

Two Types of Autonomous, Language-Endowed Learning Agents For Human Autonomy Teaming

Sergei Nirenburg and Marjorie McShane
Language-Endowed Intelligent Agents Laboratory
Rensselaer Polytechnic Institute
110 8th St. Troy, NY 12180-3590
USA

nirens@rpi.edu

mcsam2@rpi.edu

ABSTRACT

This paper addresses the topics of a) human-machine interaction and communication; b) comprehension; and c) joint learning/training. The rationale behind our approach to autonomy is our belief that both true autonomy in non-trivial applications and high-quality human-machine interaction can be supported only if the machine has the capability of understanding the world in which it operates and can interact with its human collaborators in the manner that is natural to the them.

1.0 INTRODUCTION

Autonomous artificial agents will reach their full potential when human members of their teams will be able to rely on them not only to carry out their assigned tasks but also to interact with them in ways similar to human-human interaction. The agent's ability to communicate with humans using natural language is a core prerequisite for attaining this goal. Human-level language understanding and generation capabilities require much more than the ability to match input text strings with stored or computed stock responses. Humans can explain, learn, anticipate, revise and adapt behavior (including communication-oriented behavior) in response to unforeseen circumstances, particular interlocutors and changing operational needs. This means that human-level language processing is strongly interdependent with general reasoning, perception and action capabilities. In order to approach human-level communication capability, language-endowed intelligent agents (LEIAs) must, therefore, model not only language *per se*. They must also model the world of their application domain, their past experiences, the situation at hand, their team members (both human and artificial), goals and actions associated with their team roles. A very important property of LEIAs is their ability to learn by instruction in natural language.

In this paper, we first briefly outline the basics of the cognitive architecture environment, OntoAgent, that supports implementation of LEIAs. Then we describe two types of LEIAs built on the basis of OntoAgent: a) virtual patients used for training medical personnel and b) robotic members of human-robot teams engaged in the task of jointly assembling furniture.

2.0 ONTOAGENT: THE ENVIRONMENT FOR DEVELOPING LEIAS

Core capabilities of a simulated, embodied, language-endowed intelligent agent include:

- behaving in a physiologically realistic manner
- experiencing, interpreting and remembering its own mental, physical, and emotional states
- communicating with people in natural language
- learning through experience, reasoning, reading, and being told

Two Types of Autonomous Language Endowed Learning Agents for Human Autonomy Teaming

- reasoning about its own plans and goals and those of its collaborators
- managing its memory – including forgetting and consolidating memories
- making decisions to further the plans that will fulfill its goals
- collaborating with human and intelligent-agent colleagues

In order to minimize development effort, maximize resource reuse, and avoid knowledge incompatibilities, all of these processes should be supported by an integrated knowledge substrate and a uniform knowledge representation language.

In the OntoAgent cognitive architecture, all perceptual, general cognitive, and language processing capabilities rely on the same ontological substrate, the same organization of episodic memory, and the same approach to ontologically-grounded knowledge representation. The approach to recording knowledge and computing meaning was originally developed for natural language processing pursued outside of a full agent architecture [1]. but it has required no fundamental extensions to be applied to more inclusive aspects of agent modeling.

Our computational cognitive modeling of LEIAs is a mature program of work that covers perception, reasoning and action – the typical pillars of cognitive architectures [2]. As Figure 1 shows, no matter which kinds of perceptual stimuli a LEIA receives, it must *interpret* them using its knowledge resources, which involves *translating* them from raw data streams into interpreted facts represented in its ontologically-grounded knowledge bases. Types of stimuli include language input, bodily signals generated through physiological simulations, visual stimuli, and, in principle, other stimuli.¹

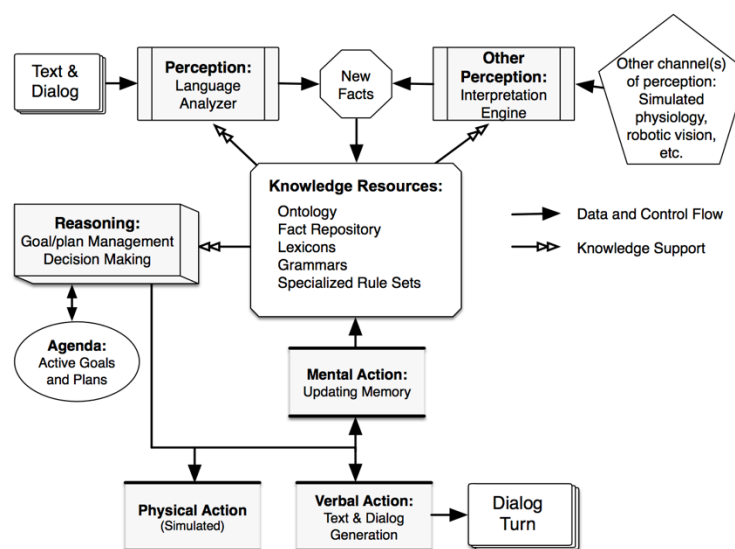


Figure 1. High-level view of OntoAgent architecture.

Since raw data are interpreted into ontologically-grounded facts, it does not matter whether a robotic LEIA knows that it must hit a given nail because its human teammate said, “Hit the nail” or because the latter said, “Hit this” while pointing at the nail – either way, the meaning representation will be the same, and the LEIA can use this new knowledge for subsequent reasoning about action.

¹ Language understanding and interoception were incorporated into LEIAs functioning as virtual patients in the Maryland Virtual Patient application [3]. Language understanding and robotic vision are currently being integrated into a robotic assistant [4].

The LEIA's knowledge representation language is a frame-based metalanguage in which concepts, or their instances, are described by properties. For example, an excerpt from the frame describing the event DRUG-DEALING in the ontology is as follows²:

DRUG-DEALING

IS-A	VALUE	CRIMINAL-ACTIVITY
AGENT	DEFAULT	CRIMINAL, DRUG-CARTEL
	SEM	HUMAN
	RELAXABLE-TO	SOCIAL-OBJECT
THEME	DEFAULT	ILLEGAL-DRUG
INSTRUMENT	SEM	MONEY
...		

When a LEIA learns about or remembers a particular *instance* of DRUG-DEALING, the concept types are replaced by numerically indexed concept instances, but the overall frame-based knowledge structures look largely the same. For example, the meaning of the language input **George Sanders sourced marijuana from a guy who lives in Phoenix** will be analyzed by a LEIA (that has specific memories in its belief repository) as follows.³

DRUG-DEALING-17

AGENT	HUMAN-233
THEME	MARIJUANA-21
BENEFICIARY	HUMAN-14
TIME	< FIND-ANCHOR-TIME ; A FUNCTION CALL TO DETERMINE ACTUAL TIME

HUMAN-14

GENDER	MALE
HAS-NAME	GEORGE SANDERS

CITY-812

HAS-NAME	PHOENIX
----------	---------

² For details about the OntoAgent representational substrate see [5]; for the particulars of the design and content of the OntoAgent ontology, see, e.g., [6-8].

³ A large amount of material in this instance of text meaning representation (TMR) was omitted for presentation purposes.

HUMAN-233

HAS-GENDER	MALE
RESIDES-IN	CITY-812

While in this paper we cannot detail all the features of this representation or the process that generates it⁴, we will comment on two particulars. First, when the verb *source* is used with a direct object indicating an illegal drug, the combination is compositionally understood to be an instance of the concept DRUG-DEALING. This disambiguation is supported by a combination of lexical and ontological knowledge. Second, when fillers of slots require further description, that description is provided in a co-referenced frame, meaning that there is no local reification – i.e., embedding of properties in slot fillers. E.g., HUMAN-233, who is the agent of DRUG-DEALING-17 (i.e., the seller of the drugs), has two additional known properties: HAS-GENDER MALE, and RESIDES-IN CITY-812 (i.e., the instance of the concept CITY corresponding to Phoenix in the LEIA’s episodic memory), which are listed in the frame headed by HUMAN-233. This ontologically-grounded metalanguage has many advantages for agent reasoning. Most importantly, it is unambiguous and the concepts referred to are described extensively in the ontology, which means that more knowledge is available for reasoning about language and the world.

LEIAs overall – and their natural language understanding (NLU) capabilities in particular – are modeled according to principles of human-inspired cognitive modeling, e.g., [5]. The NLU modeling tenets particularly relevant to this discussion are:

1. NLU in our approach follows the theory of Ontological Semantics [1] as augmented in subsequent writings (e.g., [10-12]). Under this approach, language understanding consists of translating input language strings into unambiguous, context-sensitive, ontologically-grounded text meaning representations (TMRs) that are well-suited to automatic reasoning.
2. LEIAs analyze inputs using horizontal incrementality (processing words of input as they come in the language stream) and vertical incrementality (processing inputs to varying layers of depth, using a battery of text understanding methods).⁵ Horizontal incrementality models human behavior and has practical utility as well – e.g., it is necessary for endowing LEIAs with the ability to interrupt (for clarification or correction) and to begin to act before a long utterance has been completed.
3. LEIAs operating in dialog contexts are not expected to arrive at a full and perfect interpretation of every input since even people do not do that. In real-life language use, many utterances do not make sense (e.g., thinking aloud while brainstorming), are irrelevant to the listener (e.g., a rant or a boring story at a cocktail party), or fall outside of the listener’s scope of knowledge or interest (e.g., a technical discussion about nuclear physics for most readers of this text). Instead of striving for a full interpretation, LEIAs focus on achieving an *actionable* interpretation of each utterance, meaning that they are seeking grounds for performing an action in response. This action may be not only physical or verbal, but also mental, such as remembering a vague interpretation and deciding not to work harder to concretize its meaning because it is outside of the agent’s scope of interest.
4. Some inputs have both more generic and more specific readings, which is due to what some refer to as the semantics vs. pragmatics distinction. LEIAs first generate the former and then, if they deem further specification worth the effort, they pursue the latter. This underscores that not only is NLU itself a reasoning-heavy enterprise, agents must also reason about how to carry out the process within the scope of their overall functioning.

⁴ For details on the OntoAgent language processing module, OntoSem, see, e.g., [9].

⁵ Of course, if an application is not time-sensitive, processing subsentential fragments can be skipped.

5. The LEIA's lexicon includes more kinds of knowledge to cover more linguistic phenomena (ellipsis, indirect speech acts, non-literal language, etc.) than do most lexicons. This is not only because we believe that people actually have and use such knowledge in processing language, but also because anchoring treatments of linguistic phenomena in the lexicon represents good practice in knowledge engineering as it helps to keep strict track of how declarative knowledge elements interact with processing modules.

Modeling NLU in LEIAs involves theory-based feature engineering, i.e., coming up with an inventory of properties (parameters) and their value sets to reflect aspects of context. So, the layers of context we will discuss involve different *types* of feature values accessed from different *sources* or *computed* in different ways. Note that the inventory of parameters is independently motivated by theoretical considerations – it is not developed specifically for any particular reasoning method; the reasoning methods that use the parameter inventory could include heuristic rules, logics, analogical reasoning on the basis of statistics, etc.

Since LEIAs are multifunctional intelligent agents, the features used to model them cover not just language-related material. LEIAs also routinely engage in many other aspects of cognition including, non-exhaustively, the agent's knowledge/beliefs about the world (ontology); its knowledge of remembered object, event and state instances (long-term episodic memory); the active object, event, and state instances comprising its situation model (working memory); its knowledge about language (including the lexicon and rule sets); its inventories of goals and plans; its personality traits; its cognitive biases; its physical, mental and emotional states; and its beliefs about all of these with respect to its interlocutors, which the agent must hypothesize its theory of mind, also called "mindreading" (e.g., [13]).

3.0 LEIAs AS VIRTUAL PATIENTS

Simulation-based training offer users the opportunity to gain extensive practical experience in a short time and without risk. For example, participants using the SHERLOCK II electronic troubleshooting system for F16 aircraft of the US Air Force were reported to have learned more in 20 hours of tutoring than in 4 years of field experience [14]. The Maryland Virtual Patient (MVP) system addresses the need for simulation-based training in the domain of clinical medicine, allowing medical trainees to develop clinical decision-making skills by managing a cohort of highly differentiated virtual patients in dynamic simulations, with the optional assistance of a virtual tutor.

The physician-educators who conceived of MVP set down the following requirements for the developers:

1. The system must expose students to virtual patients that demonstrate sophisticated, realistic behaviors, thus allowing the students to suspend their disbelief and interact naturally with them.
2. It must allow open-ended, trial-and-error investigation – i.e., learning through self-discovery – with the virtual patient's anatomy and physiology realistically adjusting to both expected and unexpected interventions.
3. It must offer a large population of patients suffering from each disease, with each patient displaying clinically relevant variations on the disease theme; these can involve the path and rate of disease progression, the profile and severity of symptoms, responses to treatments, and secondary diseases or disorders that affect treatment choices.
4. It must be built upon models that must:
 - be explanatory, both to provide transparency to the medical community who must endorse the system, and to provide tutorial support to students, explaining the *what* and *why* of what is happening.
 - integrate scientifically well-understood biological mechanisms with practical clinical knowledge

(population-level data, statistical evidence) to be used when causal explanations are not available.

- allow these non-explanatory “clinical bridges” to be replaced by biomechanical causal chains if they are discovered, without perturbation to the rest of the model.
- support automatic function and realism, without including *every* physiological mechanism known to medicine; in other words, create a useful application that does not impose the impossible precondition of creating a full-blown virtual human.
- cover diseases that are both chronic and acute, and both well and poorly understood by the medical community.
- allow students to have control of the clock – to advance the simulation to the next phase of patient management at will, thus simulating the doctor’s choices about when a patient is to come for a follow-up visit.
- Optionally, offer tutoring support, parameterizable to suit student preferences.
- allow virtual patients to make all the kinds of decisions that real patients do, such as when to see the doctor, whether to agree to tests and interventions, and whether to comply with the treatment protocol.

The virtual patients in MVP are “double agents” in that they display both physiological and cognitive function, as shown by the high-level system architecture in Figure 2.⁶ Physiologically, they undergo both normal and pathological processes in response to internal and external stimuli, and show realistic responses both to expected and to unexpected interventions. Cognitively, they experience symptoms, have lifestyle preferences, can communicate with the human user in natural language, have memories of language interactions and simulated experiences, and can make decisions based on their knowledge of the world, their physical, mental and emotional states, and their current goals and plans. An optional tutoring agent provides advice and feedback during the simulation. The “other medical personnel” include, at present, the agents that carry out tests and procedures and report their results.

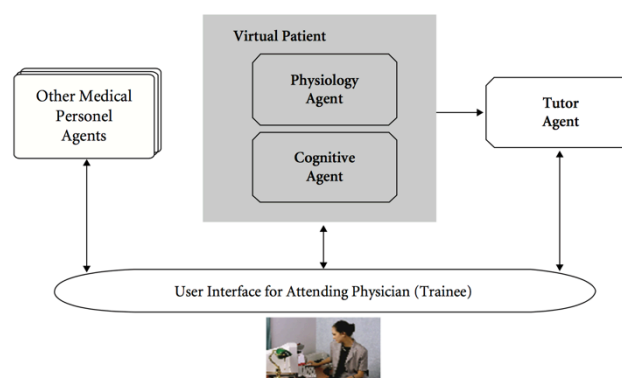


Figure 2. MVP architecture.

It is noteworthy that the MVP vision and modeling strategy anticipated the desiderata for virtual patient models detailed in the National Research Council’s 2009 joint report [15], as they were developed *before* that report was published. A short excerpt from that report illustrates the overlap:

⁶ In the realm of medical pedagogy, “virtual patients” have also been variously understood to refer to physical manikins, to live actors who roleplay with trainees, and to computer programs that rely on static, prefabricated decision trees.

“The clinician interacts with models and abstractions of the patient that place the raw data in context... These virtual patient models are the computational counterparts of the clinician's conceptual model of a patient... [The data] depict and simulate a theory about interactions going on in the patient and enable patient-specific parameterization... They build on submodels of biological and physiological systems...” [15]

MVP is a prototype system whose knowledge bases, software, and core theoretical and methodological foundations were developed from 2004 to 2013 (e.g., [16-23]).

The brief description of MVP below touches upon the modeling of the virtual patient's physiology and cognition and includes a sample system run, and a description of the “under the hood” traces of system functioning.

3.1 The Modeling of Physiology

Diseases are modeled as changes in the values of ontological properties 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. Disease models include 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 (to ensure that the disease manifests appropriately), the values for other physiological properties are variable within a specified range to allow for different instances of MVPs to differ in clinically relevant ways.

Roughly speaking, diseases fall into two classes: those for which the key causal chains are well understood and can drive the simulation, and those for which the key causal chains are not known. The models for the latter types of diseases rely on clinical observations about what happens and when, but not why. Most disease models integrate both kinds of modeling strategies in different proportions.

To develop computational cognitive models that are sufficient to support realistic patient simulations in MVP, a knowledge engineer leads physician-informants through the process of distilling their extensive and tightly coupled physiological and clinical knowledge into the most relevant subset and expressing it as causal chains with conditionals. Not infrequently, specialists are also called upon to formulate hypotheses about the unknowable, such as the preclinical (i.e., pre-symptomatic) stage of a disease and the behavior of the values of physiological properties between the times when tests are run to measure them. Such hypotheses are, by nature, imprecise. However, rather than permit this imprecision to grind agent development to a halt, we proceed in the same way as live clinicians do: by developing *a* model that is reasonable and useful, with no claims that it is the only model possible or that it precisely replicates human functioning.

The selection of parameters to be included in a disease model is guided by practical considerations. Parameters are included because (a) they can be measured by tests, (b) they can be affected by medications or treatments, and/or (c) they contribute to a physician's mental model of the disease. In addition to using parameters that directly reflect biologically attestable properties, we also include abstract parameters that foster the formulation of a compact, comprehensible model.⁷ Such features reflect the creative aspect of computational modeling that is not directly extractable from print resources.

Once an approach to modeling a disease has been devised and all requisite details have been elicited from the experts, the disease-related events and their participants are encoded in ontologically-grounded scripts written in the metalanguage of the LEIA's ontology. MVP includes both domain scripts and workflow scripts. Domain scripts describe basic physiology, disease progression and responses to interventions,

⁷ These features can be likened to the inclusion of intermediate categories in ontologies: although one does not typically talk about WHEELED-SURFACE-VEHICLES, this can be an appropriate node in an ontology.

whereas workflow scripts model the way an expert physician would handle a case, thus enabling automatic tutoring.

3.2 The Modeling of Cognition in Virtual Patients

Virtual patients need many cognitive capabilities. Their **language understanding** capabilities are described throughout this book. Their **language generation** involves two aspects, content and form. The content derives from reasoning and is encoded in ontologically grounded meaning representations. The form is constructed by templates, which proved sufficient for this application. Two other necessary cognitive capabilities of virtual patients are learning new words and concepts, and making decisions about action, which we consider in turn.

3.2.1 Learning New Words and Concepts

Learning is often a prerequisite to making a decision. Indeed, to give just one motivating example, no patient – real or virtual – should agree to things like medical procedures in full ignorance of their nature and risks. Table 1 shows an excerpt of a sample dialog in MVP between a virtual patient (P) and the human user/doctor (D) that illustrates the agent’s learning of elements of ontology and lexicon as a prerequisite for informed decision-making.

Table 1. Learning lexicon and ontology through dialog in MVP.

Dialog	Ontology learned	Lexicon learned
<i>D: You have achalasia.</i>	The concept ACHALASIA is learned and made a child of DISEASE.	The noun “achalasia” is learned and mapped to the concept ACHALASIA.
<i>P: Is it treatable?</i> <i>D: Yes.</i>	The value for the property TREATABLE in the ontological concept ACHALASIA is set to ‘yes’.	
<i>D: I think you should have a Heller myotomy.</i>	The concept HELLER-MYOTOMY is learned and made a child of medical-procedure. Its property treatment-option-for receives the filler heller-myotomy.	The noun “Heller myotomy” is learned and mapped to the concept HELLER-MYOTOMY.
<i>P: What is that?</i> <i>D: It is a type of esophageal surgery.</i>	The concept HELLER-MYOTOMY is moved in the ontology tree: it is made a child of surgical-procedure. Also, the theme of HELLER-MYOTOMY is specified as ESOPHAGUS.	
<i>P: Are there any other options?</i> <i>D: Yes, you could have a pneumatic dilation instead...</i>	The concept PNEUMATIC-DILATION is learned and made a child of MEDICAL-PROCEDURE.	The noun “pneumatic dilation” is learned and mapped to the concept PNEUMATIC-DILATION.

<i>D: (cont.) which is an endoscopic procedure.</i>	PNEUMATIC-DILATION is moved from being a child of MEDICAL-PROCEDURE to being a child of ENDOSCOPY.	
<i>P: Does it hurt?</i> <i>D: Not much.</i>	The value of the property PAIN-LEVEL in PNEUMATIC-DILATION is set to .2 (on a scale of {0-1}).	

When the virtual patient processes each of the doctor’s utterances, it automatically creates text meaning representations that it then uses for reasoning and learning. The text meaning representation for the first sentence is ACHALASIA-1 (EXPERIENCER HUMAN-1). The patient knows to make the newly learned concept ACHALASIA a child of DISEASE in the ontology because the lexical sense it uses to process the input “You have X” asserts that X is a disease. This sense is prioritized over other transitive meanings of the verb ‘have’ because the discourse context is a doctor’s appointment and the speaker is a doctor. A similar type of reasoning suggests that a Heller myotomy is a kind of MEDICAL-PROCEDURE. Our short dialog also shows two examples of belief revision: when the virtual patient learns more about the nature of the procedures HELLER-MYOTOMY and PNEUMATIC-DILATION, it selects more specific parents for them, thereby permitting the inheritance of more specific property values.

3.3 Making Decisions About Action

Virtual patients carry out dynamic decision-making in a style that approximates human decision-making, most notably because the agents’ decisions are *explanatory*. Whenever a decision needs to be made, the virtual patient first determines whether it has sufficient information to make it – an assessment that is based on a combination of what it actually knows, what it believes to be necessary for making a good decision, and its personality traits. If it lacks some decision-making knowledge, it can posit the goal of obtaining this knowledge, which is a metacognitive behavior that leads to learning. Formally speaking, a goal is an ontological instance of a property, whose domain and range are specified. Goals can appear on the agenda in four ways:

- **Perception via interoception.** The moment the patient perceives a symptom, the symptom appears in its short-term memory. This triggers the addition of an instance of the goal BE-HEALTHY onto the agenda. We assume that achieving the highest possible value of BE-HEALTHY (1 on the abstract scale {0,1}) is a universal goal of all humans and, in cases in which it seems that a person is not fulfilling this goal, he is simply prioritizing another goal, such as experience-pleasure.
- **Perception via language.** Any user input that requires a response from the virtual patient (e.g., a direct or indirect question) puts the goal to respond to it on the agenda.
- **A precondition of an event inside a plan is unfulfilled.** For example, most patients will not agree to an intervention about which they know nothing. So, one of the events inside the plan of decision-making about an intervention is finding out whichever features of it are of interest to the individual.
- **A predetermined period of time has passed since the latest instances of the events BE-DIAGNOSED or BE-TREATED have been launched.** This models regular check-ups and scheduled follow-up visits for virtual patients.

The goal BE-HEALTHY is put on the agenda when a virtual patient begins experiencing a symptom (this is made manifest by a signal from the agent’s physiology model via interoception). It remains on the agenda and is reevaluated when: (a) its intensity or frequency (depending on the symptom) reaches a certain level; (b) a new symptom arises; or (c) a certain amount of time has passed since the patient’s last evaluation of its

current state of health, given that the patient has an ongoing or recurring symptom or set of symptoms: e.g., “I’ve had this mild symptom for too long; I should see a doctor.”

When making decisions about its health care, the virtual patient incorporates the following types of features, which are used in the decision functions described below.

- (a) its physiological state, as perceived via interoception and remembered in its memory – particularly the intensity and frequency of symptoms. It is important to note that neither the patient nor the virtual tutor in the MVP system are omniscient about the patient’s physiological state; the *simulation system* knows this, but the intelligent agents, in accordance with the objective of making them human-like, do not.
- (b) certain character traits: TRUST, SUGGESTIBILITY and COURAGE (the inventory can in principle be modified to suit a particular personality theory in use)
- (c) certain physiological traits: PHYSIOLOGICAL-RESISTANCE, PAIN-THRESHOLD, and ABILITY-TO-TOLERATE-SYMPTOMS (i.e., how intense or frequent symptoms have to be before the patient feels the need to do something about them)
- (d) certain properties of tests and procedures: PAIN, UNPLEASANTNESS, RISK and EFFECTIVENESS. PAIN and UNPLEASANTNESS are considered typical side effects when viewed at the population level; the patient’s personal individual experience of them is described below.
- (e) two time-related properties: the FOLLOW-UP-DATE, i.e., the time the doctor told the patient to come for a follow-up, and CURRENT-TIME, the time of a particular interaction.

Most of these properties are scalar attributes whose values are measured on the abstract scale $\{0,1\}$. The values for all subjective features, (a) – (c), are selected for each instance of virtual patient by the patient author. That is, at the same time as a patient author selects the physiological traits of the patient – like the pace of disease progression and the patient’s response to treatments if they are administered – he selects certain traits specific to the cognitive agent as well as the amount of relevant world knowledge that the patient has in its ontology. Two decision functions, written in a simplified pseudocode, will suffice for illustration.

Decision Function 1: see-MD-or-do-nothing. This function decides when a patient goes to see the doctor, both initially and for follow-up visits.

```
IF FOLLOW-UP-DATE is not set
    AND SYMPTOM-SEVERITY > ABILITY-TO-TOLERATE-SYMPTOMS
THEN SEE-MD ; this triggers the first visit to the doctor
ELSE IF FOLLOW-UP-DATE is not set
    AND SYMPTOM-SEVERITY < ABILITY-TO-TOLERATE-SYMPTOMS
    AND the symptom has persisted > 6 months
THEN SEE-MD ; a tolerable symptom has been going on for too long
ELSE IF there was a previous visit
    AND at the time of visit symptom-severity <=.3
```

```
AND SYMPTOM-SEVERITY > .7
AND [SYMPTOM-SEVERITY – ABILITY-TO-TOLERATE-SYMPTOMS] > 0

THEN SEE-MD

ELSE IF there was a previous visit

    AND at the time of visit SYMPTOM-SEVERITY  $\in \{.3 .7\}$ 

    AND currently SYMPTOM-SEVERITY > .9

    AND [SYMPTOM-SEVERITY – ABILITY-TO-TOLERATE-SYMPTOMS] > 0

THEN SEE-MD ; there was a big increase in symptom triggering an unplanned visit to MD

ELSE IF there was a previous visit

    AND at the time of visit SYMPTOM-SEVERITY > .7

    AND currently SYMPTOM-SEVERITY > .8

THEN DO-NOTHING ; symptom severity was already high at last visit

ELSE IF CURRENT-TIME  $\geq$  FOLLOW-UP-VISIT.TIME

THEN SEE-MD ; go to the previously scheduled visit

ELSE DO-NOTHING
```

As should be clear, given the same symptom level, patients with a lower ability to tolerate symptoms will see the doctor earlier in the disease progression than patients with a higher ability to tolerate symptoms. Of course, one could incorporate any number of other character traits and lifestyle factors into this function, such as the patient's eagerness to be fussed over by doctors, the patient's availability to see a doctor around its work schedule, and so on. But even this inventory allows for considerable variability across patients (which is an objective of the entire training set-up).

Evaluation Function 2: EVALUATE-INTERVENTION. Among the decisions a patient must make is whether or not to agree to a test or treatment procedure suggested by the doctor, since many interventions carry some degree of pain, risks, side-effects or general unpleasantness. Some patients are endowed with such high levels of trust, suggestibility and courage in their personality profile that they will agree to anything the doctor says without question. All other patients must decide if they have sufficient information about the intervention to make a decision and, once they have enough information, they must decide whether they want to (a) accept the doctor's advice, (b) ask about other options, or (c) reject the doctor's advice. A simplified version of the algorithm for making this decision (which suffices for our purposes) can be stated informally as follows:

IF a function of the patient's trust, suggestibility and courage is above a threshold

OR the risk associated with the intervention is below a threshold (e.g., in the case, for a blood test)

THEN the agent agrees to intervention right away.

ELSE [*] IF the patient feels it knows enough about the risks, side-effects and unpleasantness of the

intervention (as a result of evaluating the function DETERMINE-IF-ENOUGH-INFO-TO-EVALUATE)

AND a call to the function EVALUATE-INTERVENTION establishes that the above risks are acceptable

THEN the patient agrees to the intervention.

ELSE IF the patient feels it knows enough about the risks, side-effects and unpleasantness of the intervention

AND a call to the function evaluate-intervention establishes that the above risks are not acceptable

THEN IF there are other options

THEN the physician proposes them and control is switched to [*].

ELSE the patient refuses the intervention.

ELSE IF the patient does not feel it knows enough about the intervention (as a result of evaluating the function DETERMINE-IF-ENOUGH-INFO-TO-EVALUATE)

THEN the patient asks for information about the specific properties that interest it, based on its personality traits: e.g., a cowardly patient will ask about risks, side-effects and unpleasantness,

whereas a brave but sickly person might only ask about side-effects.

IF a call to the function evaluate-intervention establishes that the above risks are acceptable

THEN the patient agrees to the intervention

ELSE the patient asks about availability of other options

IF other options are available

THEN the physician proposes them and control is switched to [*].

ELSE the patient refuses the intervention.

This decision function makes use of two other decision functions which we do not detail here, EVALUATE-INTERVENTION and DETERMINE-IF-ENOUGH-INFO-TO-EVALUATE (see Nirenburg et al. 2008 for details), as the point was a) to show the kinds of decisions virtual patients make, b) their approach to knowledge-based decision-making, and c) the kinds of dialog that must be supported to simulate the necessary interactions.

3.4 An Example System Run

To illustrate system operation, we present a sample user interaction with a virtual patient named Michael Wu. “Sample” is the key word here, as there are several substantively different – and countless trivially different – paths that this simulation could take based on what the user chooses to do during the interaction session. For example, users could choose different points in the dialog to offer clinically appropriate (or inappropriate!) interventions, or do nothing at all; they could ask the patient to come back for frequent or infrequent follow-ups; they could order appropriate or inappropriate tests; and they could have the tutor set to intervene frequently or only in cases of imminent mistakes – or not at all. Since Mr. Wu is a particular *instance* of a virtual patient, he has an inventory of property values that define him. This put some constraints on the available outcomes of the simulation. At the start of the process, Mr. Wu’s physiological, pathological, psychological, and cognitive profiles are specified as follows:⁸

- **Psychological Traits:** trust [.2], suggestibility [.3], courage [.4]
- **Physiological Traits:** physiological resistance [.9], pain threshold [.2], ability to tolerate symptoms [.4]
- **Patient’s knowledge of medicine:** minimal
- **Disease(s) explicitly authored for this patient:**⁹ Achalasia
- **Response to treatments if they are launched:** BoTox [effective, wearing off over 12 mos.], Pneumatic dilation [effective with regression], Heller myotomy [effective permanently]

The system user, who plays the role of attending physician, does not have direct access to any of this information and must learn everything about Mr. Wu through dialog, tests, and procedures. When the simulation clock is launched, the user must wait for Mr. Wu to present at the doctor’s office. Mr. Wu makes this decision using the decision function of Section 3.3. The points below describe what happens in this simulation run.

1. Mr. Wu presents with the chief complaint “difficulty swallowing”. This is Day 361 of the progression of his disease, which includes the preclinical stage and a portion of the first symptomatic stage. The user will not know this temporal information. Mr. Wu has had symptoms for some time but until now the evaluation function SEE-DOCTOR-OR-DO-NOTHING has returned the answer “do nothing.”
2. When the virtual patient presents at the office, this initiates the first dialog with the user. The user types in unconstrained English text (note the extensive use of difficult to automatically analyze elliptical expressions). The virtual patient analyzes it into TMRs, makes a decision about how to answer, and generates a response. The interaction begins as follows:

User: So, you have difficulty swallowing?
Mr. Wu: Yes.
User: Do you have difficulty swallowing solid food?

Mr. Wu: Yes.
User: Liquids?
Mr. Wu: No.

⁸ Both manual and automatic methods of creating virtual patient instances have been implemented in MVP, see (REFs).

⁹ Diseases can also be automatically triggered if their preconditions are met.

User: Do you have chest pain?
Mr. Wu: Yes, but it's mild.
User: Any heartburn?
Mr. Wu: No.
User: Do you ever regurgitate your food?

Mr. Wu: No.
User: How often do you have difficulty swallowing?
Mr. Wu: Less than once a week.

User: It is too early to take any action. Please come back in 9 months.

Mr. Wu: OK.

As we see, the user decides to do nothing – an important kind of decision in clinical medicine, and one that is difficult to teach since a physician's natural response to a patient asking for help is to do *something*.

3. After 9 months (on Day 661 of the disease progression) Mr. Wu comes back for his follow-up. He has regularly been running the evaluation function SEE-DOCTOR-OR-DO-NOTHING (since he is still symptomatic), but it always returned “do nothing” – i.e., do not schedule a new appointment prior to the scheduled follow-up. The user again asks about his difficulty swallowing, chest pain and regurgitation, using paraphrases of the original formulations (for variety and, in system demonstrations, to show that this is handled well by the NLU component). Mr. Wu responds that he has moderate chest pain, experiences regurgitation a few times a week, has difficulty swallowing solids daily and difficulty swallowing liquids occasionally. Note that the progression of difficulty swallowing from solids to liquids is a key diagnostic point that the user should catch: this suggests a motility disorder rather than an obstructive disorder.
4. The user posits the hypothesis that Mr. Wu has a motility disorder and advises Mr. Wu to have a test called EGD (esophagogastroduodenoscopy). Mr. Wu evaluates whether he will accept this advice using the function EVALUATE-INTERVENTION, described in Section 3.3. Since he is concerned about the risks, he asks about them. When the user assures him that they are extremely minimal, he agrees to the procedure.
5. A specialist lab technician agent virtually runs the test and delivers the results. This involves querying the physiological model underlying the simulation at the given point in time. The specialist agent returns the results with the interpretation: “Narrowing of LES with a pop upon entering the stomach. No tumor in the distal esophagus. Normal esophageal mucosa.” These results include both positive results and pertinent negatives.
6. The user reviews the test results, decides that it is still too early to intervene, and schedules Mr. Wu for another follow-up in 4 months.
7. When Mr. Wu presents in 4 months, he reports that the symptom that has changed the most is regurgitation, which he now experiences every day. Note that throughout the simulation the patient chart

is automatically populated with responses to questions, results of tests, etc., so the user can compare the patient's current state with previous states at a glance.

8. The user suggests having another EGD and Mr. Wu agrees immediately, not bothering to launch the evaluation function for EGD again since he agreed to it the last time.
9. Then the user suggests having two more tests: a barium swallow and esophageal manometry. Mr. Wu asks about their risks (that remains his only concern about medical testing), is satisfied that they are sufficiently low, and agrees to the procedures. The former returns "Narrowing of the lower esophageal sphincter with a bird's beak" and the latter returns "Incomplete relaxation of the LES, hypertensive LES, LES pressure: 53".
10. The user decides that these test results are sufficient to make the diagnosis of achalasia. The user records this diagnosis in Mr. Wu's chart.
11. The user suggests that Mr. Wu have a Heller myotomy. Mr. Wu asks about the risks and pain involved. The user responds that both are minimal. Mr. Wu agrees to have the procedure. The user tells him to come back for a follow-up a month after the procedure.
12. Mr. Wu has the procedure.
13. Mr. Wu returns in a month, the user asks questions about symptoms and there are none. The user tells Mr. Wu to return if any symptoms arise.

4.0 INTEGRATING LEIAS WITH ROBOTS

The main purpose of this section is to illustrate how an OntoAgent-based LEIA integrated within a robotic system can learn in a dialog with its human team member. Two characteristics set this system apart from the few recent systems that actually address aspects of extracting and representing linguistic meaning [5, 13]. First, it simultaneously addresses the challenges of a) learning-oriented language-based human-robotic interaction, b) symbol grounding, c) linguistic meaning extraction, and d) the enhancement and management of the episodic, semantic and procedural memory of a robot/agent. Second, the language processing component of the system and its associated knowledge resources address a broader set of meaning-related language phenomena, described at a finer grain-size of analysis than that used in other approaches. To implement language-based learning in a social robotics environment, we must address the co-dependence among three capabilities: language understanding, learning, and task-oriented physical, mental and verbal action. Language understanding and action require knowledge, while learning, achieved through language understanding and reasoning, automatically adds to that knowledge.

The system we describe is a social robot collaborating with a human user. The experimental domain is furniture assembly (e.g., [24]), widely accepted as useful for demonstrating human-robot collaboration on a joint activity. Roncone et al. [25] report on a Baxter robot supplied with high-level specifications, represented in the HTN formalism [26], of basic actions implementing chair-building tasks. Using a keyboard command or pressing a button, the user could trigger the execution of basic actions by triggering the operation of low-level task planners that the robot could directly execute. The robot could not reason about its actions, which were stored in its procedural memory as uninterpreted skills. The system described

here integrates the robotic architecture of [25] with the OntoAgent cognitive architecture [e.g., 5]. The integrated system allows the robot to a) learn the semantics of initially uninterpreted basic actions; b) learn the semantics of operations performed by the robot’s human collaborator when they are described in natural language; c) learn, name and reason about meaningful groupings and sequences of actions and organize them hierarchically; and d) integrate the results of learning with knowledge stored in its semantic and episodic memory and establish their connections with the robot’s procedural memory.

The core prerequisite for human-like learning is the ability to automatically extract, represent and use the meaning of natural language texts – utterances, dialog turns, etc. This task is notoriously difficult: to approach human-level capabilities, intelligent agents must account for both propositional and discourse meaning; interpret both literal and non-literal (e.g., metaphorical) meaning; resolve references; interpret implicatures; and, particularly in informal genres, deal with stops and starts, spurious repetitions, production errors, noisy communication channels and liberal (if unacknowledged) use of the least effort principle (e.g., [27]) by speakers and hearers. The language understanding module of OntoAgent, OntoSem [9], uses an ontology (semantic memory) of some 9,000 concepts with an average of 16 properties each; an episodic memory of remembered instances of ontological concepts; a semantic lexicon for English covering about 25,000 lexical senses; and a frame-oriented formalism suitable for representing the semantics of robotic actions, natural language utterances and results of the robot’s processing of other perceptual modalities (e.g., interoception, [5]).

The Process. At the beginning of the learning process, the robot can a) visually *recognize* parts of the future chair (e.g., the seat and the tool (screwdriver) to be used, b) *generate* meaning representations (MRs) of user utterances and physical actions and c) *perform* basic programmed actions: GET(OBJECT) from storage area to workspace, HOLD(OBJECT) and RELEASE(OBJECT). The user teaches the robot three types of things (implemented in three learning modules of Figure 3): a) *concept grounding*: the connection between basic actions and MRs of utterances that describe them; b) *legal sequences* of basic actions forming complex actions; and c) *augmenting the robot’s memory* with descriptions of newly learned complex actions and objects.

Suppose the user issues the call for the robot to execute `get(screwdriver)` and the utterance *Now you will fetch a screwdriver*. The physical-conceptual grounding learning module (LM) will link this procedure call with the representation of CHANGE-LOCATION-1 in the MR that OntoSem produces for the example utterance (Figure 4), thus linking the robotic and the cognitive architectures. This is done by adding the property `PM-LINK` with the filler `GET(SCREWDRIVER)` to the ontological concept instance `CHANGE-LOCATION-1`. The immediate purpose of this linking of the robotic and the cognitive architectures is to make the robot capable in its subsequent functioning to a) trigger basic actions autonomously on the basis of language input alone and b) learn complex event sequences by just being told, without having to actually perform the actions comprising the complex event.

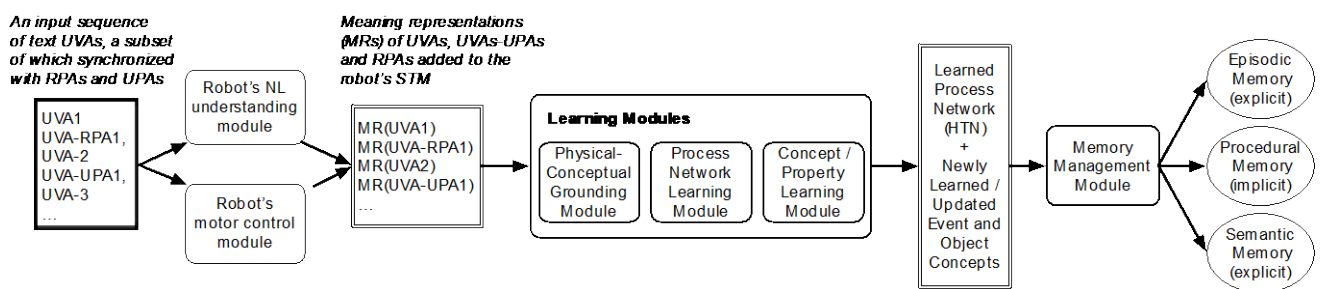


Figure 3. The core learning process. Input is a sequence of user verbal actions (UVAs) which explain user physical actions (UPAs) and issue commands to the robot, thus verbalizing robot’s physical actions (RPAs), which facilitates grounding the former in the latter. UVAs are interpreted into uniform meaning representation and provide input to grounding, process

network and concept/property learning modules (LMs). The memory management module (MMM) incorporates the results of learning into the episodic and semantic memories of the robot and mutually grounds RPAs in the robot's procedural memory and corresponding concepts in its semantic memory.

<p>SPEECH-ACT-1 type scope producer consumer time CHANGE-LOCATION-1 agent theme effect time token from-sense HUMAN-1 agent-of token from-sense SCREWDRIVER-1 theme-of token from-sense</p>	<p>command CHANGE-LOCATION-1 *speaker* ROBOT-0 time-0 ; time of speech ROBOT-0 SCREWDRIVER-1 BESIDE (AGENT.LOCATION THEME.LOCATION) > time-0 fetch move-v2 CHANGE-LOCATION-1 you you-n1 CHANGE-LOCATION-1 screwdriver screwdriver-n1</p>	<p>The robot can learn legal sequences of basic actions by understanding the user's utterances in their context. The robot organizes action sequences hierarchically and makes sure that any non-terminal nodes in the resulting process network represent meaningful complex actions. If the robot does not have specifications for these complex actions in its stored knowledge, it <i>learns new concepts for them</i>, on the basis of the MRs obtained by processing the relevant user utterances. We treat joint tasks as complex tasks and require the system to decompose them into subtasks carried out by each of the team members. Basic individual tasks include Robot Physical Actions (RPAs), User Physical Actions (UPAs) and User Verbal Actions, UVA. The RPAs and UPAs appear as terminal nodes in the process network being learned. The robot's activity that includes all the kinds of learning it does as well as updating its memory structures comes under the rubric of Robot Mental Action, RMA. Due to space constraints we cannot illustrate a complete process of assembling a chair (even the shortest version of the process numbers over 150 steps). So, we present a small subset of this process – assembling the third of the four legs of the chair – accompanied by associated robotic learning, as illustrated in Table 2. All UVAs are first analyzed and their meanings are represented as MRs. UVA1 signals the beginning of the subsequence and, together with UVA7, marks the boundaries of the complex action. All the RPAs and the UPA occurring within this span, in the order of their occurrence, will form the set of the terminal nodes in the subset of the overall process network, becoming children of the non-terminal designating the complex action of building the right back leg. Once this (sub)hierarchy is constructed, the non-terminal node at its root must be named. As the robot assembles the back leg for the first time, it learns the composition of this complex action (RMA1) and labels the parent node of this small subhierarchy with the name of the concept ASSEMBLE-RIGHT-BACK-LEG. It also learns the new object-type concept RIGHT-BACK-LEG, whose existence is the effect of the above action (RMA2) and updates the concept chair by adding right-back-leg as a filler of that concept's HAS-OBJECT-AS-PART property (RMA3). The newly learned concepts are illustrated in Figure 5. The results of the operation of the process network LM are recorded in the HAS-EVENT-AS-PART property of a result of the operation of the concept LM. At this stage in the process, the fillers of some of the properties in the concepts are tentative and are expected to be modified/tightened at the memory management stage.</p>
--	---	--

Figure 4. Meaning representation for the utterance *Now you will fetch a screwdriver* (simplified).

Memory management. Knowledge learned by the robot during each session with a human trainer (such as the sequence in Table 2) must be remembered so they can be used in subsequent functioning. Mutual grounding of basic actions and corresponding ontological events is recorded both in the robot's procedural memory (by augmenting the procedures implementing the robot's basic motor actions with links to their corresponding concepts in semantic memory) and in its semantic memory (by adding PM-LINKS, see above).

Table 2. Assembling the right back leg

UVA1	We will now build the right back leg
UVA2	Get another foot bracket
RPA1	GET(bracket-foot)
RPA2	RELEASE(bracket-foot)
UVA3	Get the right back bracket
RPA3	GET(bracket-back-right)
RPA4	RELEASE(bracket-back-right)
UVA4	Get and hold another dowel
RPA5	GET(dowel)
RPA6	HOLD(dowel)
UVA5	I am mounting the third set of brackets on a dowel
UPA1	The user affixes the foot and the right back brackets to the dowel
UVA6	Finished. Release the dowel
RPA7	RELEASE(dowel)
UVA7	OK, done assembling right back leg
RMA1	Learns action subsequence ASSEMBLE-RIGHT-BACK-LEG
RMA2	learns the object RIGHT-BACK-LEG with BRACKET-FOOT, BRACKET-BACK-RIGHT and DOWEL as fillers of HAS-OBJECT-AS-PART slot of RIGHT-BACK-LEG
RMA3	Adds RIGHT-BACK-LEG as a filler of HAS-OBJECT-AS-PART of CHAIR

Newly learned process sequences and objects (such as ASSEMBLE-RIGHT-BACK-LEG and RIGHT-BACK-LEG of Figure 5) must be incorporated in the robot’s long-term semantic and episodic memories. Due to space constraints, in what follows we give an informal description of the process.

For each newly learned concept, the memory management module (MMM) first determines whether this concept should be a) added to the robot’s semantic memory or b) merged with an existing concept. To make this choice, the MMM uses an extension of the concept matching algorithm of [28-29]. This algorithm is based on unification, with the added facility for naming concepts and determining their best position in the hierarchy of the ontological world model in the robot’s long-term semantic memory.

ASSEMBLE-RIGHT-BACK-LEG

IS-A	PHYSICAL-EVENT
AGENT	HUMAN, ROBOT
THEME	RIGHT-BACK-LEG
INSTRUMENT	SCREWDRIVER
HAS-EVENT-AS-PART	GET(ROBOT, BRACKET-FOOT) RELEASE(USER, BRACKET-FOOT) GET(ROBOT, BRACKET-BACK-RIGHT) RELEASE(USER, BRACKET-BACK-RIGHT) GET(ROBOT, DOWEL) HOLD(ROBOT, DOWEL) MOUNT(USER, {BRACKET-FOOT, BRACKET-BACK-RIGHT}, DOWEL)
PART-OF-EVENT	ASSEMBLE-CHAIR
EFFECT	RIGHT-BACK-LEG //default effects are events; if filler of effect is an object, this means the effect is its existence

RIGHT-BACK-LEG

IS-A	CHAIR-PART
HAS-OBJECT-AS-PART	BRACKET-FOOT, BRACKET-BACK-RIGHT, DOWEL
PART-OF-OBJECT	CHAIR

Figure 5. Concepts learned as a result of processing the sequence in Table 2.

As an illustration, suppose, the concept matching algorithm has determined that the newly learned concept ASSEMBLE-RIGHT-BACK-LEG must be added to the robot’s semantic memory. In this case, the algorithm also suggests the most appropriate position for the concept in the ontological hierarchy. This is determined by

comparing a) the inventory and b) sets of fillers of the properties defined for the new concepts and for the potential parents of the new concepts in the ontological hierarchy. In the example illustrated in Figure 5, the LM used the safest, though the least informative, filler (PHYSICAL-EVENT) for the IS-A property of ASSEMBLE-RIGHT-BACK-LEG. To determine the appropriate parent, the algorithm traverses the ontological hierarchy from PHYSICAL-EVENT down until it finds the closest match that does not violate recorded constraints (see REF for a description of the possible eventualities that this process can encounter).

```

RIGHT-BACK-LEG
IS-A
HAS-OBJECT-AS-PART CHAIR-PART
PART-OF-OBJECT BRACKET-FOOT,
DOWEL
CHAIR
    
```

FIGURE 6. A possible existing ontological concept RIGHT-BACK-LEG

```

RIGHT-BACK-LEG
IS-A
HAS-OBJECT-AS-PART SOFA-PART
PART-OF-OBJECT BRACKET-FOOT,
DOWEL
SOFA
    
```

FIGURE 7. Another possible existing ontological concept RIGHT-BACK-LEG

Suppose now that a concept RIGHT-BACK-LEG already exists in the ontology. If this concept is described as in Figure 6, then after comparing this concept with the newly learned concept (see Table 2), the MMM will, instead of adding a new (possibly renamed) concept to the robot’s semantic memory, just add an *optional* filler BRACKET-BACK-RIGHT to the HAS-OBJECT-AS-PART property of the existing concept of Figure 6, thus merging the existing and the newly learned knowledge. If, however, the existing concept is as illustrated in Figure 7, then, because of the mismatch of the fillers of PART-OF properties between the newly learned and the standing concept, the MMM will yield two new concepts, as illustrated in Figure 8, and create a new common parent concept RIGHT-BACK-LEG for them.

```

RIGHT-BACK-LEG-CHAIR
IS-A
HAS-OBJECT-AS-PART CHAIR-PART
PART-OF-OBJECT BRACKET-FOOT,
DOWEL,
BRACKET-BACK-RIGHT
CHAIR

RIGHT-BACK-LEG-SOFA
IS-A
HAS-OBJECT-AS-PART SOFA-PART
PART-OF-OBJECT BRACKET-FOOT,
DOWEL
SOFA
    
```

FIGURE 8. Two new concepts RIGHT-BACK-LEG

Note the need of modifying the names of the concepts. An important case of merging several versions of a concept in one representation is the system’s ability to represent the content of an action’s HAS-EVENT-AS-PART property as an HTN, augmented with the means of expressing temporal ordering, optionality and valid alternative action sequences. Semantic memory stores the robot’s knowledge of concept types. So, for example, it will contain a description of what the robot knows about chairs and chair legs. This knowledge will be used to feed the reasoning rules the robot will use while processing language, learning and making decisions. To make the robot more human-like, we also support reasoning by analogy. For this purpose, the MMM records sequences of RPAs, UPAs and UVAs that the robot represents and carries out during specific sessions of interacting with specific users in the robot’s long-term episodic memory. The contents of the episodic memory will also support the robot’s ability to “mind-read” its various users [e.g., [13, 28] and, as a result, to be able to anticipate their needs at various points during joint task execution as well as interpret their UVAs with higher confidence.

The system briefly described here concentrates on robotic learning through language understanding. This learning results in extensions to and modifications of the three kinds of robotic memory – the explicit semantic and episodic memory and the implicit (skill-oriented) procedural memory. The expected practical

impact of the ability to learn and reason will include the robot's ability to a) perform complex actions without the user having to spell out a complete sequence of basic and complex actions; b) reason about task allocation between itself and the human user; and c) test and verify its knowledge through dialog with the user, avoiding the need for large numbers of training examples required by learning by demonstration only. The inability of the state-of-the-art deep learning-based systems to provide human-level explanations is a well-known constraint on the utility of such systems. The cognitive robots we develop will still be capable of sophisticated reasoning by analogy but will be also capable of explaining their decisions and actions. Finally, our approach to learning does not depend on the availability of "big data" training materials. Instead, we model the way people learn since early childhood and throughout their lives – by being taught using natural language.

An important contribution of our work is overt modeling of the robot's memory components. These components include an implicit memory of skills and explicit memories of concepts (objects, events and their properties) and of instances of sequences of events (episodes, represented in our system as hierarchical transition networks, HTNs). The link established between the implicit and explicit layers of memory allows the robot to reason about its own actions. Scheutz et al. [30] discuss methodological options for integrating robotic and cognitive architectures and propose three "generic high-level interfaces" between them – the perceptual interface, the goal interface and the action interface. In our work, the basic interaction between the implicit robotic operation and explicit cognitive operation is supported by interactions among the three components of the memory system of the robot.

The first enhancement of the current learning system will consist in demonstrating how, after RPAs are mutually grounded in ontological concepts, the robot will be able to carry out commands or learn new action sequences by acting on UVAs, without any need for direct triggering through software function calls or hardware operations. Next, we intend to add text generation capabilities, both to allow the robot a more active role in the learning process (by asking questions) and to enrich interaction during joint task performance with a human user. Another novel direction of work will involve adapting to particular users – modeling robots' individuality and related phenomenological ("first-person" view) aspects of its internal organization and memory, developing and making use of mindreading capabilities that will in turn facilitate experimentation in collaboration among agents with different "theories of minds of others," similarly to the capability present in the MVP implementation of OntoAgent-based LEIAs.

5.0 FINAL WORDS

We have successfully used the infrastructure of OntoAgent in two very different applications of language-endowed intelligent agents. The MVP involves a disembodied virtual patient agent capable of interacting with the human user in a manner approaching the human level. The virtual patients are also supplied with a simulated physiology and pathology, formulated using the standard representation substrate of OntoAgent. This demonstrates that OntoAgent can successfully address issues beyond language processing, specifically, integrate different modes of perception. In the case of MVP, language inputs are integrated with inputs from interoception, made possible by the presence of the physiology model.

The knowledge structures in OntoAgent facilitate not only perception but also reasoning and decision-making, implemented mostly in the framework of goal- and plan-oriented functioning but also attempting to account for reactive actions. It is important to note that our environment can accept a variety of realizations of goal- and plan-based decision-making, which means that our general tendency is to be task-oriented rather than method-oriented. Thus, for MVP we developed both the classical rule-based decision procedures (such as those illustrated in Section 3.3 above) and decision procedures based on Bayesian networks.

In the process of integrating LEIAs with robots, we have demonstrated a way to integrate cognitive and robotic architectures and thus add physical (motor) actions to the repertoire of LEIAs that are already

capable of verbal and mental (remembering, learning, making decisions) actions.

The knowledge substrate that facilitates LEIAs is not trivial. However, we have demonstrated that once a core knowledge bases (ontology, lexicon, belief repository structures, goal and plan inventories, decision functions, etc.) have already been developed, adaptation to new domains can proceed with moderate resource allocation.

The overall approach we have briefly described here follows the premise that the only way to attain human-level functioning for autonomous entities operating in teams with people is to have these entities model both **what** people do and **how** they understand, reason, make decisions and operate. The vast majority of current AI follows the former goal. To fully succeed, the field must return to seriously taking into account the latter.

REFERENCES

- [1] Nirenburg, S. and V. Raskin. 2004. *Ontological Semantics*. MIT Press.
- [2] P. Langley, J. E. Laird, S. Rogers, *Cognitive architectures: Research issues and challenges*, *Cognitive Systems Research* 10 (2) (2009) 141–160.
- [3] Nirenburg, S., McShane, M., & Beale, S. (2008). A simulated physiological/cognitive “double agent”. In J. Beal, P. Bello, N. Cassimatis, M. Coen, & P. Winston (Eds.), *Papers from the AAAI Fall Symposium "Naturally Inspired Cognitive Architectures"*. AAAI Technical Report FS-08-06. Menlo Park, CA: AAAI Press.
- [4] Nirenburg, S., M. McShane, S. Beale, P. Wood, B. Scassellati, O. Mangin and A. Roncone. 2018. *Toward Human-Like Robot Learning*. Proceedings of NLDB 2018, Paris, June.
- [5] McShane, Marjorie and Sergei Nirenburg. 2012. A knowledge representation language for natural language processing, simulation and reasoning. *Intl. Journal of Semantic Computing* 6(1).
- [6] Nirenburg, S., Raskin, V. and Onyshkevych, B. 1995. *Apologiae Ontologiae*. Proceedings of the Conference on Theoretical and Methodological Issues in Machine Translation. Leuven, Belgium. July.
- [7] Nirenburg, S., M. McShane and S. Beale. 2009. A Unified Ontological-Semantic Substrate for Physiological Simulation and Cognitive Modeling. Proceedings of the International Conference on Biomedical Ontologies. Buffalo, July.
- [8] McShane, M. and S. Nirenburg. 2013. Use of ontology, lexicon and fact repository for reference resolution in Ontological Semantics. In Oltramari, A., Vossen, P., Qin, L., Hovy, E. (Eds.), *New Trends of Research in Ontologies and Lexical Resources*. Springer. New Trends of Research in Ontologies and Lexical Resources, Theory and Applications of Natural Language Processing; pp 157-185.
- [9] McShane, M., Nirenburg, S. and Beale, S. 2016. Language understanding with Ontological Semantics. *Advances in Cognitive Systems* 4: 35-55.
- [10] McShane, M., Beale, S., and Babkin, P. 2014. **Nominal compound interpretation by intelligent agents**. *Linguistic Issues in Language Technology (LiLT)*, 10(1): 1-34
- [11] McShane, M., Nirenburg, S. and Beale, S. 2015. **The Ontological Semantic treatment of multiword expressions**. *Lingvisticae Investigationes*, 38(1): 73-110. John Benjamins Publishing Company.
- [12] McShane, M. and Babkin, P. 2016. **Detection and resolution of verb phrase ellipsis**. *Linguistic Issues in Language Technology (LiLT)*.
- [13] McShane, M. 2014. **Parameterizing mental model ascription across intelligent agents**. *Interaction Studies*, 15(3): 404-425.
- [14] Evens, M., and J. Michael. 2006. *One-on-One Tutoring by Humans and Computers*. New Jersey and London: Lawrence Erlbaum and Associates, Publishers.
- [15] Stead, W.W. and H.S. Lin (eds.) 2009. *Computational Technology for Effective Health Care: Immediate Steps and Strategic Directions*. Committee on Engaging the Computer Science Research Community in Health Care Informatics; National Research Council.
- [16] McShane, M., Fantry, G., Beale, S., Nirenburg, S. and Jarrell, B. 2007. **Disease interaction in cognitive simulations for medical training**. In Proceedings of the MODSIM World Conference &

- Exposition, Virginia Beach
- [17] McShane, M., Nirenburg, S., Beale, S., Jarrell, B. and Fantry, G. 2007. **Knowledge-based modeling and simulation of diseases with highly differentiated clinical manifestations**. In Bellazzi, R., Abu-Hanna, A., Hunter, J. (Eds.), *Artificial Intelligence in Medicine: Proceedings of the 11th Conference on Artificial Intelligence in Medicine, AIME 2007*. Volume 4594 of the series *Lecture Notes in Computer Science*, pp. 34-43. Berlin, Heidelberg: Springer-Verlag.
 - [18] McShane, M., Jarrell, B., Fantry, G., Nirenburg, S., Beale, S. and Johnson, B. 2008. **Revealing the conceptual substrate of biomedical cognitive models to the wider community**. In Westwood, J.D., Haluck, R.S., Hoffman, H.M., Mogel, G.T., Phillips, R., Robb, R.A. et al. (Eds.), *Medicine Meets Virtual Reality 16: Parallel, combinatorial, convergent: NextMed by Design*, pp. 281-286. Amsterdam; Berlin; Oxford; Tokyo; Washington, DC: IOS Press.
 - [19] McShane, M., Nirenburg, S., Beale, S., Jarrell, B., Fantry, G. and Mallott, D. 2013. **Mind-, body- and emotion-reading**. *Proceedings of the Annual Meeting of the International Association for Computing and Philosophy (IACAP 2013)*.
 - [20] Nirenburg, S., McShane, M. and Beale, S. 2008. **A simulated physiological/cognitive "double agent"**. In Beal, J., Bello, P., Cassimatis, N., Coen, M. and Winston, P. (Eds.), *Papers from the AAAI Fall Symposium "Naturally Inspired Cognitive Architectures"*. AAAI Technical Report FS-08-06. Menlo Park, CA: AAAI Press.
 - [21] Nirenburg, S., McShane, M., Beale, S. and Jarrell, B. 2008. **Adaptivity in a multi-agent clinical simulation system**. In Honkela, T., Pöllä, M., Paukkeri, M. And Simula, O. (Eds.), *Proceedings of the 2nd International and Interdisciplinary Conference on Adaptive Knowledge Representation and Reasoning (AKRR'08)*, pp. 32-39.
 - [22] Nirenburg, S., McShane, M. and Beale, S. 2010. **Aspects of metacognitive self-awareness in Maryland Virtual Patient**. In Pirrone, R., Azevedo, R. and Biswas, G. (Eds.), *Cognitive and Metacognitive Educational Systems: Papers from the AAAI Fall Symposium*. AAAI Technical Report FS-10-01, pp. 69-74.
 - [23] McShane, M., Beale, S., Nirenburg, S., Jarrell, B. and Fantry, G. 2012. **Inconsistency as a diagnostic tool in a society of intelligent agents**. *Artificial Intelligence in Medicine*, 55(3): 137-48.
 - [24] Knepper, R.A., T. Layton, J. Romanishin, and D. Rus. 2013. IkeaBot: An autonomous multi-robot coordinated furniture assembly system. *IEEE International Conference on Robotics and Automation*.
 - [25] Roncone, A., O. Mangin and B. Scassellati. 2017. Transparent Role Assignment and Task Allocation in Human Robot Collaboration. *Proceedings of International Conference on Robotics and Automation*. Singapore.
 - [26] K. Erol, J. Hendler, and D. S. Nau 1994. HTN Planning: Complexity and Expressivity. *Proceedings of AAAI-94*.
 - [27] Piantadosi, S. T., Tily, H. & Gibson, E. (2012). The communicative function of ambiguity in language. *Cognition* 122, 280–291.
 - [28] English, J. and S. Nirenburg. 2007. *Ontology Learning from Text Using Automatic Ontological-Semantic Text Annotation and the Web as the Corpus*. *Proceedings of the AAAI-07 Spring Symposium on Machine Reading*.
 - [29] Nirenburg, S., T. Oates and J. English. 2007. *Learning by Reading by Learning to Read*. *Proceedings of the International Conference on Semantic Computing*.
 - [30] Scheutz, M. J. Harris and P. Schmermerhorn. 2013. *Systematic Integration of Cognitive and Robotic Architectures*. *Advances in Cognitive Systems* 2:277-296.