

# Guiding the User: An Ontology Driven Interface

Sean Bechhofer, Robert Stevens, Gary Ng, Alex Jacoby and Carole Goble  
Department of Computer Science  
University of Manchester  
Oxford Road  
Manchester M13 9PL

## Abstract

*We describe a novel query interface allowing the construction and manipulation of Description Logic expressions. The construction process is driven by the content of a conceptual model, guiding the user towards appropriate choices and providing a lucid interface.*

## 1 Introduction

A major problem in developing query interfaces is how to guide users in what queries they can ask. Users need knowledge of what it is possible to ask in a particular domain. Having to recall the nature of the data within a repository, as well as the query language itself, means that querying and exploration of data will be difficult. In this paper, we describe how using a mechanism that defines when relationships between concepts in a domain ontology may exist, allows a user interface to be generated that offers a user what it is possible to ask or say about a concept. In this way, rich, complex, but only meaningful questions can be formed. Having the possibilities for query construction laid before the user at each point allows queries to be either generalized or specialized, relying on recognition rather than recall of knowledge about a domain at any point.

Bullock [8] identifies a need for *lucidity* in information systems – a system should supply the user with an idea as to what is available, and the next steps that can be taken. This is borne out by Bates [3], who suggests that when searching, users begin by asking broad questions which are then gradually refined. Grice's [10] maxim of *Relevance* similarly states that user interfaces should provide relevant contributions.

A conceptual model can provide a space within which a user can navigate when constructing queries. Description Logics (DLs) are knowledge representation schemes which have been advocated as suitable frameworks for metadata [4]. The hierarchical compositional models provided by a

DL have the potential to support complex incremental manipulations of query expressions [6].

The TAMBIS project [1] aims to integrate biological information sources by acting as a mediator and presenting the illusion of a single data source. It achieves this through the use of a conceptual model (the Tambis Ontology [2]), which represents the metadata of the underlying data sources using a description logic. Queries are phrased against this model and are then rewritten to the appropriate sources.

In the STARCH project [4], a conceptual model represented in a DL is used as a source of terms for the description of subject content in picture archives, taking the place of mechanisms such as keyword collections or thesauri. The rich structure of the DL helps support flexible querying and navigation through the information space, via the use of the tools described here.

Interacting with a DL, however, is not always easy. In the past, systems have used textual interfaces, where the user requires not only an understanding of the underlying representation but also its particular concrete syntax. The provision of user tools is crucial in both of the above projects, which are aimed at users with no particular experience with knowledge representation formalisms.

In order to insulate the user from the representation, we have developed a suite of user tools which facilitate the construction and manipulation of DL expressions or queries. These tools are driven by constraints known as *sanctions* which are added to the DL model, and which describe the meaningful compositions which can be built. The need for lucidity or guidance can be supported through the use of sanctions as described in this paper. Both TAMBIS and STARCH make extensive use of the tools described in this paper.

## 2 Description Logics

Description Logics (DLs) are a family of class-based knowledge representation languages stemming from KL-

ONE, which allow the construction and representation of conceptual models. A DL model is based on notions of *concepts* which represent classes of objects with similar characteristics, *individuals* which are instances of those concepts, and *roles* which represent relationships between individuals.

A model is grounded on a collection of primitive concept definitions along with assertions about the subsumption (or kind-of) relations between them. Concept forming operators allow the construction of composite concept descriptions. Key to the use of a DL are a collection of services which allow us to reason about these concept descriptions. These services concern:

**Satisfiability** Checking whether a particular description is consistent;

**Subsumption** A description A subsumes another B if and only if all instances of B are necessarily instances of A. Subsumption formalises the notion of “kind-of”;

**Classification** Using subsumption, we can build classification lattices of concept descriptions;

**Retrieval** Given a concept description, we can retrieve all the individuals which are instances of that concept, in effect answering a query. Query containment is provided through the classification hierarchy or lattice.

## 2.1 The GRAIL Description Logic

GRAIL[13] is a DL developed by the Medical Informatics Group at Manchester University, and is the knowledge representation language currently used in both the TAMBIS and STARCH projects. It has a restricted set of concept formation operators and the addition of a mechanism for constraining the construction of composite concepts, known as *sanctioning*. Sanctioning plays a major part in the process which drives the construction of interfaces.

To provide some motivation for the need for sanctions, consider the following example. In general, **Biopolymers** have components which are **Motifs**. This is not to say that every biomolecule can have any motif as a component, but it is not unreasonable to consider the notion of a Biopolymer with a Motif component. The idea of anything other than a Biopolymer having a Motif component is not a reasonable notion. At a more specific level we can say that **Proteins** have components  $\alpha$ -helix, allowing the description of particular specializations applying only to Protein. Again, at the specific level, we wish to avoid misnomers such as **Nucleic acid** with an  $\alpha$ -helix component.

A sanction is a constraint that says that a concept can be combined with another in the context of a particular role to form a new description. Without a sanction, the composition cannot be formed. In order to achieve the varying levels of granularity described above, GRAIL provides two kinds of sanctions known as *grammatical* and *sensible*

sanctions. Grammatical sanctions represent the high level or general relationships which hold between concepts, such as **Biopolymer** and **Motif** above. Sensible sanctions represent relationships at the more specific level and allow formation of instantiable descriptions - in the example above the component relation would be sanctioned sensibly between **Protein** and  $\alpha$ -helix. Compositions which are only sanctioned at the grammatical level cannot be instantiated, enabling the formation of general queries, which can be specialized later. Sanctions are inherited down the subsumption hierarchy, and a grammatical sanction must be in place before a sensible sanction can be asserted.

GRAIL is implemented as a *Terminology Server* [5], providing access to a range of terminological services and operations. This separation of the core terminological services provides a clean split between the underlying representation and its use by client applications (such as the interfaces described here).

## 3 Data Entry and Query Formulation

Much work has been done on the use of GRAIL models to drive data entry interfaces. This began with PEN & PAD, has evolved from experiments with user interface requirements and now forms part of the latest version of a major computer package for General Practitioners[12]. The work focused primarily on data *entry* rather than *query* formulation. The data forms are driven by the sensible sanctions in the model, ensuring that the options available for input correspond to compositions that exist.

When we consider queries, however, the more abstract concepts permitted due to the grammatical sanctions are important. Although the concept **Biopolymer** with a **Motif component** is abstract in the sense that it is never directly instantiated, and is thus too general for use in a data entry context, it *does* form the basis of a valid query, as it subsumes concepts (such as **Protein** with an  $\alpha$ -helix **Motif**) which have instances in databases.

The issue here is that the two questions:

- What can I *say* about a concept X?
- What can I *ask* about a concept X?

are different. The first question is concerned with the specializations which can actually be built, while the second question needs to be answered at a more abstract level, allowing the use of more general compositions. When entering data, there is a need for precision. Stating the general is inappropriate. When querying, however, it is often useful to start with a general description and subsequently refine [3, 8].

## 4 Query Manipulation

Once an initial query has been formed, there are a variety of manipulations or reformulations that can be performed on the query.

**Restriction.** A query can be specialized through the addition of further criteria (role-filler pairs). Further criteria can be added to the description applied to the topic of the query. For example, a request for Protein (in other words, requesting all known proteins), could be specialized to a request for Protein which functions in Metabolism (returning a subset of all known proteins).

**Widening.** Complementary to restriction, criteria can be removed from a composite query, say moving from the query Protein which functions in Lactation and has a function as a Receptor to Protein which has a function as a Receptor.

**Replacement.** The topic of the query can be replaced by a more specific query. For example, we can replace Protein with Enzyme. Alternatively, a specific query, say Methylation Site could be replaced with a more general expression such as Motif.

These manipulations are not necessarily exclusive. Replacement of the topic could result in the restriction or widening of the query. There may be several routes to a particular specialization. Making this clear to the user is an issue in need of further investigation.

As well as the *global* manipulations described above, we can perform *local* operations, where the values of criteria can be themselves restricted or widened. When manipulating subexpressions, a further option becomes available:

**Sibling Replacement.** We may wish to allow replacement of subqueries with sibling concepts, say moving from Motif which functions in Lactation to Motif which functions in Olfaction.

All these manipulations and replacements are controlled by sanctioning, restricting the options presented for specialization/replacement, and ensuring that only reasonable queries are built.

Figure 1 shows a sample screen shot of the TAMBIS interface taken from an applet version which is currently undergoing user evaluation. A freely accessible version of this applet and videos of TAMBIS in use can be found at <http://img.cs.man.ac.uk/tambis.html>. The figure shows a complex query, together with a restrict window which offers the user a selection of restrictions which can be applied at this point in the query building process. The current query is shown in the centre of the window, broken down into its constituent parts, with indentation used to indicate nested structure. The restrict window allows selection of role/filler pairs, where the choices shown are dependent on the sanctions currently applying to the concept. These forms are generated dynamically based on the con-

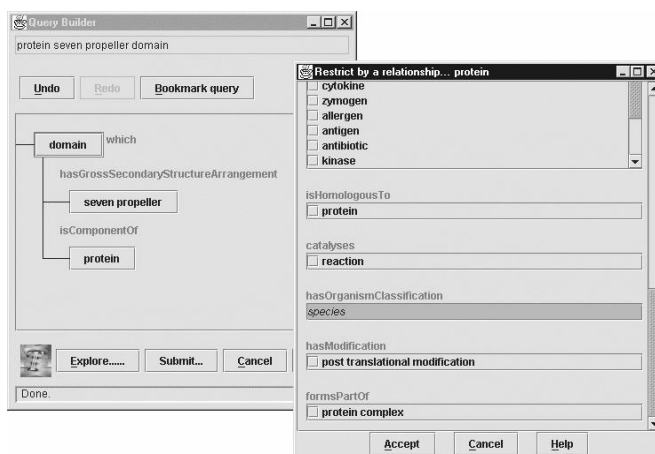


Figure 1. A query builder window

tent of the model and the current query topic. The example shows a query asking for domains which have seven propeller structure arrangement and are components of protein. The restriction window shows restrictions which can be applied to the protein subquery.

In addition to the query builder, a number of other tools are provided, allowing access to the manipulations described above. A main window provides access to bookmarks, and allows the user to make simple wild card searches to obtain initial entry points. An explorer provides a “read-only” view of the model, with the current focus displayed in the centre of the screen surrounded by closely related concepts such as parents and children. Applicable relationships (governed by sanctions) are also shown. A replacement tool allows replacement of role fillers within the query builder. The tool is similar to the explorer and presents a small “window” on the model showing those concepts closest to the current selection which can be used as replacements.

## 5 Discussion

DLs can be useful for representing the kinds of metadata found in domains such as bioinformatics. However, simply supplying access to a DL model is unlikely to prove fruitful. We must also consider how the user will interact with the language. By *controlling* the ways in which expressions and compositions can be constructed, we can help control the compositions that the user can make, and can *guide* the user in their navigation through the conceptual space in the search for appropriate query expressions.

The approach described here provides powerful operations for the construction and manipulation of DL expressions. Initial reaction from the users of our prototypes –

molecular biologists with no experience of DLs – has been positive, and a formal evaluation of the interface will take place in the near future. There are, however, still many areas in need of exploration.

An important aspect of dynamic query is the provision of *feedback* informing the user of the progress of the query and guiding toward the possible actions which can be performed. This can be separated into two levels. At the *data* level, the emphasis is on feedback concerning the answering of the query. Such an approach has been used with traditional databases [9]. The IMACS project [7] used a CLASSIC knowledge base to support data mining and knowledge discovery, providing more sophisticated feedback.

Alternatively, we can provide feedback at a *meta* or *schema* level, constraining and guiding the user based on knowledge about the information model – for example offering suitable options for specialization of a query, while preventing the formation of nonsensical queries.

The sanctions described in this paper allow us to provide this schema level feedback, used extensively in the TAMBIS interface. As TAMBIS uses a loosely couple approach, with queries answered through a rewriting process, feedback is not available at the data level. In STARCH, however, we envisage a closer coupling of the DL with the instance space, allowing us to provide data level feedback, using the classification of the instance space.

The feedback provided sits well with the four maxims of Grice [10]. As introduced above, the maxim of *Relevance* states that the interface should provide relevant contributions, while that of *Quantity* states that contributions should be as informative as is required for the current purposes of the exchange and should not be more informative than is required. By limiting, for example, replacement and restriction options in TAMBIS options, the interface does not deluge the user with spurious options when offering alternatives. In addition, the sanctioning mechanism only allows what is biologically sensible to be formed as a query in TAMBIS. In contrast, [11] criticises database query interfaces as being in conflict with Grice's maxims.

The interaction of the sanctions along with manipulation operations poses some interesting problems. When sub-queries are replaced, other parts of a query may go out of scope or become unsanctioned. Techniques are required to manage this interaction and report to the user when manipulation has side effects.

## 6 Conclusion

The use of sanctioning within an ontology has allowed us to develop query interfaces that comply with standard user interface design principles. Our ontologies can be built such that they show what it is possible to either say or ask about a concept in a domain. We feel that this can give a

lucid interface that complies with notions of relevance and quantity.

**Acknowledgements:** TAMBIS is funded by Zeneca Pharmaceuticals and the BBSRC/EPSRC Bioinformatics programme, whose support we are pleased to acknowledge. STARCH is funded by the EPSRC.

## References

- [1] P. Baker, A. Brass, S. Bechhofer, C. Goble, N. Paton, and R. Stevens. TAMBIS: Transparent Access to Multiple Bioinformatics Information Sources. An Overview. In *Intelligent Systems for Molecular Biology ISMB98*, June 28-July 1, 1998 1998.
- [2] P. G. Baker, C. A. Goble, S. Bechhofer, N. W. Paton, R. Stevens, and A. Brass. An Ontology for Bioinformatics Applications. *To Appear in: Bioinformatics*, 1999.
- [3] M. Bates. Indexing and Access for Digital Libraries and the Internet: Human Database and Domain Factors. *Journal of the American Society for Information Science*, 49(13):1185–1205, 1998.
- [4] S. Bechhofer and C. Goble. Classification Based Navigation for Picture Archives. In *IFIP WG2.6 Conference on Data Semantics, DS8*, pages 291–310, 1999.
- [5] S. Bechhofer, C. Goble, A. Rector, W. Solomon, and W. Nowlan. Terminologies and Terminology Servers for Information Environments. In *Proceedings of STEP '97 Software Technology and Engineering Practice*, 1997.
- [6] A. Borgida. Description Logics are not just for the Flightless-Birds: A New Look at the Utility and Foundations of Description Logics. Technical report, Rutgers University, 1992.
- [7] R. Brachman, P. Selfridge, L. Terveen, B. Altman, A. Borgida, F. Halper, T. Kirk, A. Lazar, D. McGuinness, and L. Renick. Integrated Support for Data Archaeology. *International Journal of Applied and Cooperative Information Systems*, 2(2):159–185, 1993.
- [8] J. Bullock. *Informed Navigation: Description Logic Based Hypermedia Linking*. Phd., University of Manchester, 1999.
- [9] G. Ellis, J. Finlay, and A. Pollitt. HIBROWSE for Hotels: Bridging the Gap Between User and System Views of a Database. In P. Sawyer, editor, *Proceedings of IDS 2*. Springer-Verlag, 1994.
- [10] H. P. Grice. Logic and Conversation. In P. Cole, editor, *Syntax and Semantics 9:Pragmatics*. Academic Press, 1978.
- [11] D. Haw, C. Goble, and A. Rector. The Pragmatics of Naive Database Enquiry. In *British National Conference on Databases, BNCOD 13*, number 940 in Lecture Notes in Computer Science, pages 79–100, Manchester, UK, 1995. Springer.
- [12] J. Kirby. PEN & PAD: The Next Generation. In *Primary Health Care Specialist Group*, Cambridge, UK, 1995.
- [13] A. L. Rector, S. K. Bechhofer, C. A. Goble, I. Horrocks, W. A. Nowlan, and W. D. Solomon. The GRAIL Concept Modelling Language for Medical Terminology. *Artificial Intelligence in Medicine*, (9):139–171, 1997.