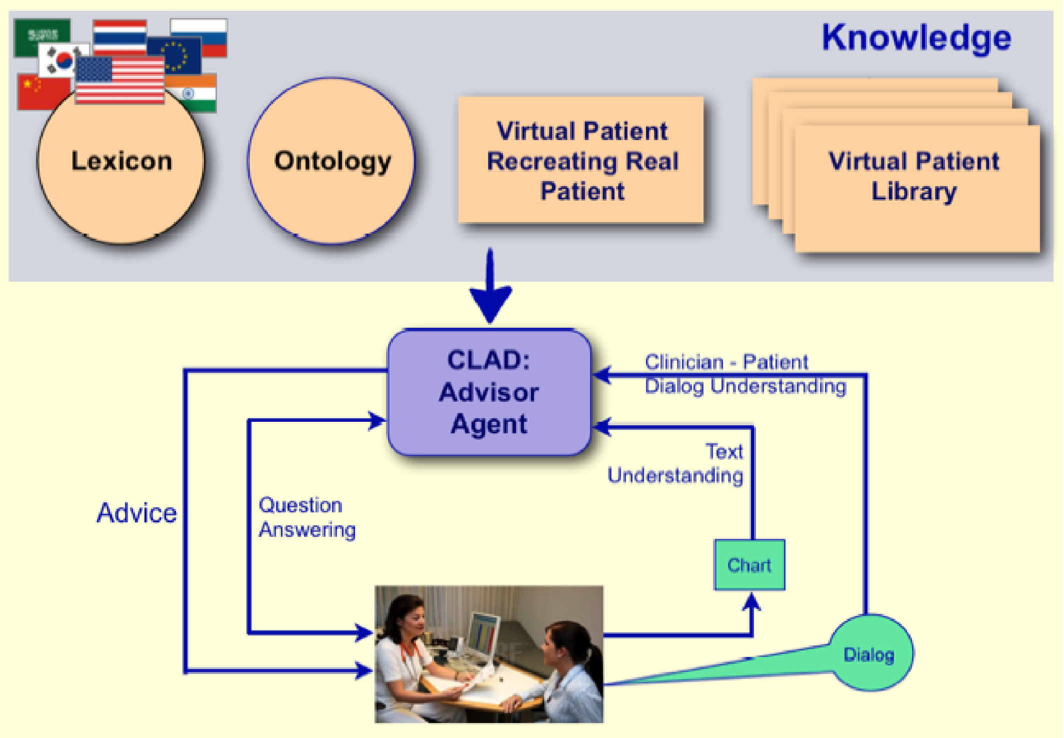


Figure 2. The architecture of CLAD.

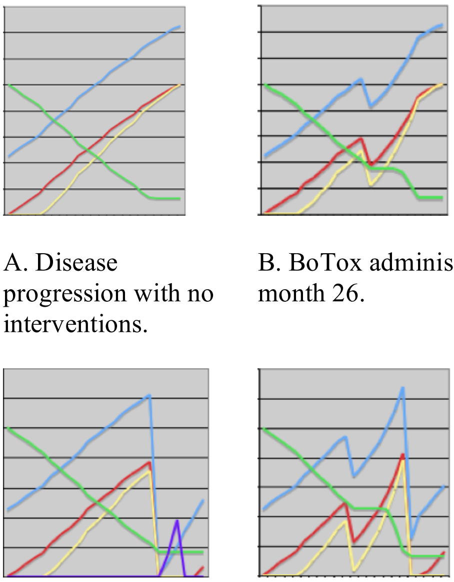
3 Four Aspects of Agent Modeling

All of the modeling in the MVP/CLAD environment is cognitive modeling in the sense that we have no manikins or robots – all agents are virtual. However, our cognitive modeling covers not only the kinds of mental processes carried out by people, it also covers modeling and simulating human anatomy and physiology in the way that people understand it to function. We begin by describing our approach to physiological simulation and its use in both the MVP and CLAD applications, then move to the more traditional areas of cognitive modeling and their realizations in both applications.

3.1 Physiological Simulation

The physiological simulation of virtual patients operates on the basis of ontological models of human physiological processes during states of health and disease (McShane et al. 2007a,b). When possible, these models employ causal chains since this permits realistic automatic function even in unexpected situations. When physiological causal chains are not known to medical science, the simulation is driven by a temporal ordering of events distilled from clinical literature: each disease is divided into conceptual stages, with each stage being associated with clinically observed physiological changes and symptom profiles. As simulated time passes, the patient's state changes incrementally using interpolation functions whose inputs are feature values at set points in the simulation.

Many aspects of each disease model can be parameterized, thus yielding a large, differentiated population of virtual patients suffering from that disease (Jarrell et al. 2008). In clinical terms, one of the important ways VPs differ among themselves is in their responses to interventions. Figure 3 shows just how dynamic the MVP application is (i.e., how it *is* not and *cannot be* pre-scripted) by graphically illustrating the variability in the disease course of a particular virtual patient based on the dynamic intervention choices of the user.



A. Disease progression with no interventions.

B. BoTox administered in month 26.

C. Heller myotomy carried out in month 34.

D. BoTox administered in month 22 and pneumatic dilation carried out in month 36.

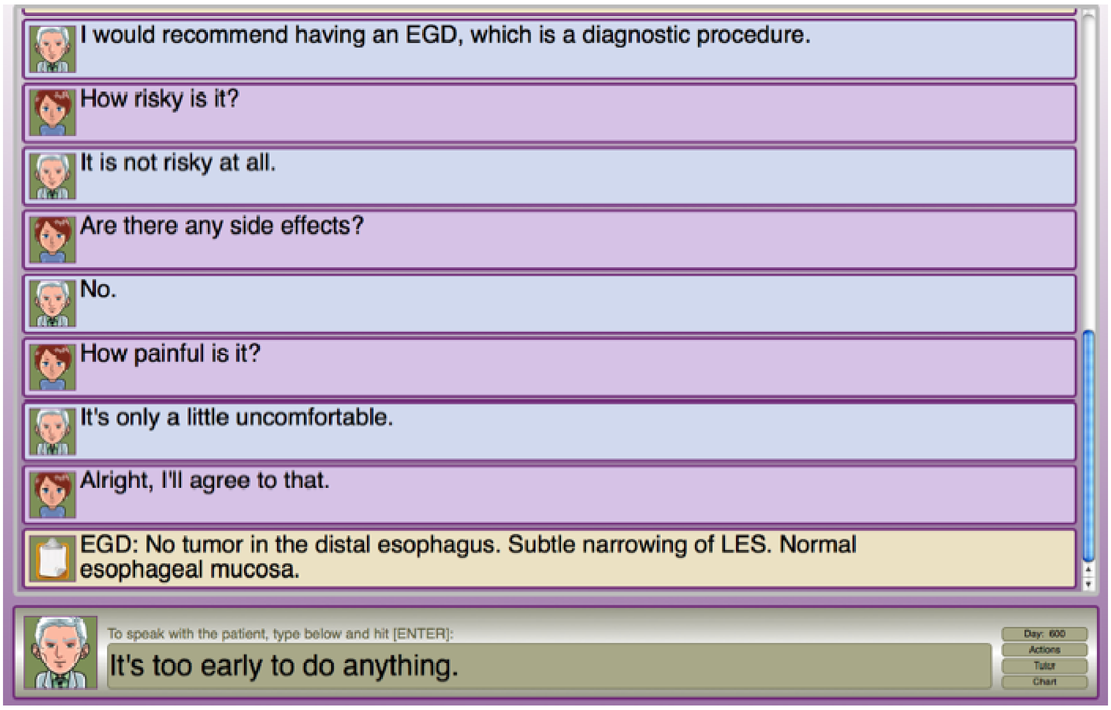
Figure 3. The same virtual patient with different intervention strategies over time.

This VP has the disease achalasia, which is a disease that raises the basal ("tight") and residual ("relaxed") pressure of the lower esophageal sphincter (LES), making it increasingly difficult for food to pass to the stomach. Figure 3 shows different changes over time of 5 of the VP's property values based on different interventions by the user at different times. We describe them spatially due to expected black-and-white printing: amplitude of peristalsis contractions (the line going from upper left to lower right); basal lower esophageal sphincter pressure (the top line going from lower left to upper right); residual lower esophageal pressure (the middle line, same orientation); difficulty swallowing (the bottom line, same orientation); and heartburn (the spike in the lower right corner of scenario C). The interventions and their times are indicated in the caption for each figure. The vertical axis represents a normalized, abstract scale and the horizontal axis represents time.

These four scenarios are but a handful of the thousands of scenarios one could create using this patient instance, since any of the treatments could be administered in any combination at any time. In addition, the system does not rule out the possibility of the user ordering *incorrect* treatments that could be harmless or could worsen the patient's condition, creating a more complicated case that the user must now manage. Moreover, hundreds of other patient instances could be authored from the basic ontological model of a disease, making the scope of cases truly wide and differentiated.

The importance of physiological simulation for MVP should be clear: one needs a simulated virtual patient to provide the patient management challenges MVP seeks to provide – open-ended simulations over time that permit trial-and-error learning in an environment that includes many salient features of real clinical practice. But the utility of physiological simulation in our environment does not end there: the advisor in CLAD (and the tutor in MVP, about which we will say little here for reasons of space) can use physiological simulation as means of *knowledge gathering* to contribute to developing *prognoses* that can affect *decision-making* and *advice-giving*. This works as follows. The MVP/CLAD environment already has ontological models that provide for the real-world scope of disease progressions and patient features. At any given time, some features of a live patient and his disease will be known and others will not. The features that *are* known can be inserted into the disease model, thus "personalizing" it – i.e., making it progressively less a generic, population-scale model and more the model of a particular patient's condition. One or more simulations can then be run using the known patient-related features and some less precise population-scale constraints over the remaining feature values. Of course, the more features that are known, the more constrained the number of disease paths and outcomes; but even with few features known, some partially personalized reckoning about what might happen to a patient in a given time frame will be possible. Let us put this into a typical context.

Assume that the best advice CLAD can give at a certain point in time is that the patient have Procedure X, but Procedure X has a recovery time of 2 weeks and the patient cannot afford that amount of time off work until his summer vacation in 3 months. He wants to know how bad things will get if he waits that long. CLAD can run simulations and present the clinician with the likely states of disease, including symptom profiles, in 3 months' time. It can either (a) present this material in a presentation format to the clinician such that the clinician interprets it and formulates a recommendation for the patient or (b) directly interpret the material, formulating its own advice about whether or not the patient can wait, with justifications for that advice. Currently, CLAD offers forecasts in tabular form but soon will be able to generate outputs like, “Waiting will not be a good idea because it is expected that the patient will have lost between 15 and 20 lbs. by then.”



I would recommend having an EGD, which is a diagnostic procedure.

How risky is it?

It is not risky at all.

Are there any side effects?

No.

How painful is it?

It's only a little uncomfortable.

Alright, I'll agree to that.

EGD: No tumor in the distal esophagus. Subtle narrowing of LES. Normal esophageal mucosa.

To speak with the patient, type below and hit [ENTER]:

It's too early to do anything.

Figure 4. The main MVP interface.

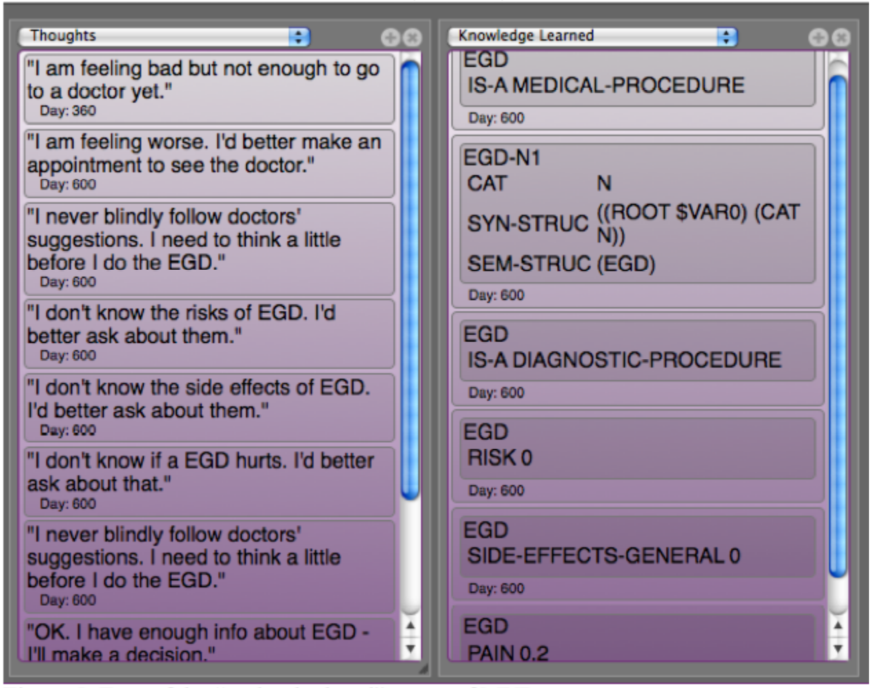
3.2 Modeling an Agent's Knowledge and Learning

Agents in the MVP/CLAD environment use three main knowledge bases: (1) a language-independent **ontology**, which contains knowledge about types of objects and events in the world and the properties that link them; it also contains the scripts that support simulation; (2) a language-independent **fact repository**, which contains remembered instances of objects and events and the properties that link them (this is part of “CLAD: Advisor Agent” in Figure 2); and (3) a **lexicon** of English, the meaning of whose tokens is described with reference to ontological concepts. Each agent has its own set of knowledge bases, reflecting its own particular state of knowledge about types and tokens in the world, memory of experiences, knowledge of its own preferences, etc. The knowledge bases of all agents overlap to a large degree, representing what we think of as “general human knowledge” and a normal level of proficiency in English.

For the medical domain, we have an expert ontology and lexicon that form the basis of (a) the physiological simulations, (b) the instruction of the tutor in MVP, and (c) the advice-giving of the advisor in CLAD. These are updated regularly to keep abreast of the latest findings in medicine, clinical guidelines, etc. By contrast, the ontology and lexicon of other virtual agents – like VPs – differ greatly with respect to their knowledge of specialized domains, like medicine. For example, one VP might know a lot about its disease and ask few questions or ask only very technical questions, while another VP might know nothing of its disease and want the doctor to teach him all about it. As for the fact repository, even medical experts cannot share a single fact repository because the fact repository stores not only long-term knowledge of tokens (e.g., the results of the biggest study of some medication's side-effects) but also memories of interactions with other agents and whatever the agent itself does in its simulated life.

Clearly, in order for an agent to populate and use its knowledge bases as a person would, it must be able to learn. Agents learn in many ways in the this environment, including the following:

Learning lexicon and ontology. **MVP**: The VP learns many lexical items and related concepts by being told by the clinician: the name of its disease, various procedures, properties of the disease and procedures, etc. We are also preparing both the VP and the tutor to learn by reading online materials. The two main challenges of automatically processing online materials are the well-known problem of disambiguation (one of the central lines of work of our group) and the need to filter out unimportant information and concentrate on important information. **CLAD** will have the same learning capabilities of our other learning agents, with learning by reading online materials being especially important for its long-term utility in a field in which the state of knowledge changes so rapidly.



"I am feeling bad but not enough to go to a doctor yet."
Day: 360

"I am feeling worse. I'd better make an appointment to see the doctor."
Day: 600

"I never blindly follow doctors' suggestions. I need to think a little before I do the EGD."
Day: 600

"I don't know the risks of EGD. I'd better ask about them."
Day: 600

"I don't know the side effects of EGD. I'd better ask about them."
Day: 600

"I don't know if a EGD hurts. I'd better ask about that."
Day: 600

"I never blindly follow doctors' suggestions. I need to think a little before I do the EGD."
Day: 600

"OK. I have enough info about EGD - I'll make a decision."

EGD
IS-A MEDICAL-PROCEDURE
Day: 600

EGD-N1
CAT N
SYN-STRUC ((ROOT $VAR0) (CAT N))
SEM-STRUC (EGD)
Day: 600

EGD
IS-A DIAGNOSTIC-PROCEDURE
Day: 600

EGD
RISK 0
Day: 600

EGD
SIDE-EFFECTS-GENERAL 0
Day: 600

EGD
PAIN 0.2

Figure 5. Two of the "under the hood" panes of MVP.

Learning facts. **MVP**: The VP remembers its symptoms, the procedures it undergoes, its conversations with the doctor, important events in its simulated life (e.g., kicking a caffeine habit), etc. The tutor remembers all the information provided by the VP, all the moves made by the trainee, all the advice it gave the trainee, etc. **CLAD**: The advisor learns everything about the patient that is recorded in the chart, the clinician's actions, clinician preferences (as generalized about over time by CLAD), etc.

As an illustration of agent learning, let us look at a scene from a simulation run in MVP. At the point shown, the trainee (doctor) is interviewing the VP and recommends that she have an EGD. The VP does not know the word or concept EGD – they are absent from her lexicon and ontology. We will compare the dialog in the main MVP interface (Figure 4) with the human-readable traces of processing in the "under the hood" view (Figure 5). For reasons of space, Figure 5 shows only 2 of the 7 system process profilers inspectable via the "under-the-hood" panes in MVP.

From the doctor's first mention of EGD, the VP learns the concept EGD and the word EGD, as shown in the first two blocks of the right panel in Figure 5. In the first clause of the doctor's input, when he says "I suggest having an EGD", the VP can already guess that EGD must be some sort of medical procedure because *I would recommend having X* practically always refers to a medical procedure during a doctor-patient interview. So when the VP first learns the concept EGD, she makes it a child of MEDICAL-PROCEDURE in her ontology. In the lexicon, this becomes the first nominal sense of the word EGD, which is a typical noun mapped to the new concept EGD. When the doctor further specifies that EGD is a diagnostic procedure, the VP changes the filler of the IS-A link in the newly learned concept EGD to the more specific DIAGNOSTIC-PROCEDURE. The lexical entry need not change because the word EGD still maps to the concept EGD.

For reasons described in Section 3.3 below, the VP asks questions about the procedure before deciding whether or not to agree to it. She "remembers" the answer to each question by storing it as a property value in the ontological frame for

EGD, as shown by the 4th, 5th and 6th blocks in the right-hand-side panel of Figure 5.

Learning in our environment works the same for all agents, regardless of the nature and goals of the agent (VP, tutor, advisor) and the source of information (conversation, accessing information in a patient chart, searching texts on the web, etc.). The most important point is that the functionality of amending and expanding knowledge bases on the fly is in place so that the learning needs of all of our current and future agents can be met.

3.3 Decision Making

The decision-making capabilities of virtual patients in MVP currently include: 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; deciding whether to agree to a recommended test or intervention; deciding on certain aspects of lifestyle (e.g., drinking caffeinated beverages, smoking, etc.).

The decision-making behavior of specific instances of VPs 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 the VP's recollection of its past states of health, past communications and decisions, knowledge of self (preferences, fears, etc.) and general world knowledge.

VP reasoning is carried out through modeling the VP's goals and plans. Our approach to building a cognitive model is ideologically close to (though methodologically not identical with) the belief-desire-intention (BDI) model of agency (e.g., Bratman 1999). Unlike the classical BDI implementations (see, e.g., Wooldridge 2000), our approach centrally involves language comprehension and production. Algorithms for VP decision-making are presented in Nirenburg et al. 2008. Here we provide just a few examples, oriented around the same simulation scene discussed above.

The left-hand-side panel of Figure 5 shows select thoughts of the patient during the simulation – namely, those that reflect the operation of decision functions. The thoughts are actually English translations of the metalanguage thoughts generated and used by the system (we must emphasize that the system does not "think" in English). The first two thoughts are from an earlier point in the simulation: when the patient first starts to experience symptoms she decides not to see the doctor, but when the symptoms continue for a certain amount of time and reach a certain intensity (as determined by certain of her built-in characteristics and thresholds), she decides to go see the doctor. Between these two "thoughts" and the next set of thoughts that reflect her decision-making about the EGD, much can happen in the simulation, but it is not recorded in the Thoughts panel because there are no calls to decision functions. (In the current version of MVP, answering questions, for example, involves only decision-making with respect to language realization, even though one might view it as involving thinking on the part of the VP.) Each of the thoughts recorded in the Thoughts panel is a direct result of a call to a decision function.

When the EGD is initially suggested by the trainee, the VP evaluates the suggestion and realizes she must ask questions because her personality profile dictates that she does not make uninformed decisions, and she does not know what she feels she needs to know about the procedure. She knows specifically which questions to ask because her ontology includes an inventory of properties that procedures can have, and her personality traits make her interested in a certain subset of those – namely, risk, side effects and pain. If we look at the Thoughts and Knowledge Learned panels side by side, we see that each piece of information the VP wants to know about is knowledge that she then learns and incorporates back into her decision-making about whether or not to have the EGD. For reasons of space, the screen shot in Figure 5 could not show her final decision to have the EGD, but the trace of that thought comes out as "Since the doctor recommends it, and the risks, side effects and pain aren't enough to change my mind, I'll go ahead with the EGD."

For the CLAD application, we are enhancing the decision-making architecture to permit (a) more complex decision-making that takes into consideration more features, including the preferences of different "stakeholders" (the patient, the doctor, the insurance company, etc.), and (b) making decisions under uncertainty – i.e., when some property values are not known or are known with insufficient precision. As regards the latter, general clinical knowledge is typically formulated at the level of populations, not individual patients, and therefore uses features whose values are probability distributions. As the doctor gets to know a patient, he learns more and more about him, making his picture of the patient more individual and less generic; however, the doctor never knows every possible feature of the patient, so some population-level reasoning is still involved.

Guided by the goals of achieving richness of reasoning and reasoning under uncertainty, we decided to use influence diagrams (Howard and Matheson 2005) as the formalism for representing clinical decision-making knowledge. Influence diagrams are a convenient graphical representation facilitating the creation of Bayesian networks and/or decision trees which are well-suited to our domain.

The first decision we prepared CLAD to make is selecting which procedure to recommend when a patient has been diagnosed with the disease achalasia. There are three options: Heller myotomy, pneumatic dilation and BoTox injection. Although one could delineate dozens of considerations affecting the decision, for our first experiment we selected four of the most salient features to serve as top-level features (shown in boldface), three of which are calculated from other combinations of feature values (shown in italics):

- the **risk** of each procedure to the patient, which is calculated from the patient's *state of health* and the *inherent riskiness* of the procedure
- whether or not the patient can **afford** each procedure, which is calculated based on his *insurance policy* (or lack thereof) and *personal finances*
- the **quality of the best available practitioner** for each procedure within the desired time frame
- the expected **psychological distress** of the procedure, which is calculated based on the *inherent psychological distress* of the procedure, *any special fears or phobias* the person might have, and the person's overall level of *courage*.

Creating a hierarchy of features, and having as few as possible at the top level, greatly simplifies the job of recording ground truth for the use of influence diagrams, which involves providing utility scores for each combination of feature values. Let us render in English a few examples of utility scores for the decision in question:

The utility score for selecting BoTox if

- *the risk is high (e.g., the patient is ill)*
- *the patient can afford it*
- *the quality of the best practitioner is acceptable (not excellent or poor)*
- *the patient has moderate (not low or high) psychological distress associated with the procedure*

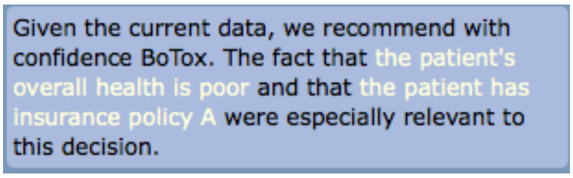
is 80/100.

If practitioner quality is increased to "excellent", the score goes up to 90/100.
If, in addition, psychological distress is reduced to "low", the score goes up to 100/100.

As mentioned earlier, all of the features can have population-level values (general clinical knowledge, recorded in the expert ontology) as well as patient-specific values (specific values obtained through interviews with the patient and tests). For example, if the typical psychological distress associated with the procedure is low at the population level but the patient has a phobia of some aspect of the procedure, then for this patient the distress is high.

At the time of making the decision, if the advisor knows the patient values, then those are used; if not, then the population-level values are used. However, the more population-level values are used, the less confidence the CLAD can have in its decision.

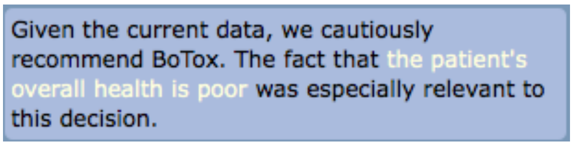
When CLAD provides advice to clinicians, it explicitly conveys its confidence in that advice as well as what was most salient in making the decision. If, for example, CLAD had the information about the patient shown in our first scenario above, it would provide the advice in Figure 6.



Given the current data, we recommend with confidence BoTox. The fact that the patient's overall health is poor and that the patient has insurance policy A were especially relevant to this decision.

Figure 6. One advice output of CLAD.

If, however, CLAD had no information about the patient's insurance or finances, it would return the advice in Figure 7.



Given the current data, we cautiously recommend BoTox. The fact that the patient's overall health is poor was especially relevant to this decision.

Figure 7. Another advice output of CLAD.

Various aspects of the CLAD interface show traces of its reasoning for review by clinicians.

Although a prototype of CLAD has already been developed, we are still in the early stages of work and expect to have significantly more results by the time we demonstrate the system at the conference.

As we further develop the decision-making architecture for CLAD, we will apply it, as needed,

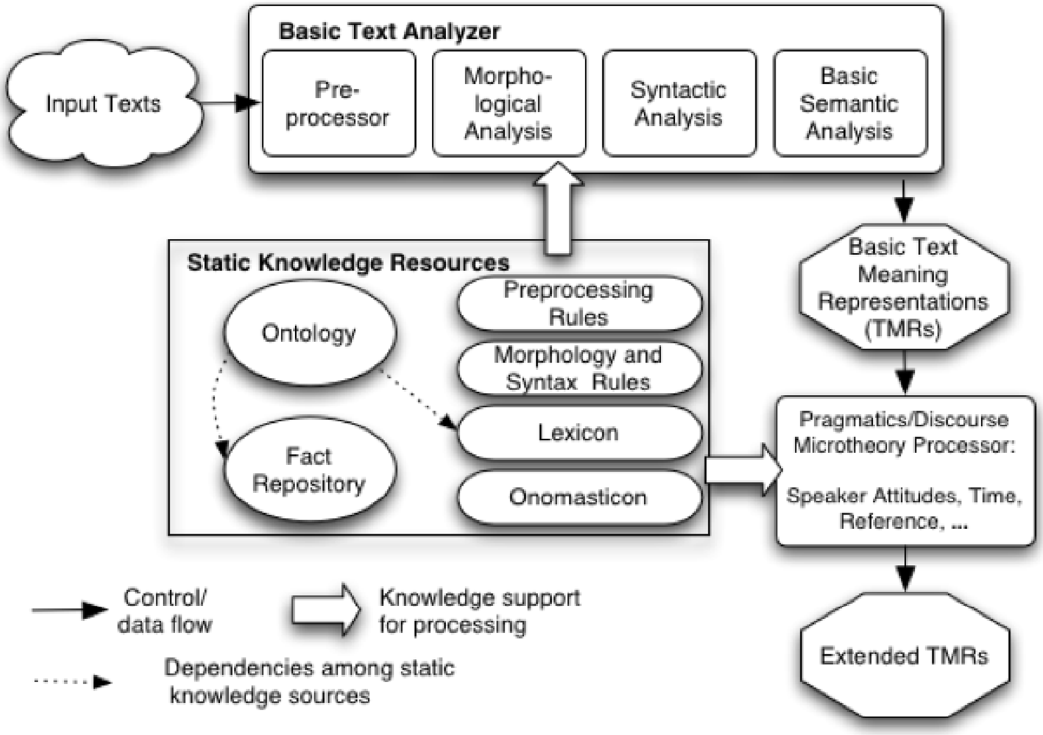

Basic Text Analyzer
Input Texts
Pre-processor
Morpho-logical Analysis
Syntactic Analysis
Basic Semantic Analysis
Basic Text Meaning Representations (TMRs)
Static Knowledge Resources
Ontology
Fact Repository
Preprocessing Rules
Morphology and Syntax Rules
Lexicon
Onomasticon
Pragmatics/Discourse Microtheory Processor:
Speaker Attitudes, Time, Reference, ...
Extended TMRs
Control/ data flow
Knowledge support for processing
Dependencies among static knowledge sources

Figure 8. Text analysis in OntoSem.

to the decision-making of both the VP and the tutor in the MVP application.

3.4 Language Processing

Our approach to treating language communication is unlike most other extant approaches in that all our agents do not reason on the basis of natural language strings: all natural language input is automatically translated into an unambiguous metalanguage using the battery of resources and processors shown in Figure 8. The agent carries out all necessary reasoning, learning, etc., using this metalanguage then, if it chooses to communicate a response to a human in its network, it translates the meaning of its response into English. To put an even finer point on it: all knowledge, reasoning, learning and simulation in our environment employs the same metalanguage, but in order to make our agents most useful in networks that include people, we give them the ability to communicate in English, despite the extra complexity and overhead of *natural language* → *metalanguage* → *natural language* translation.

Language processing is equally important in MVP and CLAD. In MVP the VP and the tutor must be able to communicate naturally with the trainee, and in CLAD the advisor must not only be able to interpret all the information in the patient chart, it must also eventually be able to follow the patient-doctor conversation just as a live advisor would. Consider some benefits of listening in on the patient-doctor conversation: CLAD could give advice/warnings before the clinician has written his notes, it might pick up important details that the clinician did not recognize were important, and it could foresee knowledge needs without the clinician having to ask for them – e.g., if the clinician suggests a surgery by a specific surgeon, CLAD could automatically look up his availability, and so on.

For details about the theory of ontological semantics underlying the OntoSem text processing system, see Nirenburg and Raskin 2004. For discussion of aspects of language processing specific to the VP in MVP, see McShane et al. 2008. Many more publications about the OntoSem approach to language processing are available for interested readers.

4 Comparisons with Others and Final Thoughts

A number of well-known cognitive architectures have been developed over the past 25 years, including ACT-R (e.g., Anderson et al. 2004), SOAR (e.g., Rosenbloom et al. 1991), PRODIGY (e.g., Carbonell et al. 1991), and many others – see, for instance the comparison table of twenty four extant cognitive architectures at http://members.cox.net/bica2009/cogarch/architectures.htm. While positioning our approach on the map of cognitive architectures is beyond the scope of this paper, in the first approximation, our work stands out because it pursues a finer-grain and broader-coverage approach to knowledge-based, language-oriented language processing than most and because it models the physiological body as well as the mind and is, therefore, able to introduce a new kind of perception – interoception. In addition to that, our approach attempts to support a variety of perception, reasoning and action operations on the basis of a uniform set of knowledge resources.

On the natural language processing side, to which we devoted relatively little space in this paper though it is a central aspect of our work, our approach has most in common with that taken in the dialog systems of James Allen and collaborators (e.g., Allen et al. 2006).

As concerns using virtual patients for cognitive training, practically all extant "virtual patients" are prefabricated branching narrative scenarios, organized as decision trees, that reflect a specific medical case (e.g., MedCases, Inc.). In these, user options are restricted and responses are highly pre-scripted, being delivered through multiple-choice questions. Most importantly, in such systems patient outcomes are fully predetermined by the prefabricated scenario. Recent work in this direction has been carried out in the European project eVIP (http://www.virtualpatients.eu/), where the center of gravity is in the area of interoperability, ease of authoring and curriculum acceptance of VPs (e.g., Ellaway 2009, Poulton 2009).

In conclusion, we have discussed two related systems whose intelligent agents are implemented in the same architecture and use largely overlapping knowledge resources and processors. By focusing on four *functionalities* of intelligent agents, we have attempted to convey that our agents are not limited to any specific profile or even to the medical domain – in fact, most of our knowledge resources and processors will support many application domains, the medical domain simply being the first to have been explored to-date. The agent architecture we have described allows a variety of configurations of processing and knowledge components to support applications involving complex multi-agent task-driven systems capable of decision making and dialog.

REFERENCES

Allen, J. F., Blaylock, N., et al. 2006. Chester: Towards a personal medical advisor. Biomedical Informatics 39(5), 500-513.

Anderson, J. R., Bothell, D., et al. 2004. An integrated theory of the mind. Psychological Review 111(4), 1036-1060.

Bratman, M. 1999. Faces of Intention. Cambridge University Press.

Carbonell, J. G., Knoblock, C. A., Minton, S. N. 1991. PRODIGY: An Integrated Architecture for Planning and Learning. Architectures for Intelligence. VanLehn, K. Hillsdale, NJ, Lawrence Erlbaum.

Ellaway, R. 2009. The realities of the virtual patient. In: International Conference on Virtual Patients. Krakow, Poland.

Jarrell, B., Nirenburg, S., McShane, M., Fantry, G., Beale, S. 2008. Revealing the conceptual substrate of biomedical cognitive models to the wider community. In: Medicine Meets Virtual Reality 16.

McShane, M., Fantry, G., Beale, S., Nirenburg, S, Jarrell, B. 2007a. Disease interactions in cognitive simulations for medical training. In: MODSIM World Conference, Medical Track.

McShane, M., Nirenburg, S., Beale, S., Jarrell, B., and Fantry, G. 2007b. Knowledge-based modeling and simulation of diseases with highly differentiated clinical manifestations. In: 11th Conference on Artificial Intelligence in Medicine.

McShane, M., Nirenburg, S., Beale, S. 2008. Two kinds of paraphrase in modeling embodied cognitive agents. In: Workshop on Biologically Inspired Cognitive Architectures, AAAI 2008 Fall Symposium.

Nirenburg, S., McShane, M., Beale, S. 2008. A simulated physiological/cognitive "double agent". In: Workshop on Naturally Inspired Cognitive Architectures at AAAI 2008 Fall Symposium.

Nirenburg, S., Raskin, V. 2004. Ontological Semantics. Cambridge: The MIT Press.

Poulton, T. 2009. From Virtual Patients to Virtual Worlds. In: International Conference on Virtual Patients. Krakow, Poland.

Rosenbloom, P. S., Newell, A., Laird, J. E. 1991. Toward the Knowledge Level in Soar: The Role of the Architecture in the Use of Knowledge. Architectures for Intelligence. VanLehn, K. Hillsdale, NJ, Lawrence Erlbaum.