Natural Language Front-Ends to Databases: Design and the Customisation Bottleneck

Anne De Roeck
University of Essex Department of Computer Science
Wivenhoe Park Colchester CO4 3SQ
e-mail : deroe@essex.ac.uk

1. SQUIRREL: Motivation and Design.

NLFE to databases have failed in a commercial context, largely because of two reasons. Current approaches to the management of ambiguity by relying on inference over a world model create ungoing customisation requirements. Furthermore the design of NLFEs is subject to constraints which research in CL/ NLP does not address. In particular, standard parsing techniques (including "robust" ones) require complete lexica and cannot be deployed because new data would create a constant need for dictionary update.

The SQUIRREL [1] system (SERC Grant GR/E/ 69485) addresses some of these problems: its design reduces customisation effort as words are interpreted without reference to world models. The lexicon is assumed to be incomplete: unknown words are given interpretations by exploiting typing information contained in the datamodel. In addition, SQUIRREL demonstrates that NLFEs can allow for interrogation of integrity constraints, usually invisible to users. It is important to note that no "new" aspects of standard database management systems are involved

SQUIRREL intends to explore to what extent the state of the art in NLP/CL and Formal Semantics can be exploited in the design of NLFE to relational databases, under constraints imposed by good sofware engineering protocol. It aims to develop a modular, portable design, to plug in to public domain database technology, requiring minimal customisation.

SQUIRREL consists of a series of mappings translating NL expressions into SQL. Its highly modular design allows parts of the system to be ported without affecting other parts. Expressions in English are assigned syntactic and semantic representations on the basis of a lexicon and a context-free feature based grammar. The lexicon is incomplete: unknown words are assigned tentative categories by the (bi-directional chart) parser. Syntactic and semantic rules operate in tandem. Semantic representations are cast in Property Theory (PT) [2], delivering "intensional" objects. These are assigned extensions in the form of first order logic (FOL) expressions. So far, the representations are independent from the domain model of any database in question.

The FOL expressions are translated into the domain relational calculus (DRC), by rules exploiting the logical structure of the FOL formulae, and a domain model. The resulting expressions are translated into SQL by a simple syntactic transduction.

The design offers several cut-off points at which modules can be re-deployed. The lexicon and grammar, currently written for a subset of English, can readily be customised for any language for which a context-free feature based grammar exists. The step via PT offers a second point where the system can be deployed to applications other than database interfaces. The mapping into the DRC makes it possible to port the system to any relational query language.

The real advance made in this system is the economy of its datamodel. It sets out how each word in the dictionary is to be understood w.r.t. the current database by direct mapping: no world knowledge or inference is required. Unknown words are filled in by typing constraints associated with domains in the datamodel.

No loss of expressiveness is entailed: this is hardly surprising as all a world model would seek to do is to (i) exaggerate ambiguity w.r.t. how a user might perceive the world, in order to (ii) reduce that ambiguity w.r.t. what the current database can provide. Under this view, step (i) is totally superfluous. The resulting gain in customisation effort is paramount.

SQUIRREL's ambiguity management strategy is to offer users a choice between all interpretations that have survived the mapping into SQL. Note that at each stage in the mapping, alternative representations may emerge, or existing ones may die off. The most powerful disambiguation tool is the exploitation of typing constraints associated with the database itself.

2. Modality: the spin-off

SQUIRREL demonstrates that a NLFE can supply information which is not open to even proficient query language users. Relational databases are associated with integrity constraints to provide consistency of data across modifications over time. These constraints are not visible to users. It is possible to view such constraints as governing "possible" legal states of the database, the current database being one. As such, they can be used to answer modal queries about alternative states of affairs. When SQUIRREL is faced with a modal query, it attempts an update (via SQL), which would change the database into the required state. If the update is rejected, it collects feed-back as to which constraints have been violated and offers it to the user. By doing this, the system turns any database with integrity constraints into a "knowledge" base, without the need for explicit inference.

References

[1] De Roeck, A., C. Fox, B. Lowden, R. Turner and B. Walls, *A Natural Language System based on Formal Semantics*, International Conference on Computational Linguistics, Penang, Malaysia, 1991.

[2] Turner, R. *A Theory of Properties*, Journal of Symbolic Logic, Vol 52 no2., 1987