

HYBRID METHODS OF KNOWLEDGE ELICITATION WITHIN A UNIFIED REPRESENTATIONAL KNOWLEDGE SCHEME

Sergei Nirenburg, Marjorie McShane, Stephen Beale
University of Maryland Baltimore County, Maryland, U.S.A.

Roberta Catizone
Onyx Consulting, Sheffield, U.K.

Keywords: Knowledge engineering, Knowledge elicitation, Ontology, Intelligent agents, Automatic reasoning, Influence diagrams.

Abstract: This paper presents a case study showing how hybrid methods of knowledge elicitation can be used to build models in support of the functioning of intelligent agents. What facilitates both the elicitation of knowledge and its conversion into actionable models is the use of a unified representational knowledge scheme – specifically, an unambiguous, ontologically grounded metalanguage that serves as the language of all recorded knowledge as well as the language in which agents remember and reason.

1 INTRODUCTION

The process of acquiring knowledge from experts to support sophisticated intelligent agents is known to be an expensive and difficult enterprise, leading to a long history of research in knowledge acquisition methodologies. As Cooke (no date) reports, reviews and categorization schemes for knowledge elicitation and modeling “abound”. But, as Ford and Sterman (1998) write, “While many methods to elicit information from experts have been developed, most assist in the early phases of modeling: problem articulation, boundary selection, identification of variables, and qualitative causal mapping... The literature is comparatively silent, however, regarding methods to elicit the information required to estimate the parameters, initial conditions, and behavior relationships that must be specified precisely in formal modeling.”

We have been experimenting with hybrid knowledge elicitation and modeling in the OntoAgent environment, whose recent applications include Maryland Virtual Patient (MVP) and CLinician’s ADvisor (CLAD). 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 artificial intelligent agents play-

ing the role of virtual patients (VPs) (McShane et al., 2007a; Jarrell et al., 2008). These VPs can suffer from various diseases and combinations of diseases (McShane et al., 2007b) and are capable of realistic physiological and cognitive responses even to unexpected actions on the part of the user (Nirenburg et al., 2008a,b). CLAD seeks to decrease the cognitive load on clinicians by providing various kinds of contextualized decision support (McShane et al., Submitted).

Both of these applications require many kinds of intelligent agent behavior. In this paper we will focus on two of them: physiological simulation and decision making in the realm of best clinical practices. Very briefly, here is how each of these functionalities is used. *Physiological Simulation:* In MVP, physiological simulation permits the virtual patient to “live” over time and respond realistically to non-scripted interventions by the user. In CLAD, physiological simulation is used by the advisor to project patient outcomes (over time and in response to different treatment strategies) as input to decision making. *Decision making about clinical practices:* In MVP, knowledge about best clinical practices is used by the tutor to provide feedback and advice to trainees. In CLAD, knowledge about best clinical practices is used by the advisor to carry out the main

function of the system – time- and context-sensitive advice giving.

As part of our work on knowledge-based applications, we have been developing knowledge elicitation methodologies that permit domain experts to independently carry out as much work as possible before collaborating with knowledge engineers. We have found truth in Hoffman and Lintern's (2006) statement that "methodology benefits from opportunism": i.e., the need of a concrete knowledge elicitation project can offer the opportunity of discovering new methodologies and new combinations of methodologies. In turn, those methodologies can foster more streamlined thinking for future modeling.

Before proceeding to the body of the paper, which addresses specific knowledge elicitation strategies used in the OntoAgent environment, some background about our knowledge representation scheme and our approach to knowledge elicitation is in order.

The Knowledge Representation Scheme. All knowledge in our system is recorded using an ontologically grounded metalanguage that derives from the theory of Ontological Semantics (Nirenburg and Raskin 2004), which is implemented in the OntoAgent (an extension of the earlier "OntoSem") environment (Beale et al. 2004; McShane, Nirenburg and Beale, 2005). The OntoAgent meaning representation language expresses meaning using unambiguous ontological concepts and their instances, which are linked to each other using ontologically recorded properties. This metalanguage is not only the language of recorded static knowledge, it is the language of thought of all intelligent agents. The OntoAgent ontology that forms the core of the environment currently contains about 9,500 concepts, most of which belong to the general domain. It includes not only slot-filler style knowledge but also domain and workflow scripts (i.e., complex events) of the type introduced by Schank and Abelson (1977). In the medical domain, these scripts cover everything from normal physiology to pathology to best clinical practices to decision-making on the part of the physician and the patient.

Since our human-like intelligent agents must be able to communicate with people in natural language, the environment includes a large suite of natural language processing resources and tools, including a large lexicon whose semantic descriptions employ ontological concepts. When intelligent agents perceive language input, they automatically translate it into the unambiguous metalanguage that they use for remembering and reasoning; on the other end, when

they have something to communicate to a person, they formulate the content in the metalanguage then translate it into English.

Our approach to Knowledge Elicitation (KE).

Note: To ground our knowledge elicitation strategy in the tradition of past work, we will point to how it conforms to all seven of Breuker's KADS "Knowledge Acquisition and Domain Structuring" principles for the elicitation of knowledge and construction of a system (Breuker, 1987, as summarized in Shadbolt and Burton, 1995).

KE, for the first six diseases modeled in our environment, was carried out through collaboration between domain experts and knowledge engineers, primarily using unstructured and semi-structured interviews. (Cf. the KADS principle that the knowledge should be analyzed before design and implementation begin.) However, based on that experience, we have been able to create an automatic KE system that guides the expert through the process of providing much of the knowledge required for disease modeling. Our approach derives largely from past work in a different domain – computational field linguistics. Our Boas system (McShane and Nirenburg 2003) was a mixed-initiative KE system aimed at quickly gathering formally organized, machine tractable knowledge about lesser-studied languages from speakers of the language without the assistance of a linguist. The mixed-initiative, expectation-driven methodology used there has translated directly into our OntoElicit system for KE in the medical domain.

Developing OntoElicit involved (a) organizing the domains of normal physiology, pathology and clinical knowledge into classes of parameters and value sets (cf. the KADS principle that the analysis should be model-driven as early as possible); (b) applying past experience in modeling 6 esophageal diseases; (c) taking into consideration the nature of the target, generalized processors that were developed to support simulation and reasoning across agents, applications and diseases; (d) anticipating the needs of non-developer domain experts, who will work with the system largely independently of knowledge engineers; and (e) having realistic expectations about what can be elicited automatically and what requires collaboration with a knowledge engineer.

OntoElicit is a web-based KE system organized as a series of tasks. Methods of progressive disclosure ("as needed" explanation) support domain experts having different levels of experience working with the system. The descriptions below highlight the aspects of medical modeling that are incorporated into OntoElicit. Aspects of modeling requiring live

collaboration with knowledge engineers are described in McShane et al. (2007a, b).

2 PHYSIOLOGICAL MODELING

Our physiological models cover normal physiology, pathology, and the physiological effects of interventions. To make modeling realistic, we model only events and their properties that have known utility in our applications: that is, they must either be part of an important chain of events, measurable by a test, or be able to be changed by a drug, intervention, the effects of another disease, etc. (Cf. the KADS principle that the analysis should include the functionality of the system being developed, which we interpret as the tenet that knowledge should not be elicited or recorded just because we *can* but because we *need to*.)

Modeling Normal Physiology. Physiological scripts are recorded as complex events in the OntoAgent ontology using the formal but still human-readable (after minimal training) formalism shown below.

```
(SWALLOW
  (AGENT HUMAN) (THEME BOLUS)
  (DURATION 10 (DEFAULT-MEASURE SECOND))
  (PRECONDITION
    (LOCATION (DOMAIN BOLUS) (RANGE MOUTH)))
  (EFFECT
    (LOCATION (DOMAIN BOLUS) (RANGE STOMACH)))
  (HAS-EVENT-AS-PART
    OROPHARYNGEAL-PHASE-OF-SWALLOWING
    ESOPHAGEAL-PHASE-OF-SWALLOWING)
  (OROPHARYNGEAL-PHASE-OF-SWALLOWING
  ...
```

This is a small excerpt from the script for swallowing, showing perhaps 1/20 of the entire script. SWALLOW is the ontological concept that heads the swallowing script. The AGENT property of SWALLOW is constrained to HUMAN and the THEME to a BOLUS, which is a small mass of liquid or chewed solid food that is swallowed. The PRECONDITION for SWALLOW is that the BOLUS be located in the MOUTH and the EFFECT is that it is located in the STOMACH. The DURATION of swallowing is 10 seconds. The rest of the script is a hierarchical encoding of fillers of the HAS-EVENT-AS-PART property. The subevents of SWALLOW are OROPHARYNGEAL-PHASE-OF-SWALLOWING and ESOPHAGEAL-PHASE-OF-SWALLOWING, which have their own subevents, which have their own properties, and so on. (Cf. the KADS principle that knowledge should be encoded in an incremental way, meaning breadth-first.)

This example shows only a fraction of the expressive means used in scripts – there are also vari-

able bindings, loops, conditions, etc. However, it suffices to make the main point: the ontological metalanguage provides a conceptual framework for eliciting knowledge about complex events.

In OntoElicit, domain experts are led through the process of creating scripts by first listing the main events of a physiological process, then adding subevents, then adding selected properties of each subevent: THEME, PRECONDITION, EFFECT and select others. They are instructed to begin at a relatively coarse grain-size, since more details can be added as found necessary. (Cf. the KADS principle that new data should be elicited only when collected data has been analyzed.) The domain expert can record the knowledge in prose or his/her own invented semi-formalism. (Cf. the KADS principle that an intermediate level representation should be encoded first.) The use of a fixed inventory of properties of interest, and a structured process for eliciting them, is similar in function to Shadbolt and Burton's (1995) inventory of "linguistic probes" used in a structured interview: "The idea here is that the elicitor engages in a type of slot/filler dialogue".

Modeling Diseases. Diseases are complex events and could, in principle, be modelled using the same type of strategy as just described for normal physiology. However, we have found a different operational metaphor to be useful: tables that track relevant property values over conceptual stages of the disease.

In OntoElicit, domain experts are asked to divide the disease into any number of conceptual stages correlating with important events, findings, symptoms or the divergence of disease paths among patients. They are also asked to indicate the typical duration of each stage as a range (x-y in Table 1) with a default value (d). Next, they are led through the process of describing the physiology, symptoms, test results and results of interventions, should the latter be administered at each stage of the disease. The high-level conceptual model is shown in Table 1. Note that experts are not asked to fill in a table like this all at once: they are led through a well-explained, step-by-step process of providing the component knowledge.

Table 1: KE for the clinical model of a disease.

	Props.	Start	Stage 1	Stage 2
Duration			x-y (d)	x-y (d)
Physiology	P1	x-y (d)	x-y (d)	x-y (d)
	P2	x-y (d)	x-y (d)	x-y (d)
Symptoms	S1	x-y (d)	x-y (d)	x-y (d)
	S2	x-y (d)	x-y (d)	x-y (d)
Test Results	T1	x-y (d)	x-y (d)	x-y (d)
	T2	x-y (d)	x-y (d)	x-y (d)
Interventions	I1	x-y (d)	x-y (d)	x-y (d)
	I2	x-y (d)	x-y (d)	x-y (d)

Tests

List tests relevant for this disease, alternative names for them, if any, and select the properties they measure. If a test measures more than one property, list it on more than one line.

If some property is not in your inventory, please return to the property inventory page and add it there before returning here, but be sure to save this page first.

Test	Alternative Name	Property Measured	Add/Remove Row
Esophageal manometry	Manometry, Esophageal motility study	Basal LES pressure	+ -
Esophageal manometry		Residual LES pressure	+ -
Barium swallow		Diameter of distal esophagus	+ -
		Basal LES pressure	+ -

Figure 1: The first KE screen for tests, partially filled out for a particular disease.

For physiology and symptoms, the expert provides the inventory of properties (Props.) that change over time, their start value before the disease begins and their expected values at end of each conceptual stage. Most values will be recorded as a range of values (x-y) covering different individual patients in the population along with a default (d) representing the most typical value. When value sets are numerical, the values at any point in a stage can be interpolated by the simulation engine.

In the test results section, the expert indicates (a) which physiological properties are measured by each test, (b) any test results that are not among those covered by the physiological model, e.g., visual findings by the administrator of the test and (c) a “specialist’s interpretation” of what the test results returned at that stage would indicate: e.g., “Suggestive of disease X.” (Raw test results for recorded physiological properties can, of course, be provided by the simulation engine.) The first part of this thread of elicitation is shown in Figure 1. The properties measured were recorded earlier and are selected from a pull-down menu.

For interventions, including medications, the expert indicates (a) what properties and/or symptoms are affected by the intervention, (b) the possible outcomes of the intervention, (c) possible side effects, and (d) if known, the percentage of the population expected to have each outcome and side effect.

After the properties and value sets provided by the expert have been translated into the ontological metalanguage, the disease models developed using this strategy are sufficient to support the simulation of diseases as they progress outside of “interventions”, which may be generated internally (as by another disease) or externally (as by the use of medication or surgery). To accommodate the effects of interventions, OntoElicit elicits knowledge in a similar way as for modeling normal physiology – i.e., using scripts – with a focus on the properties PRECONDITION and (immediate) EFFECT. The progres-

sion or regression of the disease during or after an intervention is again recorded using the table-based strategy, often with only slight modifications to the values recorded in the original model. For example, a medication might slow the rate of progression of a disease – affecting the filler of the parameter “stage duration” – but leave everything else the same, or it might relieve symptoms but leave disease progression unchanged, or it might reverse some physiological changes but leave others unaffected.

3 MODELING CLINICAL PRACTICES

One of the principles we follow is to record knowledge in the simplest way possible to support an application. As regards clinical advice giving, our applications use three kinds of recorded knowledge for three functional contexts: checking the validity of a clinical move; advising what to do next in simple, “textbook” contexts; and advising what to do next in complex contexts.

Checking the Validity of a Clinical Move. Our first priority in developing MVP was to build a simulation environment for trial-and-error learning, with the gradual addition of tutoring support. As such, our initial tutoring functionalities were narrow in scope: checking whether each move by the trainee conformed to what we call “preconditions of good practice” and, if not, providing various extents of information (based on user preferences) about why not.

The knowledge needed to support this functionality is readily encoded using the basic slot-filler structures of the OntoAgent ontology. For example, for each disease we record values for the properties SUFFICIENT-GROUNDS-TO-SUSPECT, SUFFICIENT-GROUNDS-TO-DIAGNOSE, SUFFICIENT-GROUNDS-TO-TREAT (e.g., clinical diagnosis or definitive diagno-

sis), etc.. Similar inventories of properties are used for tests, treatments, making definitive diagnoses, and so on. The content of this knowledge is both broader and deeper than that available in published “best practices” guides. OntoElicit uses tables for eliciting this information, with the experts providing prose descriptions of property fillers. These descriptions are then converted – like all other aspects of acquired knowledge – into formal, ontologically grounded structures by knowledge engineers and programmers. Table 2 shows a combination of elicited knowledge (with a clear background) and formally encoded knowledge (with a gray background) for one factoid about one disease.

Table 2: Sample precondition of good practice.

DISEASE	ACHALASIA
PROPERTY	SUFFICIENT-GROUNDS-TO-SUSPECT
Prose filler	solid and liquid dysphagia or regurgitation
Formal encoding	(or (and ((SOLIDS-STICK HUMAN YES) (LIQUIDS-STICK HUMAN YES)) (REGURGITATION-FREQUENCY HUMAN (> 0))))

One of the advantages of recording all knowledge using the same ontological metalanguage is that knowledge can be reused in both immediately obvious and to-be-discovered ways. For example, imagine that a trainee using MVP wants to posit a diagnosis, but is told by the tutor that before doing so the value of Property P must be known to be $> = x$. Imagine further that the trainee does not know which tests determine that property value. Using knowledge already recorded in the ontology, the system can look up which property values are measured by each test and return those tests that measure the needed property value.

Since each agent in our environment has a different ontology (knowledge base of object and event types and their relationships) and fact repository (knowledge base of ontological instances and their relationships), MVP can contain multiple tutors with different opinions about best practices all residing in the system and used as selected by a teacher or trainee.

Advising What to do Next in Simple and Complex Contexts. Another type of clinical advice that is central to CLAD and will soon be incorporated into MVP is *What to do next?* In the simplest case, a single answer can be arrived at using conditions recorded in the precondition and effect slots of ontological scripts. This kind of knowledge can often be found in textbooks, sometimes even in a decision tree representation. In OntoElicit, recording this

kind of knowledge is supported using the script writing methodology described above, with an emphasis on conditional statements.

However, many clinical moves must be decided upon (a) in the face of competing conditions, (b) with different preferences of different “stakeholders” (e.g., the patient, the physician, the insurance company) and (c) using incomplete knowledge of relevant property values. For those cases, we have been experimenting with the use of Bayesian networks that are constructed with the help of influence diagrams (For more on influence diagrams, see Howard and Matheson (2005); for an example of their use in another medical domain, see Lucas (1996); and for more details about our work using them, see McShane et al. (Submitted).) The knowledge encoded in influence diagrams represents an expert’s opinion about the utility scores of different combinations of property values associated with each possible decision. One of the main reasons why we chose to work with influence diagrams is that the kind of information required of experts is of a nature that they can readily conceptualize. In essence, they are asked: *Given X combination of property values, how good is solution Y? Given X combination of property values, how good is solution Z?* and so on. The properties and values are familiar to our experts because they are the same ones used to build the other models in the system. We are using the Netica (<http://www.norsys.com/>) environment to create influence diagrams. Knowledge engineers help experts to organize the problem space into subproblems, as applicable, and to develop a case-specific methodology of filling out the utility tables in the most efficient way.

Although the nature of information required of experts in an influence-diagram-driven methodology is straightforward, one problem is that the number of features involved in making a complex decision can be large, easily driving the number of feature value permutations into the tens or hundreds of thousands. As in all aspects of our modeling, we approach this problem using realistic strategies including the following: (1) Organizing the knowledge optimally – e.g., covering as many variables as possible using local decisions whose output contributes to a more general decision; (2) Simplifying the problem space and judging if the results are sufficient to yield realistic, accurate functioning – e.g., not including every parameter we can think of but, instead, focusing on those considered to have the most impact by clinicians; (3) Working toward automating the process of knowledge acquisition – e.g., using functions to provide values for many of the feature-value combina-

tions once a pattern of utility scores has been recognized. (For other issues related to reducing the complexity of knowledge acquisition of influence diagrams see Bielza, Gomez and Shenoy (2010).) Following these tenets, we successfully configured our first demonstration system of CLAD.

As regards incorporating aspects of influence diagram creation into OntoElicit, our current thinking is that experts could, in fact, be led through the process of decomposing the problem into the main variables in the decision vs. the variables in local decisions (cf. point (1) above), but we have yet to experiment with this methodology.

4 CONCLUSIONS

In this paper, we have provided a sweeping introduction to some of the different kinds of modeling strategies used within the OntoAgent environment. We have shown how our problem space design has facilitated the creation of a mixed-initiative KE system for encoding clinical knowledge in the metalanguage of the OntoAgent environment. One of the advantages of our modeling strategies is that the knowledge is formulated such that it can be understood not only by the expert him- or herself but also by the wider community, as illustrated in Jarrell et al. (2008). (Cf. the KADS principle that collected data and analysis should be documented.) Although we are well aware that such general strategies will not be sufficient to overcome all modeling challenges, we believe that they are beneficial in helping experts to conceptualize domains quickly, independently and in the most practical way. In this sense we believe that this work makes a contribution to overcoming the knowledge bottleneck in constructing practical knowledge-based systems.

REFERENCES

- Beale, S., Lavoie, B., McShane, M., Nirenburg, S., Korelsky, T., 2004. Question answering using Ontological Semantics. In *Proceedings of ACL-2004 Workshop on Text Meaning and Interpretation*.
- Bielza, C., Gomez, M., Shenoy, P. P., 2010. Modeling challenges with influence diagrams: Constructing probability and utility models. In *Decision Support Systems* (in press).
- Breuker, J. (Ed.), 1987. Model-driven knowledge acquisition interpretation models. Deliverable task AI, Esprit Project 1098 (University of Amsterdam).
- Cooke, N.J. (no date), Knowledge elicitation. Available at <http://www.cerici.org/documents/Publications/Durso%20chapter%20on%20KE.pdf>
- Ford, D. N., Sterman, J. D., 1998. Expert knowledge elicitation to improve formal and mental models. In *Syst. Dyn. Rev.* 14: 309-340.
- Hoffman, R. A., Lintern, G., 2006. Eliciting and representing the knowledge of experts. In Ericsson, K. A., Charness, N., Feltovich, P., Hoffman, R. (Eds.), *Cambridge Handbook of Expertise and Expert Performance*, pp. 203-222. New York, Cambridge University Press.
- Howard, R. A., Matheson, J. E., 2005. Influence diagrams. In *Decision Anal.* 2(3): 127-143.
- 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*.
- Lucas, P., 1996. Knowledge acquisition for decision-theoretic expert systems. In *AISB Quarterly*, 94: 23-33.
- McShane, M., Nirenburg, S., 2003. Parameterizing and eliciting text elements across languages. In *Machine Translation* 18(2): 129-165.
- McShane, M., Nirenburg, S., Beale, S., 2005. Semantics-based resolution of fragments and underspecified structures. In *Traitement Automatique des Langues* 46(1): 163-184.
- McShane, M., Fantry, G., Beale, S., Nirenburg, S., Jarrell, B., 2007a. Disease interactions in cognitive simulations for medical training. In *Proceedings of MODSIM World Conference, Medical Track*.
- McShane, M., Nirenburg, S., Beale, S., Jarrell, B., Fantry, G., 2007b. Knowledge-based modeling and simulation of diseases with highly differentiated clinical manifestations. In *Proceedings of the 11th Conference on Artificial Intelligence in Medicine*.
- McShane, M., Nirenburg, S., Beale, S., Catizone, R., Submitted. A cognitive architecture for simulating bodies and minds. Submitted to *ICAART-2011*.
- Nirenburg, S., McShane, M., Beale, S., 2008a. A simulated physiological/cognitive "double agent". In *Proceedings of the Workshop on Naturally Inspired Cognitive Architectures* at AAAI 2008 Fall Symposium.
- Nirenburg, S., McShane, M., Beale, S., Jarrell, B., 2008b. Adaptivity in a multi-agent clinical simulation system. In *Proceedings of AKRR'08 - International and Interdisciplinary Conference on Adaptive Knowledge Representation and Reasoning*.
- Nirenburg, S., Raskin, V., 2004. *Ontological semantics*. Cambridge, Mass., The MIT Press.
- Schank, R., Abelson, R., 1977. *Scripts, plans, goals and understanding*. Hillsdale, NJ, Erlbaum.
- Shadbolt, N., Burton, M., 1995. Knowledge elicitation. In Corlett, E.N., Wilson, J.R., (Eds.), *Evaluation of human work*. CRC Press.