Talking to the Semantic Web –

A Controlled English Query Interface for Ontologies

Abraham Bernstein, Esther Kaufmann, Norbert Fuchs, June von Bonin

Department of Informatics, University of Zurich, Switzerland

Abstract

The semantic web has presented the vision of the distributed, dynamically growing knowledge base, which is based on principles of formal logic. Common users, however, have been shown to have problems even with the simplest Boolean expressions. As queries from web search engines show, for example, the great majority of users simply do not use any Boolean expressions. So how can we help users to query a web of logic, which they don't seem to understand?

This paper tries to address the problem by presenting a natural language front-end to semantic web querying. The front-end allows formulating queries in controlled English, a subset of natural English. Using Attempto Controlled English the query is translated into a discourse representation structure, a variant of first-order logic, which is then translated to a semantic web querying language. As the examples in the paper show, our approach offers great potential for bridging the gap between the idea of the semantic web and its real-world users. As such it should allow users to query the the knowledge on the semantic web without having to learn a special formal language.

1. Introduction

The semantic web has presented the vision of the dynamically growing knowledge base, which akin to the World Wide Web should allow users to draw on and combine the distributed information sources specified in formal logic. Common users, however, have been shown to have problems even with the simplest Boolean expressions. Experience in information retrieval, for example, shows that users are better at understanding graphical query interfaces than simple Boolean queries [Spoerri 1993]. As queries from web search engines show, for example, the great majority of users do simply not use any Boolean expressions. *So how can we bridge the gap between the logic-based semantic web and real-world users, who are at least ill at ease and, oftentimes, unable to use formal logic concepts?*

This paper proposes to address the problem by *presenting a natural language front-end to the semantic web*. In its current form the front-end provides users with a controlled natural language interface to initiate queries. The controlled natural language used, Attempto Controlled English (ACE) [Fuchs et al. 2004; Fuchs et al. 2000], provides a unambiguously parsable and interpretable subset of English, which is then translated to a semantic web query language providing users with an almost natural language interface to the semantic web. As experience with controlled languages has shown, they are much easier to learn by end-users than formal logic. We, therefore, believe that the approach presented here has great potential in bridging the gap between the semantic web and its end-users and becoming a major enabler for the growth of the semantic web.

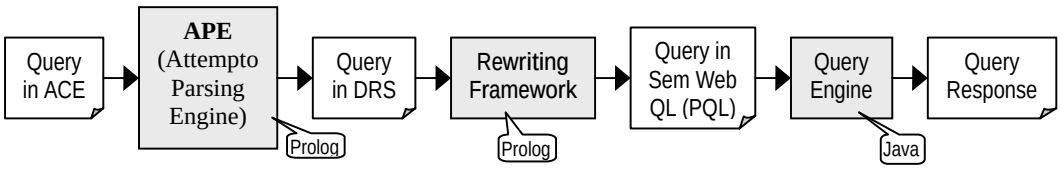

Figure 1: Overall data flow of the controlled English query front-end

In proceedings of 14th Workshop on Information Technology and Systems (WITS 2004),
Washington D.C., December 2004.

The rest of this paper closely follows the data flow of the query front-end (Figure 1). Section 2 introduces ACE and the Attempto Parser Engine (APE), on which we draw for our solution. APE translates the controlled English sentences to a discourse representation structure (DRS), a variant of first-order logic introduced by [Kamp 1981; Kamp et al. 1993] capturing among other things the linguistic phenomena of anaphoric reference (e.g. pronouns) in sentences. Section 3 introduces the rewriting framework, which translates the DRS to a semantic web query language for evaluation using a standard query engine (which we don't discuss in this paper). We then provide a first assessment of the approach posing some real-world queries to the knowledge base and close with a discussion of the limitations and future directions of the proposed solution.

2. Controlled English as a Query Language

Our query front-end automatically processes queries given in Attempto Controlled English (ACE), a controlled natural language originally designed for requirements specifications and knowledge representation [Fuchs et al. 2004; Fuchs et al. 2000]. ACE is a subset of English meaning that each ACE sentence is correct English, but not vice-versa. Its grammar is specified by a small set of construction and interpretation rules. The construction rules allow users to build simple sentences (e.g. "John sells books.") and composite sentences (e.g. "If John sells books and John's business does not fail then he is content.") The deterministic interpretation rules eliminate syntactic and semantic ambiguities, for which natural languages are highly notorious, hereby reducing the computational complexity of processing ACE sentences. As such, ACE avoids the major disadvantages of a full natural language processing (NLP) approach while maintaining the ease of use for end-users and allowing the translation of all ACE sentences to first-order logic.

In some cases ACE requires some non-intuitive sentence constructs, but those can easily be avoided by following some simple rules. As an example, consider the (infamous) sentence "The man sees the girl with the telescope." In full English the prepositional phrase "with the telescope" either modifies the verb phrase, leading to the interpretation that the man has the telescope, or the noun phrase "the girl," meaning that the girl has the telescope. The ACE interpretation rules limit the interpretation to the former, avoiding this ambiguity while not overly limiting the expressability.

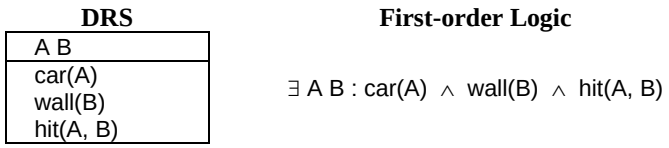

Figure 2: DRS and first-order logic representation of "The car hits the wall"

The Attempto Parsing Engine (APE) – implemented in Prolog as a Definite Clause Grammar – translates an ACE text into a discourse representation structure (DRS), representing the textual information of a sentence (discourse) [Kamp 1981; Kamp et al. 1993]. A DRS consists of discourse referents, i.e. quantified variables representing the objects of a discourse, and of conditions for the discourse referents. The conditions can be logical atoms or complex conditions built from other DRSs and logical connectors (negation, disjunction, and implication). For example, the sentence "The car hits the wall" is shown in its typical box-styled DRS representation in Figure 2 on the left. The two discourse referents, A and B, are shown at the top and the three conditions are listed below divided by a line.

3. From DRS to Semantic Web Query Language

As the next step, and one of the core contributions of this research, the system translates the DRSs produced by APE into a semantic web query language, which is then used to query an ontology. As an exemplary ontology we chose the *MIT Process Handbook* [Malone et al. 1999], which describes organizational processes. We chose the Process Handbook because it treats a real-world usage domain that everybody can relate to (as opposed to ontologies from very specific fields, e.g., the Gene Ontology), it has a large number of instances (>5000), and has been used in a number of semantic web projects. Each

process (object) in the ontology has a variety of relationships to attributes, sub-processes, exceptions, etc. and has a detailed textual description. The process query language (PQL) presented in [Klein et al. 2004] allows to pose queries which are then evaluated against the process ontology. PQL essentially allows the composition of process fragments that result in a query-by-example style specification of the sought after processes. PQL's two major statement types are ATTRIBUTE and RELATION. Attribute statements query literal properties of objects in the ontology, whereas relation statements match properties of objects, whose range are again objects (see Figure 3). As such any PQL query can be mapped to a standardized RDF-QL statement. Consequently, none of our findings are limited to the Process Handbook and PQL. They apply analogously to other semantic web query languages such as SquishQL [Miller et al. 2002].

Full-text and Keywords	PQL
"Find all processes that sell books over the internet." *Keywords:* "sell book internet"	(ATTRIBUTE "Name" OF ?process INCLUDES "sell") ∧ (ATTRIBUTE "Name" OF ?process INCLUDES "book") ∧ (RELATION ?process **HAS-MECHANISM** ?mechanism) ∧ (ATTRIBUTE "Name" OF ?mechanism INCLUDES "internet")

Figure 3: An example full-text query with its corresponding keyword and PQL query

In order to translate the DRSs generated by APE into PQL queries, we developed rewriting rules for the typical DRS structures. Each structure is first matched against a set of *ontology-model specific keyword rules*, which, when they apply, usually result in a constraint between objects (i.e., a RELATION statement). If none of those rules applies, then a set of *general-vocabulary rules* is applied, typically resulting in the comparison with a literal value (i.e., an ATTRIBUTE statement).
The *ontology-model specific keyword rules* apply if one of the keywords of the ontology (incl. its morphological or syntactic variants) appears in the DRS to be translated. For example, the expression "has a specialization" in the sentence "Which process has a specialization?" is identified as the ontology-model relationship HAS-SPECIALIZATION and, hence, translated into the following PQL statement:

RELATION ?process HAS-SPECIALIZATION ?specialization

A limitation of this approach is the choice of the vocabulary when building the ontology. In some cases we, therefore, need to include synonyms of the ontology-keywords in the rewriting rules.
Elements of the DRS not handled by the ontology-model specific keyword rules are passed to the *general-vocabulary rules.* Simple sentence structures, i.e., sentence structures not containing modifiers (such as relative sentences, and adverbs or prepositional phrases that modify verb phrases), can now be interpreted as simple literal values. For example, the verb "sell" in a sentence like "How does somebody sell consumer electronics?" is represented in the APE-DRS as "predicate(D,event, sell, A, C)." It is treated as a literal value and translated into:

ATTRIBUTE "Name" OF ?process INCLUDES "sell"

Complex structures initiate a search in the ontology-model for corresponding relationships. As an example, consider the query "How does somebody sell consumer electronics over the internet?" Here, the prepositional phrase "over the internet" indicates that a good is sold using the internet as an instrument, which is noted in the sentence's DRS (see also Example 2). As instruments, or rather their synonym "mechanisms," are included in the Process Handbook ontology-model as the HAS-MECHANISM relationships (or properties) we can translate the phrase into the following PQL statement:

RELATION ?process HAS-MECHANISM ?mechanism
ATTRIBUTE "Name" OF ?mechanism INCLUDES "internet"

If the search in the ontology-model results in no corresponding relationships, then the structure is reduced to a simple structure by treating the modifiers as literals resulting in an ATTRIBUTE statement.
Unfortunately, a full discussion of all rewrite rules is beyond the space limitations of this paper. Even so, we will try to convey the extent of the rules discussing three realistic query examples. For each example we show the ACE sentence, its DRS generated by APE, and the resulting PQL query. Example 1 shows the application of the simple general-vocabulary rules. Here the lexical elements "consumer," "electronics," and "sell" are being treated as literal values. Note that the compound "consumer electronics" is divided into its parts to improve recall. Example 2 illustrates the combination of simple and complex structures treated by the general-vocabulary rules. Finally, Example 3 uses a combination of ontology-model specific keyword rules and both types of general-vocabulary rules. Here the structure "Which sales process…" results in the first two statements of the PQL query, which can be interpreted as "Find all processes which are sales processes and which have a subtask that…" Note that the straightforward ontology-based translation of ACE sentences to PQL queries allows the user to grasp the system-inherent logic rather then having the system "guess" the user's intention based on some heuristics.

ACE	APE-DRS	PQL
How does somebody sell consumer electronics?	A B C D structure(A, dom) object(C, **consumer_electronic**, object) structure(C, atomic) quantity(C, cardinality, count_unit, B, eq, 1) predicate(D, event, **sell**, A, C) modifier(D, manner, none, **how**) query(D, **how**)	(ATTRIBUTE "Name" OF ?process INCLUDES "sell") ∧ (ATTRIBUTE "Name" OF ?process INCLUDES "consumer") ∧ (ATTRIBUTE "Name" OF ?process INCLUDES "electronic")

Example 1: Transformation of „How does somebody sell consumer electronics?"

ACE	APE-DRS	PQL
How does somebody sell consumer electronics over the internet?	A B C D structure(A, dom) object(B, **consumer_electronic**, object) predicate(C, event, **sell**, A, B) object(D, **internet**, object) modifier(C, instrument, **over**, D) modifier(C, manner, none, **how**) query(C, **how**)	(ATTRIBUTE "Name" OF ?process INCLUDES "sell") ∧ (ATTRIBUTE "Name" OF ?process INCLUDES "consumer") ∧ (ATTRIBUTE "Name" OF ?process INCLUDES "electronic") ∧ (RELATION ?process **HAS-MECHANISM** ?mechanism) ∧ (ATTRIBUTE "Name" OF ?mechanism INCLUDES "internet")

Example 2: Transformation of "How does somebody sell consumer electronics over the internet?" [1]

ACE	APE-DRS	PQL
Which sales process informs its customers over the internet?	A B C D query(A, **which**) object(A, **sales_process**, object) object(B, **customer**, person) predicate(C, event, **inform**, A, B) object(D, **internet**, object) modifier(C, instrument, **over**, D)	(ATTRIBUTE "Name" OF ?process INCLUDES "sale") ∧ (RELATION ?process **HAS-PART** ?part) ∧ (ATTRIBUTE "Name" OF ?part INCLUDES "inform") ∧ (ATTRIBUTE "Name" OF ?part INCLUDES "customer") ∧ (RELATION ?part **USES-MECHANISM** ?mechanism) ∧ (ATTRIBUTE "Name" OF ?mechanism INCLUDES "internet")

Example 3: Transformation of "Which sales process informs its customers over the internet?"

4. Validation – Query Performance of a Non-trivial Example

For the validation prototype implementation we combined Prolog and Java components, as APE is programmed in SICStus Prolog and the query engine in Java (see Figure 1). Currently, ACE sentences are entered into a Java-based user interface and then passed to the Prolog-based APE using the "Jasper" Java-to-Prolog bridge. The resulting DRSs are forwarded to the rewriting framework, also implemented in Prolog, which generates the PQL queries. These are then evaluated and presented in the Java-based query engine and user interface.

[1] Unprocessed clauses such as *structure* and *quantity* are omitted in all further DRSs to improve readability.

Using the prototype we executed a number of real-world queries (including all examples used in this paper) and compared its retrieval performance with two keyword-based retrieval approaches: one using a TFIDF-style ranking [Salton et al. 1983], the other one searching for the conjunction of keywords. Both of those approaches have a proven track record of being suitable for end-users. We then hand-coded the database to find the correct results for the natural language queries.

ACE	APE-DRS	PQL
Which sales process informs its customers over the internet and avoids unwanted solicitations with an opt-out list?	A B C D E F G H query(A, **which**) object(A, **sales_process**, object) object(B, **customer**, person) predicate(C, event, **inform**, A, B) object(D, **internet**, object) modifier(C, instrument, **over**, D) object(E, **solicitation**, object) property(F, **unwanted**, E) predicate(G, event, **avoid**, A, E) object(H, **opt_out_list**, object) modifier(G, instrument, **with**, H)	(ATTRIBUTE "Name" OF ?process INCLUDES "sale") ∧ (RELATION ?process **HAS-PART** ?part) ∧ (ATTRIBUTE "Name" OF ?part INCLUDES "inform") ∧ (ATTRIBUTE "Name" OF ?part INCLUDES "customer") ∧ (RELATION ?part **USES-MECHANISM** ?mechanism) ∧ (ATTRIBUTE "Name" OF ?mechanism INCLUDES "internet") ∧ (RELATION ?part **HAS-EXCEPTION** ?exception) ∧ (ATTRIBUTE "Name" OF ?exception INCLUDES "unwanted") ∧ (ATTRIBUTE "Name" OF ?exception INCLUDES "solicitation") ∧ (RELATION ?exception **IS-AVOIDED-BY** ?handler) ∧ (ATTRIBUTE "Name" OF ?handler INCLUDES "opt-out") ∧ (ATTRIBUTE "Name" OF ?handler INCLUDES "list")

Example 4: Transformation of "Which sales process informs its customers over the internet and avoids unwanted solicitations with an opt-out list?"

For the non-trivial query presented in Example 4 the database contained four correct answers. Our NLP query interface returned three objects, all correct results, missing only one. The TFIDF-ranking found the correct elements at the 2nd, 35th, 47th, and 183rd positions. The simple keyword matcher returned no objects as the conjunction of all keywords overconstrained the query. This example indicates that our approach provides a performance akin to logic-based retrieval engines, which usually outperform keyword engines' precision and recall, while maintaining natural language simplicity.

5. Limitations, Related, and Future Work

We can think of three limitations to the work presented in this paper. First, the use of a controlled language imposes a cost on the user. The language either has to be learned or the system has to generate meaningful error messages that are end-user friendly. Some users might be discouraged from using a system with these limitations, but experience with ACE (and other controlled languages such as Boeing Simplified English [Wojcik 2002]) has shown that learning a controlled language is still much easier than learning logic. Furthermore, some projects are currently developing query interfaces that will help people to write correct controlled English sentences by guiding them as they write [Schwitter et al. 2004].

Second, our current prototype requires some manual adaptation of the rewrite rules when using it with a new ontology (or knowledge base). Given our experience with hand-adaptation we found that most of the time an inspection of the meta-model was sufficient and we believe that the rules could be automatically generated based on the ontology structure.

Last but not least, the exemplary evaluation shown in this paper is clearly limited and can only provide an idea of the potential of this approach. Consequently, the approach needs to be thoroughly evaluated. This evaluation would include giving people retrieval tasks and comparing their performance using our front-end to other semantic web query tools (using other techniques such as plain logic, query by example, etc.). Furthermore, we would have to investigate how people's retrieval performance with the described tools is related to their background.

In our search we didn't find any other application of controlled natural language querying of semantic web content. Furthermore, we found that work on natural language interfaces to data bases (not ontologized knowledge bases) has largely tapered off since the 80's [Androutsopoulos et al. 1995] even though the need for them has become increasingly acute. The most closely related work we found was the PRECISE project [Popescu et al. 2003], which proposes a natural language interface to relational databases. PRECISE uses a data-base augmented tokenization of a query's parse tree to generate the most likely corresponding SQL statement. It is, consequently, limited to a sublanguage of English, i.e., the language

defined by the subject area of the data base. In contrast our approach limits the possible language constructs and not the subject domain. Obviously, our front-end will not return any useful answers, when none can be found in the ontology. It will, however, be able to generate an appropriate PQL statement. We hope to be able to include an empirical comparison between the two approaches in our future work.
The approach presented in this paper is clearly in its infancy. While ACE has been under development for many years, the ontology-based transformation rules are very new. Nevertheless, we believe that people's familiarity with natural languages might be the key to simplify their interaction with vast ontologies and that our approach, therefore, has the promise to provide an important step in bridging the gap between the semantic web and its users.

6. References

Androutsopoulos, I., Ritchie, G.D., and Thanisch, P. "Natural Language Interfaces to Databases - An Introduction," *Natural Language Engineering* (1:1) 1995, pp 29-81.

Bonin, J. von, "From Discourse Representation Structures to Process Query Language - A Controlled Natural Language Front-end to the Process Handbook," Diploma Thesis, Department of Informatics, University of Zurich, 2004.

Fuchs, N.E., Hoefler, S., Schneider, G., and Schwertel, U. "Discourse Representation Structures of ACE 4 Sentences," IfI-2004, Technical Report, Department of Informatics, University of Zurich, 2004.

Fuchs, N.E., Schwertel, U., and Torge, S. "A Natural Language Front-End to Model Generation," *Journal of Language and Computation* (1:2) 2000, pp 199-214.

Kamp, H. "A Theory of Truth and Semantic Representation," in: *Formal Methods in the Study of Language*, J.A.G. Groenendijk, T.M.V. Jansson and M.B.J. Stokhof (eds.), Mathematisch Centrum, Tract 135, Amsterdam, 1981, pp. 277-322.

Kamp, H., and Reyle, U. *From Discourse to Logic: Introduction to Modeltheoretic Semantics of Natural Language*, Kluwer, Dordrecht, Boston, London, 1993.

Klein, M., and Bernstein, A. "Towards High-Precision Service Retrieval," *IEEE Internet Computing* (8:1), January 2004 2004, pp 30-36.

Malone, T.W., Crowston, K., Lee, J., Pentland, B., Dellarocas, C., Wyner, G., Quimby, J., Osborn, C., Bernstein, A., Herman, G., Klein, M., and O'Donnell, E. "Tools for inventing organizations: Toward a handbook of organizational processes," *Management Science* (45:3) 1999, pp 425-443.

Miller, L., Seaborne, A., and Reggiori, A. "Three Implementations of SquishQL, a Simple RDF Query Language," The International Semantic Web Conference, Sardinia, Italy, 2002, pp. 423-435.

Popescu, A.-M., Etzioni, O., and Kautz, H. "Towards a Theory of Natural Language Interfaces to Databases," 8th International Conference on Intelligent User Interfaces, Miami, FL, 2003, pp. 149-157.

Salton, G., and McGill, M.J. *Introduction to modern information retrieval*, McGraw-Hill, New York, 1983.

Schwitter, R., and Tilbrook, M. "Dynamic Semantics at Work," JSAI, Kanazawa, Japan, 2004.

Spoerri, A. "InfoCrystal: A visual tool for information retrieval management," Second International Conference on Information and Knowledge Management, Washington, D.C., 1993, pp. 11-20.

Wojcik, R.H., Personal Communication, Location?? 2002