HARNESSING CYC TO ANSWER CLINICAL RESEARCHERS' *AD HOC* QUERIES

Douglas Lenat, Ph.D., Cycorp doug@cyc.com
Michael Witbrock, Ph.D., Cycorp witbrock@cyc.com
David Baxter, Ph.D., Cycorp baxter@cyc.com
Eugene Blackstone, M.D., Cleveland Clinic Foundation blackse@ccf.org
Chris Deaton, Cycorp cdeaton@cyc.com
Dave Schneider, Ph.D., Cycorp daves@cyc.com
Jerry Scott, Research Intelligence jdscott@clearwire.net
Blake Shepard, Ph.D., Cycorp blake@cyc.com

Abstract. By extending Cyc's ontology and KB approximately 2%, Cycorp and Cleveland Clinic Foundation (CCF) have built a system to answer clinical researchers' *ad hoc* queries. The query may be long and complex, hence only partially understood at first, parsed into a set of CycL (higher-order logic) fragments with open variables. But, surprisingly often, after applying various constraints (medical domain knowledge, common sense, discourse pragmatics, syntax), there is only one single way to fit those fragments together, one semantically meaningful formal query *P*. The system, SRA (for Semantic Research Assistant), dispatches a series of database calls and then combines, logically and arithmetically, their results into answers to *P*. Seeing the first few answers stream back, the user may realize that they need to abort, modify, and re-ask their query. Even *before* they push ASK, just knowing approximately how many answers *would* be returned can spark such editing. Besides real-time *ad hoc* query-answering, queries can be bundled and persist over time. One bundle of 275 queries is rerun quarterly by CCF to produce the procedures and outcomes data it needs to report to STS (Society of Thoracic Surgeons, an external hospital accreditation and ranking body); another bundle covers ACC (American College of Cardiology) reporting. Until full articulation/answering of precise, analytical queries becomes as straight-forward and ubiquitous as text search, even partial understanding of a query empowers *semantic search* over semi-structured data (ontology-tagged text), avoiding many of the false positives and false negatives that standard text searching suffers from.

AI Magazine, Volume 31, Number 3, Fall, 2010

1. INTRODUCTION

Artificial intelligence systems are increasingly capable of doing the inference required to flexibly answer queries, and an increasing amount of data is becoming available in forms that support such inference (Lehmann, Schüppel and Auer 2007). Current successes in the area of knowledge capture promise a rapid increase in such formally represented data, and a large scale knowledge base such as Cyc (Lenat and Guha 1989, Matuszek et al. 2006) which contains appropriate background knowledge (domain knowledge and general knowledge) supports *semantically integrating* that data to answer queries. A substantial barrier to the widespread use of these systems is *query formulation:* getting the system to correctly understand what the user is trying to ask.

In previous knowledge stores (e.g., relational databases), fixed data schemata supported the skilled construction of fixed formal queries, often embedded directly in application program code and expressed in unambiguous query languages such as SQL. At the same time, the small number of relations in these data bases made them comprehensible, allowing query construction — by SQL-fluent programmers or by end users via a custom query-construction application for that database — after a fairly short training period.

Querying *knowledge* bases, even those with weak inferential support such as the current generation of RDF triple stores, is an entirely different matter. With a potential relational- and type-vocabulary in the millions of terms, users need much more support in constructing even straightforward queries. And when the query language itself is more expressive – supporting, e.g., nested logical quantifiers and temporal and modal operators – the need to support users in correctly articulating their intended query is even more dramatic. This paper describes progress we have made in developing such a query articulation assistant, and how we are applying it in the domain of healthcare.

Clinical researchers — and clinicians — need to pose queries that are quite long and convoluted. To further complicate matters, patient health records and procedure notes are generally fragmented across many different, large, stove-piped databases and knowledge stores, especially where those records cross hospital departments and cross decades of time. Cycorp and Cleveland Clinic Foundation (CCF) have built an *ad hoc* query answering application called SRA (for Semantic Research Assistant), based on Cyc (Lenat and Guha, 1989). A physician types a query in English to SRA. Then, working together in English, they translate it into a logically equivalent unambiguous predicate calculus form *P* from which Cyc then designs and executes appropriate database calls. SRA displays answers as they stream back, and can give symbolic rationales justifying each, bottoming out in general medical facts (with provenance), expert-articulated rules, specific patient records, contemporaneous operation notes, etc.

Preliminary results are encouraging: SRA is now used to ask each clinical research query involving cardiothoracic surgery, cardiac catheterization, and percutaneous coronary intervention. Prior to SRA, approximately 300 new queries in those domains had been posed and answered each year, with most queries requiring 1-10 weeks (occasionally several tens of weeks) of real time to be answered to the physician's satisfaction; in 2010, using SRA, such

queries take 5-50 *minutes* to produce satisfactory answers (occasionally several hours), and over 2,000 queries are processed each *week*. Some of that large throughput is due to the fact that persistent bundles of queries in those domains are re-run each month (for internal quality testing purposes) and quarterly (for external third-party reporting purposes): e.g., one bundle of 275 queries produces the procedures and outcomes data CCF needs to report to STS (the Society of Thoracic Surgeons, a hospital accreditation and ranking body), and a bundle of 256 queries produces the data CCF needs to report to ACC (the American College of Cardiology).

This same approach has also been applied, in virtually unchanged form, to support queries against a terrorism knowledge base (Deaton et al. 2005), corporate financial data, and wireless network activity (Fortuna et al. 2009); we call that domain-independent portion of SRA "CAE" for "Cyc Analytic Environment" (Siegel et al. 2005). It is supported by systems for knowledge capture which, again, do not require knowledge of the underlying representational target (Schneider et al. 2005). Text search is ubiquitous and useful today, thanks to Google and its predecessors, despite the high frequency of false positives and false negatives and the shallowness of inference being performed (due to lack of understanding of the query and lack of understanding of the text corpora being queried against.) Our long term goal for CAE is to make the precise articulation (and answering) of analytical queries over multiple knowledge sources almost as straight-forward for end users, almost as useful, and through that path almost as ubiquitous as text search is today.

2. THE CHALLENGE

Clinicians and clinical researchers often want to pose *ad hoc* queries, such as:

> Q1: "Are there cases in the last decade where patients had pericardial aortic valves inserted in the reverse position, to serve as mitral valve replacements, and how often in such cases did endocarditis or tricuspid valve infection develop, and how long after the procedure?"

The *researcher* here is looking for patient cohorts for clinical trials worth proposing and undertaking — in this case, e.g., investigating whether there are unusually high (or low) risks of infection by using pAV prostheses in ways they were most definitely not designed for, and whether there have been enough cases for a trial (to which the answer is *No*, for the data bases of hundreds of thousands of CCF patients treated over the past 20 years – there have not yet been enough cases for a trial.)

A *clinician* might ask the very same *ad hoc* query when looking for assistance choosing among treatment options — for example, if their patient is a young female addict with an extremely small mitral valve annulus and a history of repeated episodes of tricuspid valve infection. The clinician could issue this query, knowing that aortic valves come in smaller sizes than mitral prostheses, and because they remember reading something (Cardarelli et al. 2005) about pAV's (pericardial aortic valve prostheses) being unusually resistant to infection and anticoagulation compared to mitral valve prostheses. Here the answer is *Yes:* that usage of pAVs is rare but definitely not unprecedented.

The Cleveland Clinic Foundation (CCF) is one of the leading medical research institutions in the world: clinical researchers formulate hypotheses and ask *ad hoc* queries about the hundreds of thousands of patients whose records have been painstakingly maintained over decades (Kaple et al. 2008, Mihaljevic et al. 2008, Koch et al. 2008, Hoercher et al. 2008, Gillinov et al. 2008, Sabik et al. 2008, Hickey et al. 2008). And yet, even at CCF, getting an *ad hoc* query answered has been a long and convoluted process, of consultation with multiple intermediaries some of whom are familiar with the underlying medicine and some of whom are familiar with the available data bases and registries. Often a back-and-forth clarification dialogue occurs between the researcher and the medically-trained intermediary: "*What exactly does 'isolated procedure' mean in your query?*" "*When you say 'recently', how long ago do you mean to include*?". A second intermediary, a data base access specialist (DBA), transforms the resulting specification into an actual SQL or SPARQL query, does the "data pull", and sends the results back to intermediary#1, who sends them back to the physician. Often further back-and-forth dialogue occurs between the two intermediaries, occasionally requiring intermediary#1 to go back to the physician for some further clarification. It is not uncommon for this entire process to iterate several times, as the query is refined: the email logs tracking 900 of these queries over the last few years at CCF show a mean time for this process to complete of approximately *one month* of real time, effectively limiting researchers to about a dozen such queries per year.

Our aim with SRA is to enable physicians to pose their complex *ad hoc* questions *directly*, getting them understood and answered in four minutes rather than four weeks. Clinical *researchers* might explore what today is a typical year's worth of hypotheses in one afternoon, and *clinicians* — who today cannot even consider asking *ad hoc* queries relevant to a particular patient – could perform an individually tailored outcome analysis in real-time for that patient. As health-care providers move towards ubiquitous adoption of electronic patient records, the power of such data-driven clinical practice will only increase.

Although the application presented in this paper, SRA, is focused on medical research, similarly complex *ad* hoc queries, and similarly convoluted data acquisition and aggregation processes, occur in many other domains. A similar iterative query-articulation process, but with human research librarians as intermediaries, was once the standard (Lang, Tracy and Hepburn 1957) in many fields.

Why was it that, until SRA, neither the clinician nor the clinical researcher could expect to have *ad hoc* queries like Q1, above, answered in minutes instead of weeks? Partly it is because of the many, and significant, AI challenges which have stood between the enquirer and a deep understanding of their query:

Challenge 1: Getting the literal query understood: converting it from highly ambiguous natural language to an unambiguous logical form. Typical queries such as those found on NIH's clinicaltrials.gov website are likely to contain numerous inclusion and exclusion criteria; 100- and 200-word queries are common[1]. But the state of the art of natural language parsing today cannot reliably parse even shorter *ad hoc* queries such as Q1 into a precise, unambiguous logical or data base query language representation.

[1] For example, http://clinicaltrials.gov/ct2/show/NCT01030328?term=pav&rank=4 takes 258 words just to state its inclusion and exclusion criteria.

Challenge 2: Getting the intended query understood. Often the physician will leave off some obvious clauses and details: temporal, spatial, causal constraints, equality or inequality constraints, etc. For example, in Q1, the physician might mean "*...patients at this medical center*", and/or "*...aortic valves with the type and manufacturer we have in stock now*", and/or "*...ignoring cases where the endocarditis developed more than a year after the procedure*", and/or "*...in which the patient survived at least 6 months post-procedure*".

Challenge 3: Given a complete, unambiguous, logical form of the intended query, finding the answer to that query. This involves identifying the relevant rules and algorithms that will serve as an acceptable basis for computing an answer to that query; deciding which of many (inevitably heterogeneous) data bases and other structured information sources to retrieve information from; actually gathering the relevant data from those sources; and, finally, carrying out the computations and reasoning steps to produce the answer.

- At an infrastructure level, this means worrying about protocols and channels to access the *n* information sources, dispatching the *m* different low-level SQL or SPARQL or other API atomic queries, combining the sub-queries' answers, etc.
- At a higher level, this means being able to formulate a complex plan for efficiently asking those *n* data sources those *m* atomic queries. For each atomic query, there may be additional reasoning required to plan, e.g., the best order of conjuncts. [2]

Challenge 4: Present the answers to the physician in a *useful fashion*. This utility derives from presenting data in a clear on-screen layout, and in a timely fashion; what "useful" means may change from user to user, situation to situation (e.g., if they are faced with a critical realtime decision), and query to query.

- SRA explicitly reasons about presentation, transforming the underlying logical data into human-interpretable form – for example, choosing appropriate rows and columns, and appropriate row and column *headers*, for a matrix of answers which it then presents to the user in the form of a table. Furthermore, the contents of an individual cell in that table are converted from the formal, and often idiosyncratically coded, language returned by the information sources into something that will be meaningful to the physician. To take an extreme example, a cell displaying as "#bnode-50943" would mean nothing to the physician, compared to the form produced by SRA's use of Cyc Natural Language Generation: "The CABG+MVA performed at CCF by Dr. Joshua Stuyvesant at 8am on March 3, 2007".) [3]
- A second aspect of "useful fashion" here refers to *temporal* presentation as well: if there are going to be 4718 cases matching the criteria, it can be much better to start streaming

---

[2] Although SQL optimization is standard in relational data base systems today, an increasing amount of medical data is represented in the newer RDF/OWL semantic triple store systems accessible by SPARQL queries, for which such optimization has not yet become available, resulting in queries taking orders of magnitude too long. We expect this problem to solve itself in the next five years, as commercial SPARQL optimization catches up with SQL optimization.

[3] For HIPAA reasons, these and other instance-level healthcare data presented in this article, are anonymized references to fictional patients and events.

a few of them in every second, rather than waiting 4 minutes and then displaying them all at once. Not only are users impatient, they often can spot "mistakes" in the first few answers returned, e.g. due to a clause they omitted – after which they would just abort the query, revise it, and re-ask it.

- A third component of what is meant here by "useful fashion" is to properly integrate and organize information coming from several different sources, placing those pieces down to form a coherent mosaic picture of the patient as a whole. E.g., given the cities and time-stamps on a large number of disparate elements of this patient's data, arrange them into a single chronology of where this patient resided and for how long.
- A fourth component of "useful" here refers to assessing the quality, certainty, and relevance of the answers, and then sorting or filtering or annotating the answers based on that assessment

Challenge 5: In cases where the system would otherwise fail to return an answer, it should "fail soft": i.e., provide some form of *semantic search* results, drawing from available texts in unstructured prose (or almost unstructured form, e.g., free text that has been tagged with terms from an ontology). That means fetching existing documents — recent literature, Web pages, internal reports — relevant to the user's query. The challenge is to produce higher retrieval accuracy than keyword-based search engines, by drawing on general knowledge, medical knowledge, discourse knowledge, and context, to avoid false positive inclusions and false negative omissions.[4]

3. MEETING THE CHALLENGE

3.1. From the user's point of view

In meeting this challenge, SRA implements a query-handling workflow illustrated in Fig. 1, presented via the interface shown in Fig. 2.

The numbers 1-4 in the red circles on Figures 1 and 2 correspond to each other, and also correspond to the paragraph numbers **1-4** on the next few pages, explaining the workflow:

1. The user types in an English query. Since accurate parsing of complex medical queries to precise logical representations is well beyond the state of the art, the main process used is an interactive clarification dialogue between the system and the user (see 2, below). The system reliably identifies concepts in the query like "AVR" and "left atrial enlargement", and uses the Cyc semantics of those concepts to identify simple temporal, spatial, and role relationships, which are used to construct candidate components for a predicate calculus query. Some of these

[4] Although fail-soft capabilities were implemented in the CAE, on which SRA is based, and have been applied experimentally to the use of outcome data in end-user search (see Section 3.2.4 below), they have not been integrated deeply into the SRA's initial research cohort selection application.

components have open variables that will be used in connecting the components together into a complete query. Even at this point, learned knowledge (a trained decision tree) and background knowledge from the KB have been used to filter the possible fragments into a manageable set with a high likelihood of expressing the user's intent.

2. Each fragment is represented in predicate calculus, internally, but what the user of the system sees is a paraphrase of each fragment back into English as a set of fill-in-the-blank *fragment* phrases, where the blanks represent variables (e.g., "*pericardial valve model* _?x_ *was implanted*"). Another of the fragments listed in Figure 2 is "the *patient ID* is __"; this is a straightforward example of inferring what the user *intended* to say but didn't literally say (see Challenge#2, above), since most user complete queries end up with a column in the answer table containing CCF patient id numbers, the system infers the need for such a query fragment. The user highlights the fragments representing parts of the query they had in mind, and tells the system to combine them.

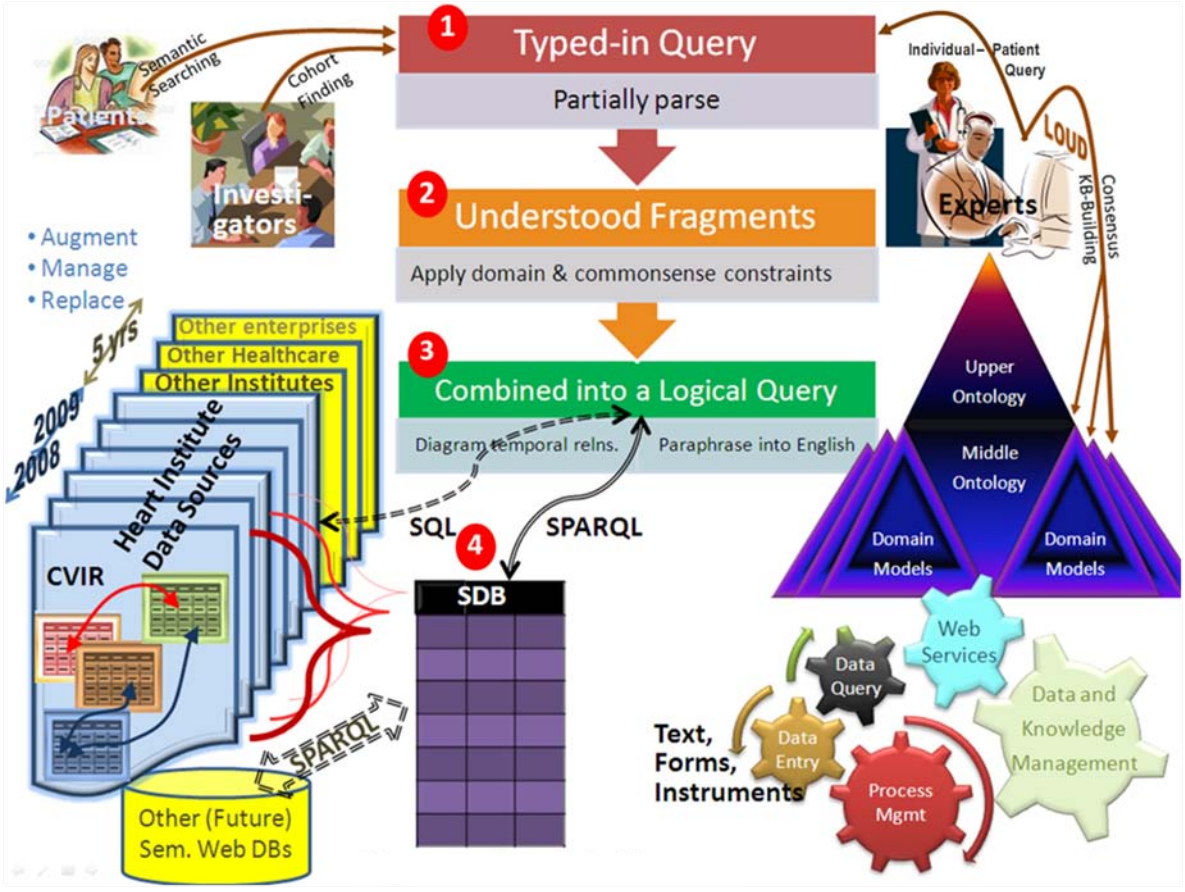

Figure 1: SRA interacts with clinical researchers in English to build and execute precise logical queries against multiple knowledge sources. It uses the Cyc ontology and Cyc inference, data access, and natural language components to support query building, knowledge federation, and answer presentation. The same capabilities can be used to support collaborative KB building. The numbers 1-4 in the red circles refer to more detailed discussion in the text, and also correspond to the like-numbered UI components in Fig 2.

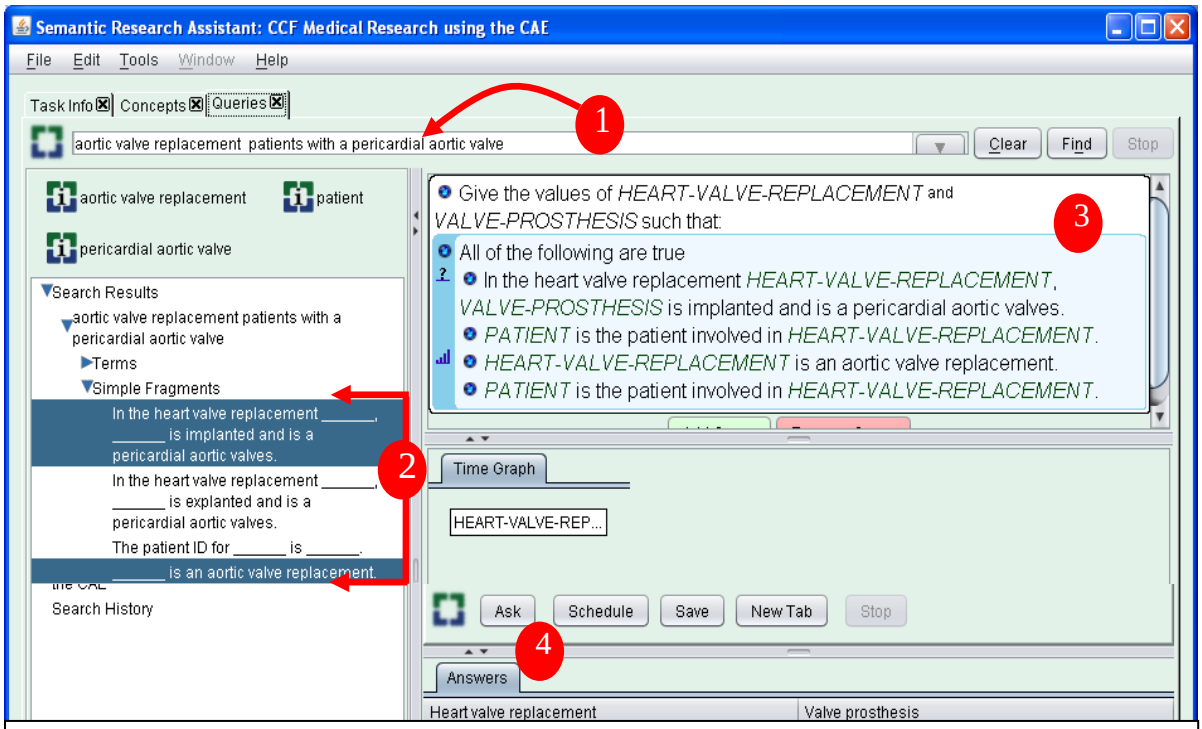

Figure 2. The SRA (Semantic Research Assistant) system in use, building a formal understanding of the user's intended query (for a cohort of patients).

3. It is not a simple matter to combine a large number of fragments, often with two or more free variables, into a single correct n^{th}-order predicate calculus query. The huge conceptual vocabulary from which the fragments have been selected makes the problem especially difficult, since it would be impractical[5] to construct the corresponding set of hard-wired combination rules. SRA brings the entire Cyc knowledge base and inference engine to bear in support of the combination process. Common sense, discourse pragmatics, context, medical knowledge, syntax, etc. all come into play. At a predicate calculus level, two of the most common and most important decisions being made are: (a) which variables unify with which other variables (i.e., refer to the *same thing*)? and (b) what is the *type* of each quantifier (universal or existential) and the scope/nesting of the quantifiers? In this case, e.g., the variables might include the patient, the surgeon, the valve replacement procedure, the valve that is implanted, the date/time of the procedure, etc. Common sense enables Cyc to conclude that the patient and surgeon are distinct variables, and also enables it to determine that the valve and the implanting are distinct variables. Discourse and domain knowledge enable it to infer that "the patient" refers to a single individual, within the query, as otherwise it would be absurdly productive (lead to a vast number of unrelated answers). By leveraging the enormous existing Cyc KB (Fig 3), it was only necessary to add the specifics for this project: for example, that AVRs are surgical procedures, and that pericardial aortic valves are medical implants[6]. The former generalizes in Cyc's ontology to *event*, and the latter generalizes to *tangible object*, and Cyc has, since 1985, understood the sort of disjointness between those collections (Lenat and Guha 1989) which in

[5] The cardinality of such a set would exceed the number of atoms in the universe.

[6] The Cyc term MedicalCareEvent was created fifteen years earlier, on Jan 24th 1996. The Cyc term Implant-Medical was created on Aug 26th 1999 and had additional assertions added in 2001, '02, '03, '04, '06, '07, '08, and '09.

turn entails that different variables must represent these two concepts all the way through to the combined query. By contrast, a *patient* is known to be a human being, which is exactly of the correct type to play the role "recipient of service" in a service event such as a surgical procedure. Therefore, only one variable is needed to represent the CCF patient (who necessarily has some CCF id number) and the recipient of the AVR procedure. If the user now adds a clause about the *primary surgeon*, Cyc uses medical knowledge to infer that the patient is not the surgeon.

08/26/1999

11:14:15 (genls MedicalDrugImplant Implant-Medical) by Meyer

11:14:15 (denotation Implant-TheWord CountNoun 0 Implant-Medical) by Meyer

11:14:15 (genls Implant-Medical SinglePurposeDevice) by Meyer

05/13/2007

11:19:18 (genls
(CollectionIntersection2Fn CardiacValveProsthesis-Biological-Pericardial AorticValveProsthesis) Implant-Medical)

Figure 3: This assertion about pericardial aortic valves (representing the fact that they are medical implants) was added in 2007, as a simple subset (genls) ground atomic formula. This assertion allows Cyc to infer many things about pAV's, such as the fact that -- drawing on years-old Cyc assertions – the valves should not in general be treated as events or people.

4. The user clicks **ASK**, and the SRA system makes use of Cyc background and domain knowledge, together with meta-knowledge about the CCF data bases, to produce the appropriate SPARQL and/or SQL query or queries, dispatch them to the appropriate data bases, and then arithmetically and/or logically combine the results into an answer table (this general capability, which we call Semantic Knowledge Source Integration (SKSI) (Masters and Güngördü 2003). Because these results are returned from inference as logical symbols, which range from nearly incomprehensible to completely incomprehensible, Cyc's NLG (natural language generator) (Coppock and Baxter 2009, Baxter et al. 2005) is used to render table entries comprehensible. For the simple query shown, 1132 answers were found.

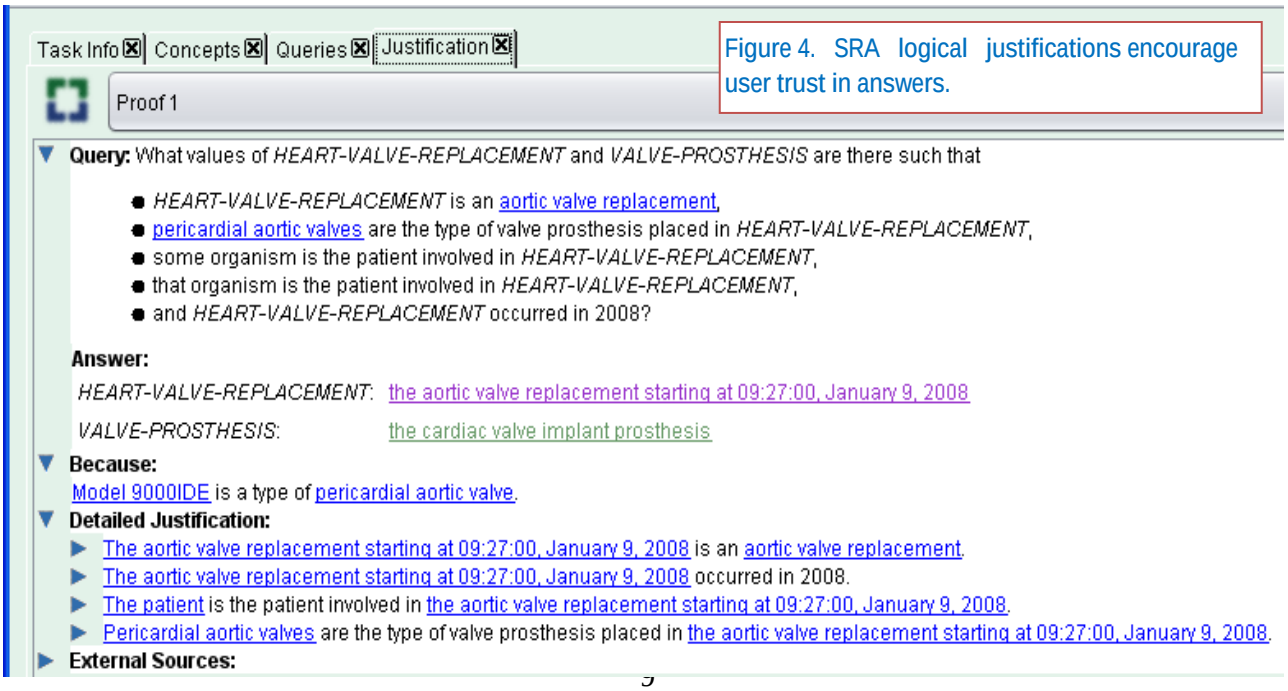

Figure 4. SRA logical justifications encourage user trust in answers.

Figure 4 illustrates how the user can click on an answer to display the logical "proof" that led SRA to it, rendered as a natural language argument (Baxter et al. 2005). The data store being queried did not represent this device as a pericardial aortic valve, but as a Model9000IDE; Cyc provides the background knowledge that each 9000IDE is a pericardial aortic valve prosthesis, and (from its ontology of processes) that an implantation of an aortic valve prosthesis is a replacement of the patient's aortic valve with that prosthesis, and so on.

Such small "impedance mismatches" between the way the query is stated and the way the various data base schemata carve up and represent the data are pervasive; they are part of what makes this a challenging problem. E.g.:

- The physician's query asks for "...mild valve regurgitation..." but the data base represents this as "valve_regurg 1+".
- The physician asks for "isolated CABGs" but the data base merely contains a set of primitive properties from which one could infer which procedures were isolated and which were not isolated.
- The physician refers to patients with "left atrial enlargement" but the data base stores the left atrium diameter in centimeters and medical knowledge must be brought to bear to decide which patients do and don't fall into that category (in this case, the Cyc KB has one rule that says that adult males fall into that category if their left atrial diameter exceeds 4.2cm, and another rule that says that for adult females the cutoff is 3.8cm.)

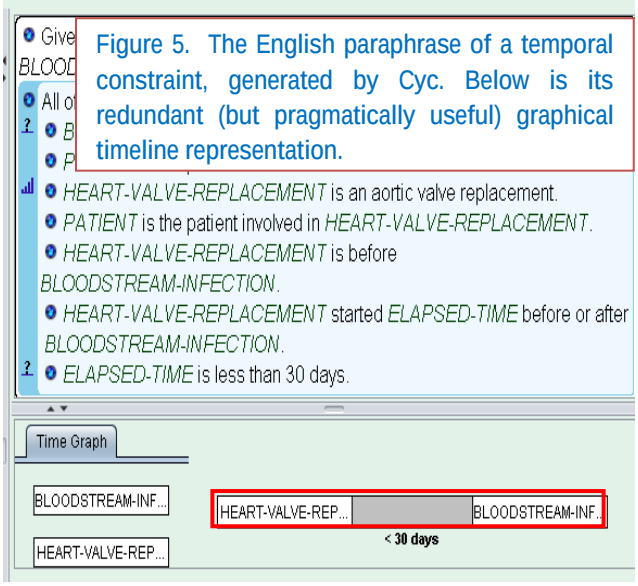

Figure 5. The English paraphrase of a temporal constraint, generated by Cyc. Below is its redundant (but pragmatically useful) graphical timeline representation.

This illustrates a partial realization of the promise of AI systems, in this case the use of inference to flexibly apply knowledge to solving novel problems. By representing the *meaning* of the medical terms, and the *meaning* of each data base's schema elements, it is possible for Cyc to reach similar conclusions about how data should be connected and therefore find the same answers as collaborating human experts with medical and database skills.

Although SRA enables users to formulate their queries using English, it also takes advantage of the fact that it's a computer communicating via a GUI. It turns out that users have a difficult time keeping temporal constraints straight, if they are presented as English phrases; doing so is much easier when they are also drawn graphically. The "Time Graph" (Fig. 5, Fig. 6) visually depicts one or more timelines, and events can be placed in relative or absolute positions on those time-lines. Again, the underlying representation is predicate calculus, so the time line and English representations of the queries are automatically kept consistent. The query in Fig. 5 concerns patients who had septicemia or bacteremia less than a month after an AVR; the 3-box Time Graph timeline clarifies (and is equivalent to) the more confusing final five lines of the textual paraphrase of the query.

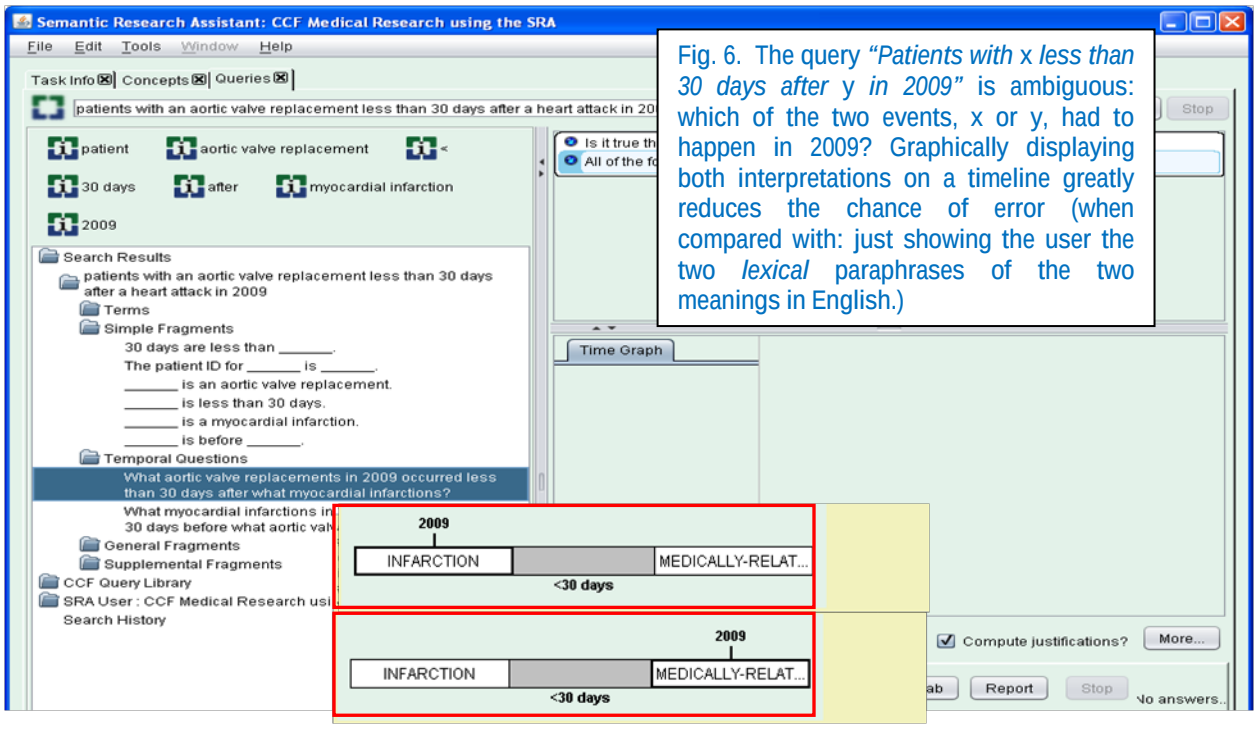

Fig. 6. The query *"Patients with* x *less than 30 days after* y *in 2009"* is ambiguous: which of the two events, x or y, had to happen in 2009? Graphically displaying both interpretations on a timeline greatly reduces the chance of error (when compared with: just showing the user the two *lexical* paraphrases of the two meanings in English.)

Both the Time Graph and the textual paraphrase of the combined query (labeled "3" on Fig. 2) are dynamic; a user can interactively modify, extend, and "explore" them. A context menu on "aortic valve replacement", e.g., displays the ontology of broader, narrower, and related terms, from which the user might select a replacement. The small "cellphone-reception-bars" icon on Fig. 2 indicates how many answers that part of the query is likely to generate, if asked in its present form. Often the user can tell from the presence of too many, or too few, "reception bars", that they must not have finished correctly articulating their query.

A reader might wonder whether, and how, the full KB and inference system of Cyc is required for this task. To address this, we metered the SRA system's use of pre-existing Cyc knowledge (that is, assertions entered into Cyc before our collaboration with CCF started in 2007). We certainly expected *some* re-use, but were surprised to find empirically that *hundreds* of pre-existing pieces of prior and tacit knowledge in Cyc were used *for each ad hoc query*. Cyc KB content was used during each step: interpreting the literal meaning, inferring the intended meaning, carrying out the clarification interaction with the user, putting the fragments together into a meaningful integrated whole, coming up with a plan for answering the query by going out to data bases, optimizing each DB query dispatched, and deciding how best to display the Answers to the user. While there are certainly parts of the Cyc KB that are unlikely to be used in the medical domain (facts entered for a historiography thesis about Merovingian France, for example), the scale of re-use suggests that identifying the reusable elements in advance, and constructing them afresh for each new application would be a difficult and expensive proposition. Having designed Cyc for broad re-use, all those years ago (Lenat *et al* 1983; Lenat and Guha 1989; Lenat 1995) is now paying off. In domains where the user is likely to inject

metaphors and analogies into their queries, even the more esoteric regions of Cyc knowledge-space may turn out to be useful for understanding the intent of their query.

3.2. SRA as Natural Language Technology

Our emphasis in designing the SRA, and the CAE more generally, has been on supplying a usable, responsive, and predictable user experience. We have therefore avoided the use of the most sophisticated parsing techniques available in the Cyc platform and elsewhere in NLP research (e.g. Klein and Manning 2003, Kaplan et al. 2004); while they have the potential to produce interpretations of longer spans of the input text than current, lexical-semantics-based technique, they do not do so consistently enough and rapidly enough for a predictable user experience. Moreover, the relatively technical nature of medical queries, which are not generally highly ambiguous at the *lexical* level, makes them well suited for a shallower approach based around identifying semantic terms used in the query. The shallow semantic interpretation in SRA has been augmented with specific parser for important common relations such as temporal constraints. Interpretation, then, depends on dealing with the limited lexical ambiguity that does exist, and dealing comprehensively with ubiquitous syntactic ambiguity. This includes producing a manageable set of alternatives from which the user may indicate component elements for a final query. The Cyc natural language generation (NLG) system is relied on particularly heavily in this assembly process, both to present candidate fragments for user selection, and to generate a clear reflection of the overall query under construction. NLG is also used for presentation, translating table headers and cell entries into user-comprehensible form, and to foster user trust by providing a facility to review system-generated justifications of its answers (Fig. 4). The next sections provide a little more detail.

3.2.1. Term interpretation and filtering: First, a user query is scanned for single or multi-word terms that are known to Cyc. Coverage is already high (around 24% of the 126,000 most accessed Wikipedia pages from a typical hour had a corresponding existing Cyc concept).; for domains for which custom knowledge represenation has been done (such as cardiothoracic surgery, in the SRA), term coverage is nearly complete. Readers can experiment with a slightly limited version of this lexical lookup by using the "find" web-service exposed at **ws.opencyc.org**[7]. This phrase lookup produces a set of candidate interpretations, which are then filtered using a decision tree trained for the domain, which eliminates domain-improbable senses. A portion of the tree for SRA is shown in Figure 7, along with an example from another domain; because both the training and use of these trees take advantage of the Cyc ontology, they can make decisions at a general level (e.g. OrganismPart, MedicalEvent). This enormously reduces the number of training examples that must be used; the SRA filter was initially trained, for example, by automatically tagging and then manually annotating the relevance of the concepts found in a mere 29 example query sentences.

[7] The query http://ws.opencyc.org/webservices/concept/find?str=surgery will, for example, return an XML document identifying the URI http://sw.opencyc.org/concept/Mx4rvViynJwpEbGdrcN5Y29ycA which is the OpenCyc concept for the CycL collection #$Surgery

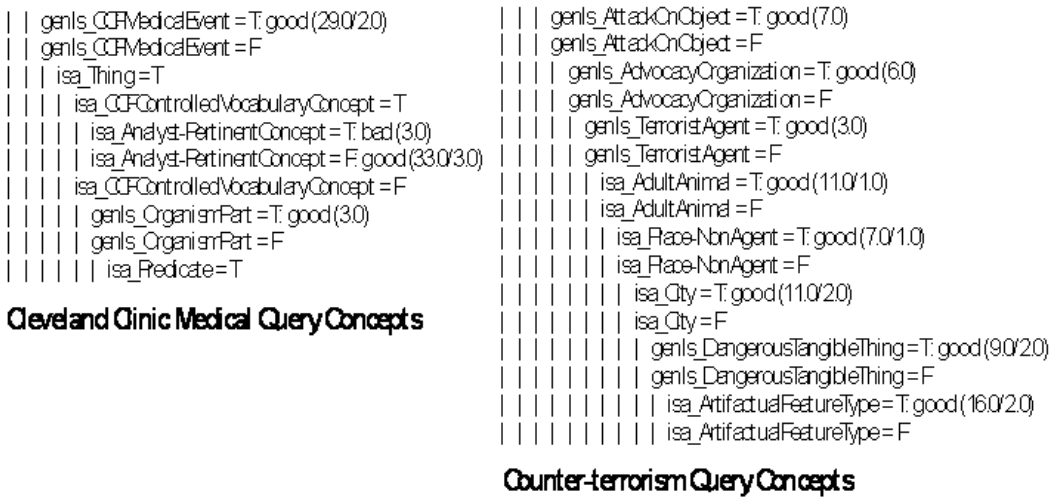

Figure 7: To provide the precision needed for reasoning, English terms can have many possible logical interpretations. Decision trees are used to filter these interpretations of terms in a query to ones appropriate to a domain. By using the Ontology, this filtering is done at a conceptual level that requires few training sentences and few decision points. The fragments shown above are substantial fractions of the trees in use. Such filtering rules would be nearly impossible to learn at the lexical level.

3.2.2. Syntactic analysis and query composition: To understand what the user is saying to it, SRA recognizes terms and then infers partial meaning – expectations and hypotheses about the user's intent (Shah et al. 2006); syntax is used as an adjunct to this process. As an example, the presence of the term 'Hancock Model 342R' (a type of valve prosthesis) in a query, together with the expectation-driving assertion

```
(generateFormulasForElements-TermGenls
        CardiacValveProsthesis
        (TheSet  valveProsthesisTypeImplanted  valveProsthesisTypeExplanted))
```

causes the system to look for possible arguments for these latter predicates (i.e., valveProsthesisTypeImplanted and valveProsthesisTypeExplanted), based on their argument type constraints. Assertions in the Cyc KB constrain the first argument of the ternary predicate valveProsthesisTypeImplanted to be an instance of HeartValveReplacement-SurgicalProcedure, i.e., a particular surgical event; constrain the second argument to be a type of CardiacValveProsthesis; and constrain the final argument to be a particular individual CardiacValveProsthesis.

The second argument is clearly the valve type Hancock Model 342R whose mention triggered the expectation, but once that expectation has been set, any nearby mention of a specific surgery will be strong candidate for argument#1, and a mention of a specific valve (e.g., by its unique manufacturer serial number) will be a strong candidate for argument#3. If suitable arguments are not available, the unfilled positions are left as open variables – typed variables that will most likely get unified, under inference based constraint, when the user selects other fragments. At that time, all those puzzle pieces, with their accompanying constraints, get fitted together into a consistent and plausible whole. Variables that *still* remain will be open variables in the database queries and will therefore define what *columns* need to be present in the answer matrix. E.g., a common one of those is the exact date and time of the surgery; another is the patient's ID#.

SRA's expectation-driving assertions for the medical domain have been generated manually by knowledge engineers, in consultation with domain experts, to maximize usability; this is possible because the domain is somewhat narrow. For broader applications, however, and where less control is needed, such expectations can be generated by forward inference. The "generateFormulas" sentence above, for example, could have been generated entirely automatically using the facts that (1) the specificity of its second argument type is high and (2) this argument type constraint does not apply to many predicates. This sort of meta-reasoning about predicates and the contents of the KB is straightforward, pervasive, and (therefore has been engineered to be) particularly efficient in Cyc.

Generally, the filtering decision trees described above, and the use of specific expectations to combine terms into fragments, are sufficient to offer users a tolerably small set of potential fragments from which to form a query. In some cases, though, syntax is very helpful – in the SRA application, for example, where the ordering of events is particularly important, mixed semantic/syntactic templates are used to recognize and understand temporal constructions. For example, matching the pattern "<Isa:CCFMedicalEvent> *between* <Isa:TemporalThing> *and* <Isa:TemporalThing>" causes its arguments to be interpreted as (temporallyBetween-Inclusive <arg1> <arg2> <arg3>).

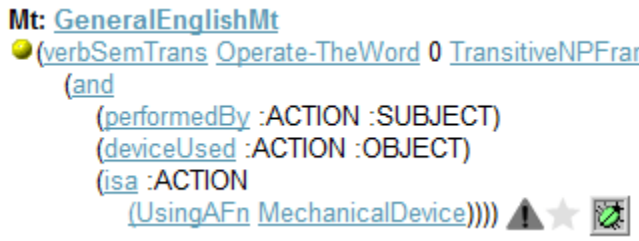

Figure 8: General Cyc parsing encodes the lexical semantics of words using semantic translation rules. The use of heuristic level (HL) modules obviates the need to run these rules dynamically during SRA operation.

It's worth noting that the broader Cyc NL system supports the use of patterns of this kind for almost all predicates and event types. For example, Figure 8 shows the pattern that enables parsing of phrases "<AGENT> [operate] <DEVICE>", for any form of the word "operate", to be interpreted as an event in which a device was used (e.g. "Marvin Minsky operated the PDP-6").

Figure 11, below, shows the final stage in query composition, where Cyc uses inference (usually supported by assertions about predicate argument type constraints and collection disjointness, as in this case, but potentially using any assertion in the KB) to determine which ways of combining a new fragment with an existing query are plausible and which are incoherent. In this surprisingly typical case, it is able to eliminate all possibilities but the correct one in a fraction of a second. Limited meta-reasoning is performed: if two clauses are added with descriptions that differ only with respect to specificity (i.e a description of a surgery, and a valve repair), they are assumed to refer to different entities; even though it is logically possible that the surgery in question *is* the valve repair, it is unlikely that this was the user's intent.

3.2.3. Natural Language Generation is used both for the interaction with users as they express their queries, and in displaying and justifying the answers found during inference. Three kinds of generated text are particularly important: query fragments, variables and table headers, and table cell contents. Query fragment generation is driven from KB content that

describes how to generate syntactically correct renderings of predicates and their arguments. In fact, as we'll describe below (and have described in Baxter et al. 2005), Cyc NLG can render more complex logical sentences, and SRA uses that capability both for temporally complex fragments, to confirm the overall query, and, on demand, to furnish justifications of answers. For brevity, here we'll confine detailed discussion mainly to the generation of fragments.

```
Mt: EnglishParaphraseMt
(genTemplate valveProsthesisTypeImplanted
    (ConcatenatePhrasesFn
        (BestNLPhraseOfStringFn "in the heart valve
                                 replacement")
        (TermParaphraseFn-NP :ARG1)
        (BestNLPhraseOfStringFn ",")
        (TermParaphraseFn-NP :ARG3)
        (HeadWordOfPhraseFn
            (BestVerbFormForSubjectFn Be-TheWord
                (NthPhraseFn 2)))
        (BestNLPhraseOfStringFn "implanted and is a")
        (TermParaphraseFn-NP :ARG2)))
```

Figure 9. A Cyc NLG (natural language generation) assertion.

Consider valveProsthesisTypeImplanted, the ternary predicate which relates a particular valve surgery to the type of prosthesis used, and is offered as a fragment whenever a user mentions something that is known to be a (kind of) valve prosthesis. The Cyc assertion in Fig. 9 expresses how this predicate and its arguments should be generated, including the requirements that the arguments be rendered as noun phrases, and that the first verb in *"in the heart valve replacement :HEART-VALVE-REPLACEMENT, :VALVE-PROSTHESIS is implanted and is a :TYPE-OF-VALVE-PROSTHESIS"* should be an appropriate tense form of "to be" that agrees in number with the paraphrase of the first argument of the predicate.

```
(#$and
  (#$isa ?PROCEDURE1
         :PROCEDURE-TYPE1)
  (#$isa?PROCEDURE2
         :PROCEDURE-TYPE2)
  (#$after-CCF ?PROCEDURE1
               ?PROCEDURE2)
  (#$dateOfEvent-CAE
         ?PROCEDURE2   :DATE))
```

```
(#$and (#$isa ?INFARCTION #$HeartAttack)
   (#$after-CCF   ?INFARCTION
                  ?MEDICALLY-RELATED-EVENT)
   (#$isa ?MEDICALLY-RELATED-EVENT
      (#$SubcollectionOfWithRelationToTypeFn
 #$HeartValveReplacement-SurgicalProcedure
         #$objectActedOn   #$AorticValve))
```

"What aortic valve replacements in 2007 occurred before what myocardial infarctions?"

Figure 10: Multiple components of a logical sentence can be selected for simultaneous paraphrase. Because the logical sentence on the right matches the pattern on the left, a specific generation template can be used to generate the clear English shown in red below in quotes.

The arguments of the predicate are replaced by concrete events, items and types, variables, or sequences of underscores, as appropriate. For speed, when SRA first displays this fragment, it does so without agreement; full generation is done in the background, and each of the phrases is replaced with the morphologically correct variant as it is ready.

Because it is important to render phrases involving time clearly, specific patterns for rendering portions of a logical sentence are used in these cases. These patterns, which are produced by

forward inference, involve a template, as shown on the left of Figure 10, and a generation template similar to the one shown in Figure 9, and produce a concise paraphrase of all matching parts of a logical sentence. The query sentence in the figure is paraphrased as “"What aortic valve replacements in 2007 occurred before what myocardial infarctions?".

Since SRA users are formulating queries, the system needs to have a way to refer to the items they are trying to find. It does this using variables and corresponding table headers. Both are generated using constraints derived from the context in which they appear. In some cases, Cyc has explicit knowledge of how to refer to the role of a predicate argument; for example the assertion *(denotesArgInReln Diagnose-TheWord CountNoun hasDiagnosis 2)* means that the second argument of the predicate “hasDiagnosis” can be referred to as “diagnosis”, the count noun form of the word “diagnose”. There are 1750 such assertions in the KB, but if this information is not available, more general constraints are used: the argument type constraints for the predicates in which the variable is used are gathered (for example, valveProsthesisTypeImplanted, which we saw above, is constrained to have a valve replacement procedure as its first argument, a type of heart valve prosthesis as its second, and a particular valve as its third), along with explicit type constraints on the variable (via “isa” [instantiation], or “genls” [subclass], clauses in the query). The most specific of these constraints are tried first, and the first one that can be rendered as a non-plural noun, has not been used elsewhere, and is not more than 30 characters long, is used. In the user interface screenshots, one can see several variables and column headers that have been generated this way, including *“PATIENT”*, *“BLOODSTREAM-INFECTION” and “ELAPSED-TIME”*. Recently, in response to user feedback, the system was altered to maximize variable name consistency; it no longer replaces a variable name with a new one *merely* because its constraints have tightened during query refinement.

The current SRA attempts to compromise between the reach of the NLP techniques applied, and the need for responsiveness. As machines become more powerful, it becomes possible to attempt more sophisticated analysis. In the short term, in work with Elizabeth Coppock, we are exploring applying semantic combination rules, in which the cooccurrence of specific patterns of logical interpretation in parts of an input query triggers the production of a correct (possibly different) representation of an overall situation, and the rejection of alternatives. In the longer term, we are exploring techniques for automatically learning logical interpretations of constructions, by reading (Curtis et al. 2009)

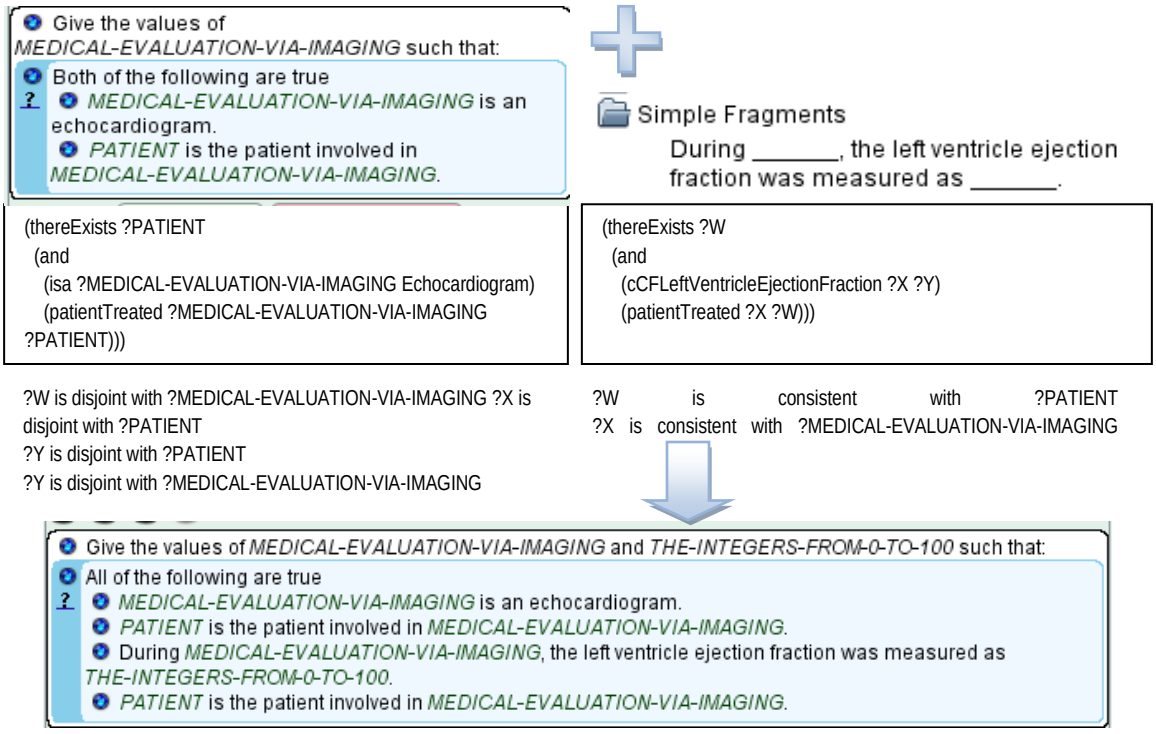

Figure 11: Inference based on (1) explicit type information (isa and genls) and (2) predicate arg. constraints determines how to combine new fragments to form a more complete query.

3.2.4. Failing soft: Semantic Search based on Cyc

The above process does not always succeed, e.g., when the data required to answer the query is still "locked up" in more or less unstructured form such as natural language texts. This brings us to Challenge#5, *semantic searching* (versus just keyword searching) in cases where the correct answer cannot be calculated due to failure to understand the query, or due to missing structured data. Our approach to this is similar to #1-4 at the internal SRA representation and algorithms level, but visually appears quite different to the user. in Figure 12, semantic search is enabled for the paragraphs and pages of the annual "Outcomes" booklet issued by the cardiothoracic surgery division of the Cleveland Clinic. The user, a prospective patient, types in

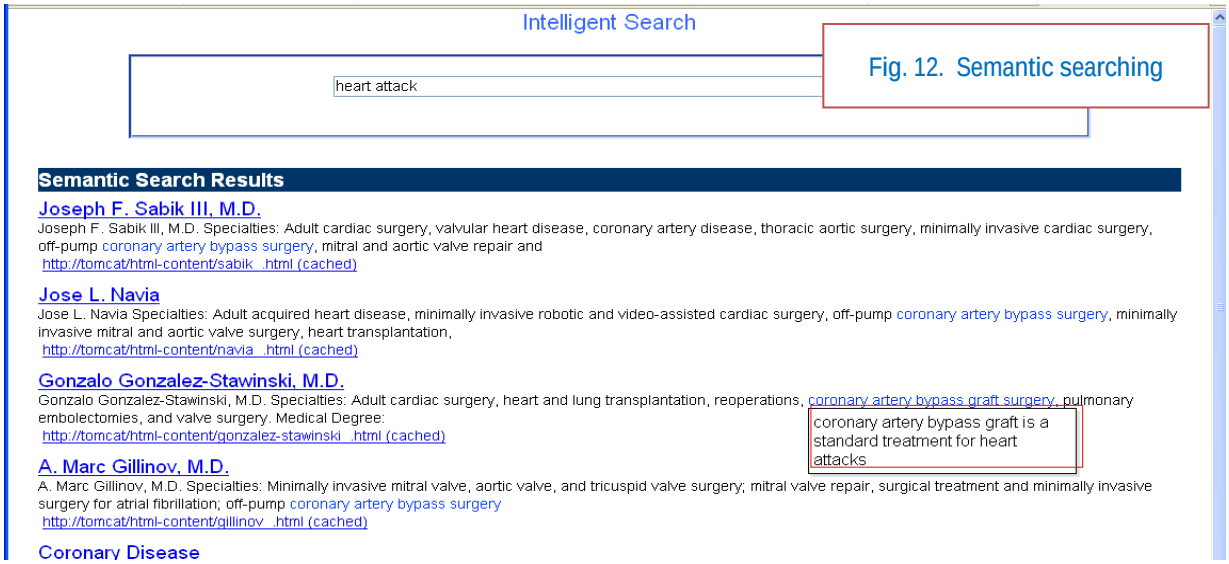

Fig. 12. Semantic searching

"heart attack". But the Outcomes booklet does not contain that colloquial term anywhere. Even worse, the only places where those two terms *do* co-occur in proximity are on pages that are both irrelevant and frightening to the prospective patient (e.g., about heart-lung transplants.) Nevertheless, relevant "hits" are returned because the Cyc ontology knew that "heart attack" was a denotation for myocardial infarction, and the Cyc KB knew that CABG is a common treatment after MI's, and because semantic tagging had identified which paragraphs and pages were *about* CABGs. Similarly, semantic representations of MI's, flesh eating bacteria, heart-lung transplants, etc., allowed it to *not* retrieve those irrelevant pages even though a string-based search engine would not have understood and would have included those false positives.

If the user clicks on Gonzalez-Stawinski here, the system utilizes its partial understanding of the query, and of the retrieved pages, and displays not only the usual "page" about that surgeon, but also an extra graph that does not normally appear "out of context" on that page but is very useful to a prospective patient. This graph, derived from the CCF data bases, shows the number of CABG procedures that surgeon has performed each year for the past decade.

4. Conclusion and Next Steps

Scaling up: SRA and, more broadly, the Cyc Analytic Environment CAE, are intended to serve as a bridge towards a future where our systems deeply understand the intent behind user queries, where our systems actively seek out and background knowledge and data that must be used to satisfy them. We have experimented with the CAE, on which SRA is based, in the terrorism and financial domains, and believe that it is generally useful. To realize the broadest benefit, though, it needs to be the case that nearly every query term will be understood by the system; part of this requirement is being met by initiatives such as linked open data, which is driving a great increase in the availability of data-grist for inference. SKSI allows Cyc to make use of such data, and data in more conventional databases, during inference.

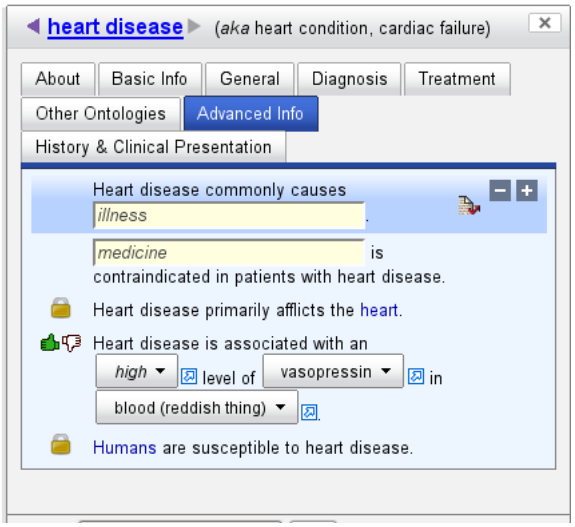

Fig. 13: This interface, CURE (Content Understanding, Review or Entry), allows non-logicians to enter knowledge by answering questions. As initial questions are answered in the form, inference based on their answers prompts additional questions.

But to support natural queries, the terms must be described in enough detail to allow their lexicalizations to be recognized, and their likely relations to other terms to be identified. Although the manual effort of building Cyc has been worthwhile, as a sort of "priming of the pump", we now have interfaces that allow us to bootstrap from that knowledge in acquiring more. The CURE (Content Understanding, Recognition, or Entry) interface, shown in Figure 13, allows concepts to be created, and fleshed out

with relevant assertions, by untrained users. CURETTE is a lightweight version of CURE that can easily be embedded in web pages. In the longer term, the prospects for increasingly automated knowledge acquisition seem bright. We have been working on automated rule learning over large conceptual and relational vocabularies (Cabral et al. 2005, Curtis et al. 2009), and are participating in the DARPA Machine Reading program, in support of this goal.

The other key to broad applicability is simply having the inferential scale needed to support queries depending on very large rule sets applied to web scale data. We have steadily increased the speed with which the Cyc inference engine operates, and the size of the KBs that it can handle, and are pursuing paths to even greater scalability through our participation in the EU LarKC research program (Fensel et al. 2008), which is attempting to build a platform (based on part of the Cyc source code) for web-scale inference.

Within SRA, a clinical researcher should be able to explore novel hypotheses requiring logically or statistically combining information from multiple medical specialties; using SRA, a clinician should be able to state a cluster of potentially interrelated attributes and values for their patient, and ask about similar patients' treatments and outcomes. The natural way to investigate this will be by expanding the underlying ontology and KB to more and more domains (e.g., the next targets at CCF include electrophysiology, interventional and diagnostic cardiac catheterization, heart failure and transplantation, and infectious disease.) We wish to explore, as those domains are added, whether some of the components of SRA (e.g., the parser) "scale up" better or worse than others, and whether the SRA becomes qualitatively more useful by handling queries cutting across many departments and data bases.

Even using tools like the CURE, domain scaling requires considerable but tolerable effort; consider cardiac catheterization ("cath"). Even though at CCF there are separate departments and separate data bases for diagnostic cath and interventional cath, there is sufficient overlap in concepts and terminology that they may be treated as one domain for SRA purposes. The approximately 500 *new concepts* and 6500 new assertions which are currently being added, for this domain, include knowledge about types of catheters and attachments, associated devices such as those for stemming post-removal blood loss, common procedures and their sub-steps (down to the level of ordering and other constraints among the sub-steps of a procedure), diagnostic rules, relevant anatomy, diseases, medications, indications and contra-indications, and heuristics (rules of good judgment) about degrees of risk and likelihood of outcomes. About half of the 6500 new assertions for this domain are lexical assertions, expressing the various ways each of the 500 new concepts is denoted in "medical English" and tying it to standards including Snomed and ICD-9 and -10, along with more traditional linguistic assertions indicating for example whether each noun is a count noun or mass noun. The other half of the 6500 represent pieces medical knowledge about cath, assertions involving one or more of the 500 new terms and, in almost all cases, also involving one or more of the preexisting 500,000 concepts in the Cyc ontology, partially defining those new concepts and integrating them into the existing ontology.

The initial acquisition of concepts, terminological assertions, and medical knowledge assertions for each domain is done top-down. E.g., for cardiac catheterization, the first step was to use (Kern 2004) as a reference. The next "pass" after that, which is currently underway, is to expand the ontology and the KB as needed by looking at a representative sample of clinical research and clinical queries involving terms from that domain. Many of the former can be harvested automatically from websites such as clinicaltrials.gov, and some of both types can be retrieved from logs of recent manually-translated-into-database-form queries.

Smarter Data Entry: Patients who are admitted to multiple departments at a medical center often are asked the same or related questions (e.g., about family history) repetitively. By installing the SRA "behind" the data acquisition screens, some of this can be avoided. Some such data can be inferred unambiguously from already-entered data about that patient; in other cases, the range of possible answers can at least be constrained (resulting in, e.g., a small(er) menu of choices). When contradictory information inevitably is added, about a patient, there is at least the possibility of recognizing it in real time – deducing that there is a logical conflict -- and flagging it. And when there are multiple "blanks" yet to be filled in, instead of providing no guidance (or, even worse, locking the data enterer into a fixed sequence of queries to respond to), the system could infer and highlight the queries that would be "best" to answer next. In this case "best" includes an information-theoretic component (answering this query next is likely to constrain many other as-yet-unasked queries), an outcomes component (answering this query next might turn out to be vital to providing this patient's urgent care), a cognitive load component (don't "jump around" changing contexts more than necessary), and other heuristics no doubt apply.

Clinical Use: Although the SRA has been developed in the context of cohort-selection for clinical outcome studies, the current push towards standardized electronic patient records suggests an even more powerful future use: directly data-driven clinical practice, in which treatment outcome predictions for a particular patient are dynamically produced by analysis of the outcomes of the most similar other patients. The SRA would be used to query about individual cases; e.g.: "This patient has had elevated creatinine levels since their mitral valve repair and has a history of renal failure. What have been the recommended treatments over the past five years for patients with these conditions?" The same kinds of data base queries would be generated, but instead of a cohort of patients being returned, sets of treatment options and outcomes would be retrieved and statistically analyzed.

Relating qualitative and quantitative terms: Often, part of the "full understanding" of the user's query means interpreting qualitative terms like "small", "minor", "enlarged", "significant", "unusual", etc. While relative terms such as these *can* be expressed in Cyc, often the physician "really" has some more precise meaning in mind. E.g., Figure 14 shows an assertion recently added to SRA (i.e., to the Cyc KB), expressing in predicate calculus a criterion for *left atrial enlargement* in women: in working with the physicians to articulate this and express it sufficiently rigorously in CycL, it turned out that what they meant – in their domain – was: having an atrial diameter exceeding 3.8 cm.

```
(implies
(and
    (cCFhasLeftAtriumDiameter ?EVT ?D)
    (greaterThan ?D ((Centi Meter) 3.8))
    (patientTreated ?EVT ?PAT)
    (patientSex ?PAT FemaleHuman)
    (rdf-type ?EVT ?TYPE)
    (genls ?TYPE CCF-Evaluation))
  (isa ?EVT EvaluationThatIndicates-
                LeftAtrialEnlargement))
```

Figure 14. A typical domain assertion added to SRA.

More deeply infer what the user plausibly intended by their query: The goal is to steadily reduce and *eliminate* the need for human intermediaries "in the loop", and to reduce and *eliminate* the need to ask the physician any follow-up clarifying questions. This is an iterative process, incrementally approaching competence by training the system on a large corpus of examples. The existing CCF library of over 1000 intermediary-processed queries forms a natural starting point for this corpus. Augmenting this are tens of thousands of others

from various domains on **www.clinicaltrials.gov**. To expand the corpus, clinical researchers should produce alternate versions of each query, providing a number of different plausible syntactic forms and wordings for the same semantic query.

At present, the SRA system uses three sources of information to establish meaning: syntax, statistics, and background knowledge. All three could be utilized even more than they currently are. *Syntactically,* we can expand detailed parsing from its current application to identifying relations and argument, and deep understanding of time expressions to cover correct assignment of the roles in a syntactic frame, and to analyzing the internal structure of novel noun phrases. This should significantly reduce the number of candidate frames. In *statistics,* we hope to extend the trained filtering that currently identifies plausible senses of terms given the topic to jointly maximize the probability of an interpretation over multiple ambiguous query terms. We will train a probabilistic model of modifier attachment, to allow more "query fragments" to be automatically assembled. Finally, regarding *background knowledge*, we plan to write new disambiguation and "fragment" addition rules, and tighten the logical constraints on arguments of logical relations, to enable more effective use of the knowledge added for interpretation.

Part of the source of power being tapped by SRA is the fortuitous fact that natural language understanding for detailed queries, even quite long queries, can – at least in the medical domains explored to date – be performed in a largely *compositional* fashion, recursively constructing and refining pieces of the overall query, rather than having to reason very much about the query as a whole. Only once the query is mostly understood, and few ambiguities remain, is it practical to reason about "far apart" pieces of the query to see whether medical knowledge, discourse pragmatics, or data in the target DBs can point to a resolution.

Synthesizing a terser yet more comprehensible answer for the user. Condensing, formatting, and exporting the answers to a user's query sounds like a "frill", compared to the task of actually getting the *correct* answers to their question. So we were surprised to find that empirically this has been one of the biggest factors affecting whether and to what extent physicians directly use the SRA.

The first and easiest "side" of this task to focus on will be getting SRA to intelligently *pare down* the answers, and especially the justifications for the answers, removing as much prior and tacit knowledge as possible. SRA will do this by drawing on much the same knowledge used in *understanding* the queries and in formulating a plan to retrieve elements of data from which to answer the query. Producing a clear answer or justification has syntactic features (combining *n* attributes of a procedure into a single descriptive noun phrase), trainable probabilistic features, and background knowledge. But besides general knowledge and medical knowledge, success at this task will depend on building up and using a powerful explicit model of the user – e.g., what do they know and not know; what sorts of details do they like and not like to see included; what queries have they recently asked of the system; what is their purpose in asking this query? Consider e.g. the last of those variables, their purpose: even at a very broad level, if they have a clinical research purpose in asking the query, the sort of answers, time frame for the answers, etc., is quite different than if they are a clinician asking about a particular patient. This notion of the user's context is represented explicitly in Cyc, and thus can be easily represented in SRA. Experimental approaches for using explicit user and task models that were developed for intelligence analysis (in the Cyc Analytic Environment, CAE, on which the SRA was initially based) will be applied and extended to the medical domain. The important user and task attributes, and the rules associated with each one, will be captured in post-usage debriefing sessions. User modeling research indicates that even relatively small user models and context

models are sufficient for establishing enough details of to sustain a high degree of user comfort with question-answering programs. In particular, we expect this to lead to very few new concepts being added to the ontology, but to a large number of rules being added relating user variables (and variables about the context in which the user is currently interacting with the system) to display modality, location, priority, format, and editing choices.

Extending the current Semantic Searching capability: There are two methods by which Cyc-based semantic searching is performed. The "strong" version is to partially parse a large corpus of text documents, much as SRA partially parses users' queries. This leads to an identification of what that document (and that paragraph in that document) is *about*, the ontological terms – individual objects, collection, predicates and relations – and some of the fragment-like clauses (predicates applied to arguments, sometimes with some of the arguments being left as quantified variables). By partially parsing the user's query, Cyc can then perform inference to find connections (and their semantic strength) between the query and each document in the tagged corpus, or even each *paragraph*.

The second, "weak" version of semantic searching involves taking the English paraphrase of the query, to the extent available, or the initially typed query, to the extent the paraphrasing failed, and then augmenting the query with "OR" clauses – disjoining Boolean terms – based on their being alternative ways of denoting the same terms or very close "relatives" in the ontology, and augmenting the query with conjoined "AND NOT" clauses where there are different, unintended denotations for some of those very same words and phrases, in each case finding some very close "relatives" of those unintended concepts ("betrayers") so that any false negative page found for the term is likely to contain one or more of those betrayers. In a query like "Rhinoplasties performed in TX or MI during 1991", "MI" refers to Michigan, so synonyms of "myocardial infarction" would be the AND-NOT terms augmenting the query before handing it to Google or PubMed.

Unlike the other SRA extensions we have just described, this one may succeed or fail based more on the algorithms developed for it. For example, one possible algorithm would be to generate alternate paraphrases of the query, find "hits" for each paraphrase, and upgrade "hits" that turned up for multiple paraphrases.

One of the factors we do not yet have much in the way of preliminary results about, is the extent and way in which the clinical researcher and the clinician will make use of this capability, and that will be one of the things we hope to discover empirically. We already described how one use of Semantic Searching is as a fallback: the user will still likely want to see pointers to e.g. relevant recent literature even in cases where SRA *can* answer their query. Seeing such articles may be of value to them in more rapidly converging on the queries they most want to ask, queries which in some cases *will* be answerable by SRA.

Final Conclusion: We have made progress in getting SRA to answer physicians' ad hoc queries about patient data orders of magnitude faster than what had been "best practices", but there is much room for, and many different directions for, future improvement and wider application. As was the case with search engines, once the process of formal ad hoc query articulation via clarification dialogue is sufficiently fast and easy to use, and incorporates appropriate privacy controls, the general public may become the heaviest users, leading to a qualitative change in the way that patterns are first detected in patient data, and to a qualitative improvement in patient informedness, involvement, satisfaction, and outcomes.

ACKNOWLEDGEMENTS. The authors would like to express their appreciation to the many organizations which have provided support for this work, including CCF, Cycorp, DARPA, IARPA, NIH, NIST, NSF, Rome Labs (AF), and to the staff members of those various organizations who have contributed directly or indirectly to the technology.

REFERENCES

Baxter, D.; Shepard, B.; Siegel, N.; Gottesman, B.; and Schneider, D. 2005. Interactive Natural Language Explanations of Cyc Inferences In *AAAI 2005: International Symposium on Explanation-aware Computing*, Washington, DC.

Cabral, J.; Kahlert, R.C.; Matuszek, C.; Witbrock, M.; and Summers, B., 2005. Converting Semantic Meta-Knowledge into Inductive Bias. In *Proceedings of the 15th International Conference on Inductive Logic Programming*, Bonn, Germany.

Cardarelli, M. G,; Gammie, J. S.; Brown, J. M.; Poston, R. S.; Pierson, R. N.; and Griffith, B P. 2005. A Novel Approach to Tricuspid Valve Replacement: the Upside Down Stentless Aortic Bioprosthesis", *The Annals of Thoracic Surgery* 80:507-510.

Coppock, E.; and Baxter, D. 2009. Translation from Logic to English with Dynamic Semantics in *Proceedings of Logic and Engineering in Natural Language Semantics VI*, Tokyo, Japan.

Curtis, J.; Baxter, D.; Wagner, P.; Cabral, J.; Schneider, D.; and Witbrock, M., 2009. Methods of Rule Acquisition in the TextLearner System, in S. Nirenburg & T. Oates (Eds): *Proceedings of the 2009 AAAI Spring Symposium on Learning by Reading and Learning to Read*, Palo Alto, AAAI Press. pp. 22-28,

Deaton, C.; Shepard, B.; Klein, C.; Matans, C.; Summers, B.; Brusseau, A. P.; Witbrock, M. J.; and Lenat, D. B. 2005. The Comprehensive Terrorism Knowledge Base in Cyc. In *Proceedings of the 2005 International Conference on Intelligence Analysis*. McLean, Virginia.

Fensel, D.; van Harmelen, F.; Andersson, B.; Brennan, P.; Cunningham, H.; della Valle, E.; Fischer, F.; Huang, Z.; Kiryakov, A.; Lee, T.K.; Schooler, L.; Tresp, V.; Wesner, S.; Witbrock, M. and Zhong, N. 2008. Towards LarKC: a Platform for Web-scale Reasoning in *Proceedings of the Second IEEE International Conference on Semantic Computing*, IEEE-ICSC2008, Santa Clara, CA, USA

Fortuna, C.; Ivan, B.; Padrah, Z.; Bradesko, L.; Fortuna, B.; and Mohorčič, M. 2009, Demonstration: Wireless Access Network Selection Enabled by Semantic Technologies. In 2009 International Semantic Web Conference (ISWC), , Washington DC.

Gillinov, A. M.; Blackstone, E. H.; Alaulaqi, A.; Sabik, J. F. III; Mihaljevic, T.; Svensson, L. G.; Houghtaling, P. L.; Salemi, A.; Johnston, D. R.; and Lytle, B. W. 2008. Outcomes after Repair of the Anterior Mitral Leaflet for Degenerative Disease. *Ann Thorac Surg* 86:708-717.

Hickey, E. J.; McCrindle, B. W.; Caldarone, C. A.; Williams, W. G.; and Blackstone, E. H. 2008 Making Sense of Congenital Cardiac Disease with a Research Database: The Congenital Heart

Surgeons' Society Data Center. *Cardiology in the Young* 18 (Suppl 2):152-162.

Hoercher, K. J.; Nowicki, E. R.; Blackstone, E. H.; Singh, G.; Alster, J. M.; Gonzalez-Stawinski, G. V.; Starling, R. C.; Young, J. B.; and Smedira NG. 2008. Prognosis of Patients Removed from a Transplant Waiting List for Medical Improvement: Implications for Organ Allocation and Transplantation for Status 2 Patients. *J Thorac Cardiovasc Surg* 135:1159-1166..

Kaplan, R. M.; Riezler, S.; King, T. H.; Maxwell, J. T.; and Vasserman, A. 2004. Speed and accuracy in shallow and deep stochastic parsing. *In Proceedings of the Human Language Technology Conference/North American Chapter of the Association for Computational Linguistics Meeting*. Boston; MA.

Kaple, R. K.; Murphy, R. T.; DiPaola, L. M.; Houghtaling, P. L.; Lever, H. M.; Lytle, B. W.; Blackstone, E. H.; and Smedira, N. G. 2008. Mitral Valve Abnormalities in Hypertrophic Cardiomyopathy: Echocardiographic Features and Surgical Outcomes. *The Annals of Thoracic Surgery* 85(5):1527-1535.

Kern M. (ed.). 2004. *The Cardiac Catheterization Handbook, 4th Edition*, Philadelphia: Mosby/Elsevier.

Klein, D.; and Manning, C.D. 2003. Accurate Unlexicalized Parsing. In *Proceedings of the 41st Meeting of the Association for Computational Linguistics*, pp. 423-430.

Koch, C. G,; Li, L.; Shishehbor, M.; Nissen, S.; Sabik, J.;, Starr, N. J.; and Blackstone, E. H.; 2008. Socioeconomic Status and Comorbidity as Predictors of Preoperative Quality of Life in Cardiac Surgery. *J Thorac Cardiovasc Surg* **136**:665-672..

Lang, W. (Dir.); Tracy, S. (Perf.); and Hepburn, K. (Perf.) 1957. *Desk Set*, 20th Century Fox,

Lehmann, J.; Schüppel, J.; Auer, S. 2007. Discovering Unknown Connections – the DBpedia Relationship Finder. In Proceedings of 1st Conference on Social Semantic Web, CSSW2007, Leipzig. Volume P-113 of GI-Edition – Lecture Notes in Informatics (LNI). Bonner Köllen Verlag..

Lenat, D., Borning, A., McDonald, D., Taylor, C., and Weyer, S. 1983. Knoesphere: Building Expert Systems With Encyclopedic Knowledge. *Proc. IJCAI 1983*: 167-169.

Lenat, D.B. 1995 Cyc: A Large-Scale Investment in Knowledge Infrastructure. *Communications of the ACM* 38(11).

Lenat D. B.; and Guha R. V.; 1989. *Building Large Knowledge Based Systems: Representation and Inference in the Cyc Project*, Addison-Wesley..

Masters, J., and Güngördü, Z. 2003 Structured Knowledge Source Integration: A Progress Report In *Integration of Knowledge Intensive Multiagent Systems*, Cambridge, Massachusetts, USA.

Matuszek, C.; Cabral, J.; Witbrock, M.J.; and DeOliveira, J. 2006. An Introduction to the Syntax and Content of Cyc. In *Proceedings of the 2006 AAAI Spring Symposium on Formalizing and Compiling Background Knowledge and Its Applications to Knowledge Representation and Question Answering*, Palo Alto, CA.

Mihaljevic T., Nowicki, E. R.; Rajeswaran, J.; Blackstone, E. H.; Lagazzi, L.; Thomas, J.; Lytle, B. W.; and Cosgrove, D. M. 2008. Survival after Valve Replacement for Aortic Stenosis:

Implications for Decision Making. *J Thorac Cardiovasc Surg* 135(6):1270-1279..

Sabik, J. F. III; Stockins, A.; Nowicki, E. R.; Blackstone, E. H.; Houghtaling, P. L.; Lytle, B. W.; and Loop, F. D. 2008. Does Location of the Second Internal Thoracic Artery Graft Influence Outcome of Coronary Artery Bypass Grafting? *Circulation* 118 (14 Suppl):S210-215..

Schneider, D.; Matuszek, C.; Shah, P.; Kahlert, R.; Baxter, D.; Cabral, J., Witbrock, M.; and Lenat, D.B. 2005. Gathering and Managing Facts for Intelligence Analysis, In *Proceedings of the 2005 International Conference on Intelligence Analysis*, McLean, Virginia.

Shah, P.; Schneider, D.; Matuszek, C.; Kahlert, R.C.; Aldag, B.; Baxter, D.; Cabral, J.; Witbrock, M.; and Curtis, J. 2006. Automated Population of Cyc: Extracting Information about Named-entities from the Web, In *Proceedings of the Nineteenth International FLAIRS Conference*, Melbourne Beach, FL, pp. 153-158.

Siegel, N.; Shepard, B.; Cabral, J.; and Witbrock, M. J. 2005. Hypothesis Generation and Evidence Assembly for Intelligence Analysis: Cycorp's Nooscape Application. In *Proceedings of the 2005 International Conference on Intelligence Analysis*. McLean, Virginia.

AUTHOR BIOGRAPHICAL STATEMENTS:

Douglas Lenat received his PhD in Computer Science from Stanford, investigating automated discovery based on "interestingness" heuristics, for which he received the 1977 IJCAI Computers and Thought Award. He was one of the co-founders of Teknowledge, and of AAAI, and in the inaugural set of AAAI Fellows. Besides professoring at CMU and Stanford, he was Principal Scientist at MCC, where he founded the Cyc Project in 1984 – something he called "ontological engineering" to distinguish it from "knowledge engineering". At the end of 1994, he founded Cycorp, where he continues to serve as CEO. Dr. Lenat is a Fellow of the AAAS, has authored almost a hundred refereed papers and several books and book chapters, ranging from machine learning to knowledge based systems, representation, and inference, and is an editor of the J. Automated Reasoning, J. Learning Sciences, and J. Applied Ontology. He is an Advisory Board member of TTI Vanguard, and has consulted for numerous companies, agencies, and the White House.

Michael Witbrock holds a PhD in Computer Science from Carnegie Mellon University, and is VP Research at Cycorp and CEO of Cycorp Europe. He is particularly interested in automating the process of knowledge acquisition and elaboration, extending the range of knowledge representation and reasoning to mixed logical and probabilistic representations, and in validating and elaborating knowledge in the context of task performance, particularly in tasks that involve understanding text and communicating with users. He is author of numerous publications in areas ranging across knowledge representation and acquisition, neural networks, parallel computer architecture, multimedia information retrieval, web browser design, genetic design, computational linguistics and speech recognition.

David Baxter received his Ph.D. in Linguistics from the University of Illinois at Urbana-Champaign and has been a member of Cycorp's Natural Language staff since 1998. He has developed and maintained compositional parsers and Cycorp's natural-language generation functionality, the declaratively represented Cyc-English lexicon, and is a lead developer of the Semantic Research Assistant.

Eugene Blackstone received his MD degree in 1966 from U. Chicago. In 1972 he joined the faculty of the U. Alabama at Birmingham (UAB) where he directed the cardiothoracic surgery research program. In 1993, he and Dr. John Kirklin proposed a proof-of-concept Computerized Patient Record consisting entirely of values for variables that would facilitate patient care and provide discrete data for generating new knowledge, linking each value with both context information (ontology) and medical process information. The resulting directed acyclic graph was a forerunner of semantic technology

based on RDF that became SemanticDB at Cleveland Clinic. Dr. Blackstone left UAB for Cleveland Clinic in 1997, and directs Clinical Investigations for the clinic's Heart and Vascular Institute. He represents Cleveland Clinic in W3C

Chris Deaton is a senior ontologist at Cycorp. Chris received a BA in Philosophy from Western Washington University and an MA in Philosophy from the University of Massachusetts in Amherst, where he specialized in philosophy of language and contemporary metaphysics. With Cycorp since 2002, he has designed large additions to Cycorp's terrorism and medical ontology. He is the project manager and lead ontologist for Cycorp's current collaboration with the Cleveland Clinic semantic database group.

Dave Schneider received his PhD in Linguistics from U. Delaware, and currently is Cycorp's Natural Language development lead. His graduate work focused on computational and psychological aspects of incremental natural language understanding. Much of his work over the last ten years at Cycorp has focused on knowledge acquisition, where he has worked at the intersection of Cyc's NLP, ontology, and inference systems. His publications include work on automated and semi-automated knowledge acquisition, natural language generation and understanding, and psycholinguistics.

Jerry Scott has founded and/or served as CEO of several healthcare informatics companies including Healthcare Communications (Cloverleaf), Discover Systems, Cyberplus (the predecessor to Health Language), MedBiquitous, and now Research Intelligence. Under his management, and in conjunction with SNOMED, many of the foundation research and development projects were carried out which are enabling the healthcare industry to move toward adopting universal medical terminology, and to integrate disparate terminologies prior to that eventuality.

Blake Shepard is an Ontologist at Cycorp. He received his Ph.D. in Philosophy from The University of Texas at Austin and has been with Cycorp since 1999, where he has authored and co-authored several papers and technical reports. He manages portions of, and is a senior ontologist for, Cycorp's current collaboration with the Cleveland Clinic semantic database group. His long-standing interest is in the development of ontologies with maximal fidelity and inferential tractability.