Delivering Terminological Services

Sean Bechhofer **Carole Goble**
Department of Computer Science
University of Manchester
Oxford Road
Manchester M13 9PL
{seanb, cag}@cs.man.ac.uk
Tel: +44 161 275 6145

ABSTRACT

Terminologies or controlled vocabularies are important as they provide a framework within which communities can share knowledge. We describe three applications using a terminology represented in a Description Logic and delivered via a Terminology Server and suggest that such a service oriented architecture is essential to the adoption of terminologies within the component based architectures of today's software environments.

1 Introduction

The use of terminologies or controlled vocabularies is widespread within communities which have a need or desire to unambiguously share standardised information. Such terminologies are important as they provide a framework within which communities can communicate and express ideas in a consistent manner.

A terminology is a collection of terms or concepts with relationships between them, particularly the *is-a* or *subsumption* relationship, which is used to build a *classified hierarchy* of concepts. Classification supports notions such as generalization and specialization – descriptions can be refined, incrementally adding more detail when necessary, and queries can use the subsumption hierarchy to ask general questions.

The application of terminologies is now receiving attention from the more general information management community because they appear particularly powerful for describing semi-structured information flexibly but with coherence and rigour. Furthermore, they are extensible in that data instances can be incrementally described – extremely useful if we cannot always predict how we wish to describe a data instance at the outset. An instance can be described using the terminology in a flexible way – this effectively types the instance with the description. As the description changes, so does the type. Terminologies are a promising representation for metadata for several areas.

Content-description of semi-structured information: the description of the content of information instances such as documents or images cannot necessarily be strictly typed at the outset unlike classical database schema where instances populate a predefined, static and complete schema. Application examples include: digital libraries and document/image archives [19], broadcast video archives [10], the World Wide Web and open hypermedia systems [7].

The description of highly complex domains: domains such as medicine and art are so complex, and the variability of description of instances is so great, that a static type system is impracticable without massive loss of descriptive richness [13].

Ontologically-supported mediation between diverse information sources: a terminology can be used to represent a common ontology that different information sources can map to. The common description must be rich enough to be all encompassing, and the mappings will be incomplete, imprecise and changeable as sources have missing data or are altered. As sources are described they should be classified with respect to one another to support imprecise cross-mappings [1].

Schema consistency checking: the consistency of ER schemas can be checked by translating the schema into a terminological framework with a rigorous semantics such as a Description Logic [8].

Most of the above applications have similar requirements of their metadata:

- the data to be described is *complex* and frequently unstructured or semi-structured requiring a rich and expressive metadata model;
- the data is gathered and processed *incrementally* over time, collecting metadata (e.g. images collect annotations) so that the fundamental assumption that all metadata is known at data capture time is flawed;
- information is *incomplete* and *imprecise* as we cannot always predict how we wish to describe a data instance such as a document, unlike conventional databases where the data instances are strictly typed at the outset;
- data descriptions are *dynamic* and *evolutionary*. Flexible extensible descriptions are required as the same

data may be reused from many different perspectives, and dynamically classified by many different, unpredictable, and possibly contradictory interpretations of the same contents, requiring data to be multiply classified as descriptions are elaborated and data is reused;

- incomplete descriptions lead to *imprecise queries* and *incomplete results* where the answer to a question might well just be an initial reduction of the search space. For example, (1) we may wish to present an example description and answer the question "retrieve objects that are similar to this" where the similarity is a metric of how closely the descriptions are classified w.r.t. to one another; (2) upon presentation of a request for an image containing a male politician, if the answer is empty we would like to be offered images of politicians as a more general alternative by relaxing the query constraints.

We propose the use of terminologies implemented through Description Logics for describing metadata, and a service-oriented architecture that encapsulates the description logic in a **Terminology Server** – a resource that delivers a range of terminological services.
In this paper, we will describe three applications that characterise the features described above. The **TAMBIS** project integrates biological information sources, using a Description Logic model as a mediator. In **STARCH**, we use a Description Logic to provide terms for semantic metadata in picture catalogues. Finally, the **GALEN** and **GALEN-IN-USE** projects provide terminological services for clinical applications in the medical domain. Descriptions of the projects are used to illustrate the architecture.

2 Description Logics

Description Logics (DLs) are a family of knowledge representation languages that allow reasoning with compositional structured information. In particular, a DL supports hierarchical classification through the use of a well-defined notion of subsumption. For a full description of DLs and their uses, see [5]. DLs are closely related to propositional modal and dynamic logics – recent work has provided a sound formal basis for several DLs along with results concerning their complexity and expressiveness [6, 18].
A DL models an application domain in terms of concepts (classes), roles (relations) and individuals (objects). The domain is a set of individuals, and a concept is a description of a group of individuals that share common characteristics. Formally, a concept is interpreted as a subset of the individuals which make up the domain. Roles model relationships between, or attributes of, individuals, and a role is interpreted as a set of binary tuples relating pairs of individuals. Compositional concept descriptions can then be built up using recursive term constructors, where terms are concepts or roles. Individuals can be asserted to be instances of particular concepts and pairs of individuals can be asserted to be instances of particular roles.

2.1 Reasoning Services

DLs provide a variety of services [2] that make them particularly attractive as models for describing semi-structured and complex information [5].
Subsumption The power of DLs is derived from the automatic determination of subsumption between compositional descriptions. Given two conceptual definitions A and B, we can determine whether A subsumes B, in other words whether every instance of B is necessarily an instance of A. Formally, subsumption is defined as an implicit subset/superset relationship between the interpretations of the two concepts.
Classification A collection of conceptual definitions can be organised into a partial order based on the subsumption relation. This provides a multi-axial hierarchy of definitions, ranging from the general to the specific. Primitive concepts have no characterising attributes and must be explicitly placed in the hierarchy by the system designer, but new, composed definitions have their position determined automatically. Thus classification is a dynamic process where new compositions can be added to an existing hierarchy.
Retrieval Given a concept definition, we can retrieve all the instances of that concept (which of course includes all instances of subsumed concepts).
Realization Given an individual, we can provide the most specific concepts (w.r.t. subsumption) that the individual is an instance of.

2.2 Benefits of a Description Logic

The services described above allow a DL to be used in two ways.

2.2.1 A type system.

The conceptual hierarchy can be used as a type system. The expressivity of DLs make it possible to express the semantics of information systems and are often more expressive than traditional Semantic Data Models or Object Oriented data models. Type refinement is provided automatically through the subsumption ordering on descriptions and hence provably correct subsumption algorithms can be used for type checking. By checking whether a compositional concept's description will classify we are able to verify a schema's consistency. Finally, each compositional description contains only the necessary and sufficient descriptions for the concept. Hence terms that are expressed in different ways but come down to the same description are equivalent and redundancy is reduced.

2.2.2 An indexing/query system.

When combined with an instance space, the hierarchy of concept definitions can be used as an index. DLs form a dynamic multiaxial classification scheme which supports incremental elaboration and partial information. Concepts can be incrementally specialised, with the automatic classification capabilities of a DL taking care of relationships between concepts. As more information is attributed to the description,

the concept migrates through the classification hierarchy. Imprecise querying and query generalisation/specialisation is supported by a) generalising or specialising concept terms; b) relaxing or restricting cardinality constraints or c) adding or removing terms to navigate through the concept hierarchy and explore potential relationships between terms and their associated instances. Finally, DLs are naturally suited for expressing queries and defining views, as the subsumption relationship can be used to organise queries automatically and views, hence supporting data exploration and query optimisation.

3 A Terminology Service

Interacting with DL implementations can often prove difficult. In the past, solutions involved embedding the logic in large monolithic systems with all the attached problems of maintenance and implementation, lack of consistency, and the hampering the exchange of terms between applications. A terminological resource can be used to drive, or interact with, a number of different systems and therefore should be perceived as a separate component.

We propose a move towards a service-oriented architecture and the encapsulation of a DL in a **Terminology Server** or TeS – a resource that delivers a range of terminological services. These services are used by a variety of applications such as modelling tools, query engines, data entry interfaces and so on. A terminology server approach is essential if we are to build generic "plug and play" mixed systems, for example if a semantic hypermedia system is to use the TeS as a link resolution service along with conventional link following or information retrieval services [15].

3.1 Services

The services provided by a TeS can be broken down into three main types: pure ontological or *concept* services, language or *linguistic* services, and *extrinsic information* services.

Concept Services. Along with maintaining the representation of the concepts, the services include operations which both extend and query the content of the model; these operations are commonly known as TELL and ASK [5]. Operations for extension include the introduction of new primitive concepts, the introduction of new relations, and the addition of new constraints on roles. Queries include those to determine the primitive concepts and relations; requests for the parents (generalizations) or children (specializations) of a concept definition (w.r.t. the subsumption hierarchy); usage of concepts and information about how concepts and relations can be combined to create new potential composite concepts.

Classification operations take a conceptual definition and determine its place in the hierarchy (according to the rules defined in the logic). Such an operation may also perform validity checking that the concept is consistent and fits with any constraints the model imposes – in this way the server is providing a form of type-system (see Section 2.2.1). Table 1 includes some examples of a concept service's operations.

ASK	*Type checking*: Is this a legal expression? Are these two definitions the same?
Type Based	*Type exploration*: what further can be said about the expression – i.e. how can I specialize it?
Index-based	*Classification*: Where does the expression sit in the hierarchy?
	Query relaxation: How can I generalize the expression?
	Query tightening: How can I specialize the expression?
TELL	Introducing new elementary concepts; Introducing new rules about term composition;

Table 1: Concept Services

ASK	*Language rendering*: What are the natural language expressions for this concept?
	Mediation & Translation: How does this expression correspond to some external representation I have?
TELL	Naming concept definitions; Providing maps to external representations

Table 2: Other Services

Linguistic Services. The server deals with concepts. In order to aid interaction with these concepts, and for user interfaces, they should be available in representations other than the underlying one used by the DL. Thus the server should offer linguistic services which provide the conversion to and from natural and other language expressions. The separation of conceptual and linguistic services can facilitate the support of multiple languages, by making clear the distinction between a conceptual definition and the representation used to render it.

Extrinsic Information Services. Concept and linguistic services deal with the terminology in isolation. In addition to these, we may expect to be able to relate terms from the terminology to objects from the rest of the world. As this information associated with the terminology has no effect on the underlying classification, we refer to these as extrinsic information services. These operations are of course not completely divorced from the ontological operations – services may well use the classification in order to sparsely decorate the hierarchy and allow general querying. Table 2 includes examples of linguistic and extrinsic services.

In addition, terminologies can be used as mediators between information sources – the server can provide information about the concrete representation of a conceptual definition in a particular information source. These mappings to other representations can be seen as examples of extrinsic information services [1].

One could argue that it is simply the conceptual services which should be encapsulated in the server. However, as we

discuss in Section 4, services such as linguistic services are essential if applications are to render conceptual definitions in a readable format. Similarly, the ability to relate concepts with external representations is essential if a new system is to prove compatible with existing work – domains in which a terminology server would prove useful are often those domains which already have existing terminologies. For these reasons, we feel that linguistic and extrinsic services should be considered as core services of a TeS.
By encapsulating the logic and operations as a collection of services, the terminology can be used as a resource in a distributed system; for example as a common facility in a CORBA distributed environment.

4 Applications

Here we provide a brief description of three projects which are using a DL, delivered through a TeS. We highlight some of the requirements which the sytems have and relate these to the services that a TeS can provide.

4.1 TAMBIS

There are an increasing number of biological information sources including on-line databases, tools and applications, and flat files. In order to access the sources, users must know both the location of the source, and the access method – e.g. SQL for databases, appropriate commands for tools, or some searching mechanism based on the format of a file. This is unsatisfactory, leading to underuse of these distributed sources. TAMBIS [11] aims to answer these problems by acting as a single interface to the information sources, providing the illusion of a single source. This is done through the use of a conceptual model of the biological domain, expressed in a DL. A sources and services model provides a mapping between the conceptual definitions in the DL and the existing information sources – for example recording the fact that instances of of the concept Protein may be found in database Swissprot using the query Q. User queries are then expressed in terms of the DL and need not refer to particular sources. For example the user can now ask for *Motifs which occur in Aardvark Proteins*. The system rewrites the query expression to requests to appropriate sources. Source wrappers deal with differences in source format. DLs have been used for information integration in this way in projects such as SIMS [1].
It would be unreasonable to expect users to interact directly with the DL – a form based interface is provided which allows the user to build up queries interactively and in a graphical fashion.
TAMBIS makes particular use of concept services, for navigation of the concept hierarchy, determining how concepts can be specialized and combined and the classification of composites. In addition, interfaces require text (possibly in several languages) generated from the concept expressions for use on buttons, fields etc, and mappings from natural language to concept expressions are used to provide entry points to the model.

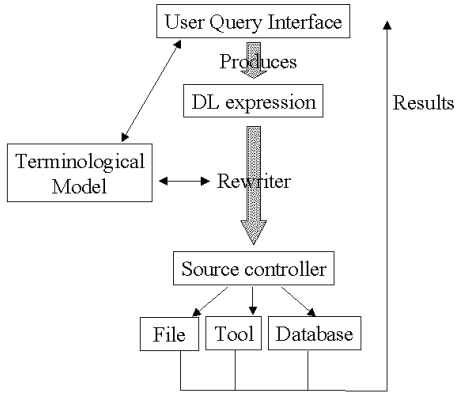

Figure 1: Tambis architecture

The sources and services model used in TAMBIS, mapping concepts to particular sources is an example of extrinsic information.
A diagram of the system architecture demonstrating the communication between the interface, rewriter and terminological model is shown in Figure 1.
The separation of the biological terminological model into a networked resource facilitates the use of the terminology in projects other than Tambis – the model is now being used in tasks such as annotation checking and function prediction.

4.2 STARCH

STARCH [3] is a project which aims to use a terminological model to represent semantic metadata for picture archives. Traditionally, keywords or free text annotations have been used to represent the content of images in catalogues, but these have their problems as discussed in [3]. Representing the terms in a DL brings advantages relating to the coherency and consistency of the model, allows the support of multiple views through multi-axial classification, and can provide a space around which the user can navigate, supporting serendipitous browsing.
The use of the concept hierarchy as an index (see Section 2.2.2) allows flexible query, in particular through the use of abstract queries and generalization
Starch has similar requirements to TAMBIS, with additional requirements of the conceptual services, in particular to do with retrieval of instances.
TourisT [7] is a related project, using the conceptual model to manage links in a semantic hypermedia system. Extrinsic services are used to manage views of the conceptual model, and linguistic services are important when visualising links in a web based browsing system.

4.3 GALEN-IN-USE

The GALEN project [16] was concerned with providing advanced terminological services for clinical applications in the medical domain – many of the ideas concerning terminology servers discussed in this paper were developed during GALEN. In the follow on GALEN-in-USE project, a TeS

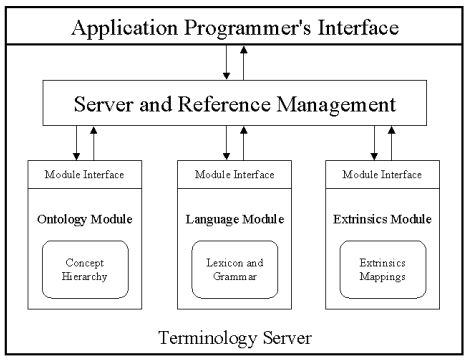

Figure 2: The architecture of a server

is being used to help in constructing the next generation of medical coding schemes [17], relying on the automated classification to improve the consistency and coherency of the static schemes. Multilingual support is provided, facilitating the sharing of terminologies between international communities, and extraction mechanisms are provided which enable local terminology snapshots to be taken and used locally, but which are coherent with others and the common model.
A fundamental part of the GALEN work is the production of a model of medical terminology. A large suite of tools was developed to aid in both the modelling process (i.e. building the models) and in the use of the models by classification centres – such tools can be seen as an important class of application in their own right. In the past, DL systems have been supplied with minimal tool sets, often based on a command line interface. Building and maintaining large conceptual models (the GALEN model has over 12,000 conceptual definitions, implying many millions of composite definitions) and relating them to existing schemes without the help of browsing and navigation tools would be an extremely difficult, if not impossible, task.

5 Architecture

The identification of classes of services leads to an internal architecture as shown in Figure 2. There are three central modules each with a rough responsibility for services described above. Each module has a well-defined interface through which other modules communicate – this allows for easy replacement of any particular collection of services. For example, different applications may wish to make use of different classification or subsumption algorithms.
Reference management and communication between modules is controlled by a server management level.
The server is a dynamic resource – rather than having applications hold on to conceptual definitions which contain all the required information, the server hands out references to concepts held in the Concept Module. When further information about a concept is required, the application will query the server. References are thus opaque and carry no structure.
We can envisage multiple Terminology Servers in a distributed environment with applications coupled to a local server occasionally using other servers. A server might also support multiple concept models. This leads to two possible forms of referencing concepts: persistent *global* references, interchangeable between servers, and *local* references for a particular server. We can also see the requirement for transient but fast session-dependent references for a particular server. Referencing policies are discussed in more detail in [4].
References provide a naming service, with local references forming a local name space (valid for a particular server) and global references a global name space (valid for all servers). These references refer to concepts in an internal representation; applications will also need to refer to them as (natural) language expressions and in terms of existing hierarchies or schemes.

6 Related Work

There are several current initiatives aimed at identifying the services which a TeS or TeS-like object should supply. These include the CorbaMED Lexical Query Service [14], the Generic Frame Protocol (GFP) [12, 9] and systems such as Ontolingua, which can be seen as a resource providing terminological models rather than terminological services.
The DL community has also recognised the advantages of component based architectures *within* the implementations of representations, with proposal for standardised architectures which will allow builders of systems to plug and play with the internals, for example selecting an appropriate reasoning module for a particular application.

7 Conclusions

Terminologies and Description Logics can be useful in a wide range of applications – however they should not be embedded in applications, but should be shared as a distributed resource. By identifying the services required and providing them through a central resource, a terminology can play an important part in a distributed information system.

8 Acknowledgements

The work described in this paper was supported by a variety of bodies including EPSRC, BBSRC, the EU and Zeneca Pharmaceuticals. The authors would also like to thanks the members of the Manchester Medical Informatics and Information Management Groups, in particular Professor Alan Rector.

REFERENCES

[1] Y. Arens, C. Knobleck, and W-M. Shen. Query Reformulation for Dynamic Information Integration. *Journal of Intelligent Information Systems*, 6(2/3):99–130, 1996.

[2] Baader, Franz and Bürckert, Hans-Jürgen and Heinsohn, Jochen and Hollunder, Bernhard and Müleer,

Jürgen and Nebel, Bernhard and Nutt, Werner and Profitlich, Hans-Jürgen. Terminological Knowledge Representation: A Proposal for a Terminological Logic. Technical report, Deutches Forschungszentrum für Künstliche Intelligenz (DFKI), 1991.

[3] Sean Bechhofer and Carole Goble. Classification based Navigation and Retrieval for Picture Archives. Submitted for publication to: 8th IFIP WG 2.6 Working Conference on Data Semantics, 1998.

[4] S.K Bechhofer, C.A. Goble, A.L. Rector, Solomon. W.D., W.A. Nowlan, and the GALEN Consortium. Terminologies and Terminology Servers for Information Environments. In *Proceedings of STEP 97, International Workshop on Software Technology and Engineering Practice*, London, 1997. IEEE Computer Society Press. 484–497.

[5] A. Borgida. Description Logics in Data Management. *IEEE Transactions on Knowledge and Data Engineering*, 7(5):671–782, 1995.

[6] A. Borgida. On the relative expressiveness of description logics and first order logics. *Artificial Intelligence*, 82:353–367, 1996.

[7] Joe Bullock and Carole Goble. TourisT: The Application of a Description Logic based Semantic Hypermedia System for Tourism. In *Proceeedings of Hypertext'98*, Pittsburgh, PA, 1998.

[8] Diego Calvanese, Maurizio Lenzerini, and Daniele Nardi. Description logics for conceptual data modeling. In *Logics for Databases and Information Systems*, pages 229–264. Kluwer Academic, 1998.

[9] Vinay K. Chaudhri, Adam Farquhar, Richard Fikes, Peter D. Karp, and James P. Rice. The Generic Frame Protocol 2.0. available from http://www.ai.sri.com/g̃fp, 1997.

[10] C.A. Goble, C. Haul, and S. Bechhofer. Describing and Classifying Multimedia using the Description Logic GRAIL. In *Conference on Storage and Retrieval of Still Images and Video IV*, San Jose, 1996. SPIE Vol 2670.

[11] Carole Goble, Norman Paton, Patricia G. Baker, Andy Brass, Sean Bechhofer, and Robert Stevens. Transparent Access to Multiple Biological Information Sources. Submitted for publication to: Journal of Intelligent Information Systems, 1998.

[12] P. D. Karp, K. Myers, and T. Gruber. The Generic Frame Protocol. In *Proceedings of the International Joint Conference on Artifical Intelligenve, IJCAI 95*, pages 768–774, 1995.

[13] W. A. Nowlan, A. L. Rector, S. Kay, B. Horan, and A. Wilson. A Patient Care Workstation Based on User Centred Design and a Formal Theory of Medical Terminology: PEN & PAD and the SMK Formalism. In *Fifteenth Annual Symposium on Computer Applications in Medical Care. Proceedings of SCAMC91*, pages 855–857. McGraw-Hill Inc., 1991.

[14] OMG. Object Management Group Corbamed Lexicon Query Services RFP Response. OMG TC Document CORBAmed/98-03-22, March 1998.

[15] J. O'Neill. *An Investigation into the use of GRAIL as a Hypermedia Authoring Tool*. University of Manchester, Department of Computer Science, 1996.

[16] A. L. Rector, P. Zanstra, W. D. Solomon, and The GALEN Consortium. GALEN: Terminology Services for Clinical Information Systems. In M.F. Laires, M.J. Ladeira, and J.P. Christensen, editors, *Health in the new Communications Age*, volume 24 of *Health Technology and Informatics*. IOS Press, 1995.

[17] J. E. Rogers, Solomon W. D., A. L. Rector, P. Pole, P. Zanstra, and E. van der Haring. Rubrics to Dissections to GRAIL to Classifications. In *MIE 97*, 1997.

[18] K. Schild. A Correspondence Theory for Terminological Logics: Preliminary Report. In *Proceedings of the 12th International Joint Conference on Artificial Intelligence (IJCAI-91)*, pages 466–471, 1991.

[19] Joachim Schmidt, Gerald Schroder, Claudia Niederee, and Florian Matthes. Linguistic and Architectural Requirements for Personalized Digital Libraries. *International Journal of Digital Libraries (JODL)*, 1(1), 1996.