Knowledge Representation meets Digital Libraries

Enrico Franconi
Department of Computer Science
University of Manchester, UK
franconi@cs.man.ac.uk
http://www.cs.man.ac.uk/∼franconi/

Abstract

In this short paper, the basic ideas behind a project on the application of Knowledge Representation formalisms and technologies for Conceptual Modelling and Query Management are presented. It is argued that good *Conceptual Modelling* is required to support powerful information access through *Intelligent Navigation*, a fundamental task in Digital Libraries.

1 Introduction

In recent years, data and knowledge base applications have progressively converged towards integrated technologies that try to overcome the limits of each single discipline. Research in Knowledge Representation (KR) originally concentrated around formalisms that are typically tuned to deal with relatively small knowledge bases, but provide powerful deduction services, and the language to structure information is highly expressive. In contrast, Information Systems and Database research mainly dealt with efficient storage and retrieval with powerful query languages, and with sharing and displaying large amounts of (multimedia) documents. However, document representations were relatively simple and flat, and reasoning played only a minor role.

This distinction between the requirements in Knowledge Representation and Databases is vanishing rapidly. On the one hand, to be useful in realistic applications, a modern Knowledge Representation system must be able to handle large data sets, and to provide expressive query languages. This suggests that techniques developed in the Database area could be useful for knowledge bases. On the other hand, the information stored in digital libraries is now very complex and with deep semantic structures, thus requiring more intelligent modelling languages and methodologies, and reasoning services on those complex representations to support design, management, and retrieval. Therefore, a great call for an integrated view of Knowledge Representation and Information Systems is emerging.

Description Logics (DL) are a very promising research area in KR with applications in Databases. The main effort of the research in DL is in providing both theories and systems for expressing structured knowledge and for accessing and reasoning with it in a principled way. Recently, basic progress has been made by establishing the theoretical foundations for the effective use of DL in Information Systems [Calvanese *et al.*, 1998c; Borgida, 1995]. This turns out to be of relevant importance also for the digital libraries field, since DL offer promising formalisms for solving several problems concerning Conceptual Data Modelling [Calvanese *et al.*, 1998c], Intelligent Information Access [Stock *et al.*, 1993; Bresciani and Franconi, 1996; Bresciani *et al.*, 2000], and Information Integration [Calvanese *et al.*, 1998a; Jarke *et al.*, 2000].

Two major problems that arise while designing and managing an information system are *conceptual modelling* and *information access*. In the digital libraries context, these problems become more complex and challenging, and require new ideas, technologies, and tools to be solved. In this short paper, the basic ideas behind an EPSRC funded (the British Research Council) project on the application of Knowledge Representation technologies for Conceptual Modelling and Query Management are presented. The aim of the project is to devise and evaluate algorithms, methodologies, techniques, and interaction paradigms to build a tool for conceptual modelling and query management of complex data repositories, based on a framework with its formal foundations on DL. It is argued that good *Conceptual Modelling* is required to support powerful information access through *Intelligent Navigation*, a fundamental task in Digital Libraries.

A tool supporting the design, the management, and the storage of conceptual schemas has been already delivered. The toool is connected in the background with a Description Logic inference server, and it is based on a formal framework and a methodology for conceptual modelling borrowed from the most recent ideas in the Description Logic research. An extension of the tool for query management and intelligent navigation is on the way, this time connected in the background with a Description Logic inference server augmented with query reasoning capabilities. The query manager is based on the recent research result on query management paradigms which exploit the knowledge in the conceptual schema.

This presentation has been divided into two Sections. In the first Section, the data model and the methodology for conceptual design will be introduced. In the second Section, the query management problem in the presence of the previously devised conceptual model will be considered: a global framework will be introduced, together with various basic tasks involved in intelligent information access and navigation.

2 Conceptual Modelling

Good *conceptual data models* put their emphasis on the correct and semantically rich representation of *complex* properties and relations that may exist between documents. They should allow for an abstract representation of documents which resembles the way they are actually perceived and used in the real world, thus shortening (with respect to the more traditional data models) the semantic gap between the domain and its representation.

Conceptual modelling deals with the question on how to describe in a declarative and reusable way the domain information of an application, its relevant vocabulary, and how to constrain the use the data, by understanding what can be drawn from it. Recently, a number of conceptual modelling languages has emerged as de-facto standard, in particular we mention Entity/Relationship (ER) for the relational data model, UML and ODMG for the object oriented data model, and XML, RDF an OIL for the semi-structured data model. Still, many such languages do not have a formal semantics based on logic, or reasoners built upon them to support the designer. Not surprisingly, conceptual modelling tasks have always been in the mainstream of KR research – see for example the research on Ontology representation and design – and can be considered now one of the main applications of KR languages and reasoning techniques. DL can be considered as an unifying formalism, since they allow the logical reconstruction and the extension of representational tools such as object-oriented data models (e.g., UML and ODMG), semantic data models (e.g., Entity/Relationship and ORM), frame-based ontology languages (e.g., OKBC, OIL and XOL) [Calvanese *et al.*, 1998c; 1999; Fensel *et al.*, 2000]. In addition, given the high complexity of the modelling task when complex documents are involved, there is the demand of more sophisticated and expressive languages than for normal databases. Again, DL research is very active in providing expressive languages for conceptual modelling (see, e.g., [Artale and Franconi, 1999; ; Franconi *et al.*, 1999; 2000; Franconi and Sattler, 1999]).

As to the basic conceptual modelling language, we have decided to adopt an extended ER (EER) conceptual data model. Basic elements of ER schemas are *entities*, denoting a set of objects called *instances*, and *relationships*, denoting a set of *tuples* made by the instances of the different entities involved in the relationship. Since the same entity can be involved in the same relationship more than once, participation of entities in relationships is represented by means of *ER-roles*, to which a unique name is assigned. ER-roles can have *cardinality constraints* to limit the number instances of an entity involved in the relationship. Both entities and relationships can have *attributes*, i.e., properties whose value belong to some predefined domain – e.g., `Integer`, `String`. Additionally, the EER model includes *taxonomic* relations to state inclusion assertions between entities and between relationships, with the possibility to specify optional *covering* and *disjointness* constraints.

The most interesting feature of the extended modelling language is the ability to completely *define* entities and relationships as *views* over other entities and relationships of the conceptual schema [Calvanese *et al.*, 1998c]. The adopted view language is $\mathcal{DLR}$ [Calvanese *et al.*, 1998a], a Description Logic over unary and n-ary relationships. $\mathcal{DLR}$ is an interesting decidable fragment of first order logic: among others, inclusion dependencies with $\mathcal{DLR}$ views can express (a) unary inclusion dependencies, (b) typed inclusion dependencies without projection, (c) existence dependencies, (d) exclusion dependencies, and (e) full key dependencies. An implementation of of the $\mathcal{DLR}$ Description Logic already exist, and it has been developed within our group [Horrocks, 1999; Horrocks *et al.*, 1999b].

The conceptual data model includes the modelling of multidimensional aggregations [Franconi and Sattler, 1999; Franconi *et al.*, 1999]. That is, the conceptual data model is able to represent the structure of *aggregated entities* and of *multiply hierarchically organised dimensions*. In a much more radical way than in UML, aggregations become first class citizens of the representation language: it is possible to describe the components of aggregations, and the relationships that the properties of the components may have with the properties of the aggregation itself; it is possible to build aggregations out of other aggregations, i.e., it is possible for an aggregation to be explicitly built on top of aggregated dimensions. The ability of representing aggregations at the conceptual level is crucial in modelling structured documents in digital libraries.

In this project, we have built a tool implementing the above EER conceptual data model, and we have devised a conceptual modelling methodology for it [Franconi and Ng, 2000; Jarke *et al.*, 2000]. Conceptual modelling is a highly iterative process. Let's assume that we start with a (possible empty) partial schema and want to add some new information, that is, some additional objects or properties of objects that a user is interested in and that thus should be modelled conceptually. The methodology should provide support for the design of the schema by making explicit all (implicit) consequences of facts that have been modelled so far. Moreover, it will also detect all inconsistencies in the schema. Likewise, the new schema that is the output of the modelling process not only contains all the facts we have explicitly modelled, but also the implicit consequences that were detected by the reasoner together with the freshly measured quality factors of the objects in the new schema. This supports the designer during the conceptual modelling process in understanding whether the schema developed so far actually captures the intended meaning. This either increases the belief in the correctness of the conceptual schema, or gives an argument for its inconsistency. We can summarise as follows the representation and reasoning support which is involved in the methodology.

- It is possible to capture important basic facets of document semantics, including the structure of complex entities and ontological dimensions such as time and aggregation.
- It is possible to check whether the global information conveyed in a schema forces some specific class to be inconsistent. Moreover, one could check the consistency of the whole schema, also with respect to possible integrity constraints.
- The system could not only check the consistency of the schema itself, but also make deductive inferences asserting new explicit facts regarding the entities and the relationships in the schema.
- Views - i.e., pre-defined descriptions, grounded on the terms of the schema - are also part of the formalism, and it is possible to reason with them. As an example, is it possible to check containment between views and organise them into an hierarchy.
- It is possible to reduce the redundancy of a schema, to discover equivalent descriptions, to reuse descriptions, and to refine the descriptions.

3 Query Management

The problems of *information access* and *query management* in digital libraries do not appear to be close to a solution, because they involve the *content* of complex documents, which is inherently difficult to understand and model, let alone handle algorithmically. The availability of large amounts of documents which are heterogeneous in structure and origin has made this problem still more pressing: in this case an excess of information can be equivalent to an absence of information. It is therefore necessary to use tools which "put the data in order", namely to organise documents into intelligible and easily accessible structures and return answers at various levels of detail to support analytic and decisional activities.

Only recently has KR research started to have an interest in query processing and information access. Recent work has come up with advanced reasoning techniques for query evaluation and rewriting using views under the constraints given by the conceptual schema – also called view-based query processing. This means that the notion of accessing information through the navigation of an Ontology modelling the document's domain – which can be seen as a knowledge base – has its formal foundations.

Our proposal considers DL for formalising not only the conceptual model but also the query processing as well ([Bresciani and Franconi, 1996; Franconi, 1997; Stock *et al.*, 1993]). By reusing and adapting the results known in the literature, this choice has the advantage that it is possible to *reason* on the query, and to see the query evaluation process as *view-based query processing* [Ullman, 1997; Calvanese *et al.*, 2000b]. The conceptual schema as defined in the previous section can be seen as a set of constraints over a vocabulary which is usually richer that the logical schema of the document base it is modelling. In some sense, quite often the conceptual schema plays the role of an ontology of the domain, very close to the user's rich vocabulary, rather than of a set of constraints over the poor logical vocabulary structuring the data. With this perspective in mind, the user would prefer to query the information system using the rich vocabulary of the conceptual schema. The vocabulary of the basic data could be seen in turn either as a subset of the conceptual vocabulary – this is the simplistic view – or more generally as a set of (materialised) views over the vocabulary of the conceptual schema. However, in this case we have to solve the problem of view-based query processing. The problem requires to answer a query posed to a database – the one defined by the conceptual schema – only on the basis of the information in a set of (materialised) views, which are again queries over the same database. In the process, the information contained in the conceptual schema of the document base should be of course taken into account.

There are two approaches to view-based query processing, namely query rewriting (see, e.g., [Beeri *et al.*, 1997]) and query answering (see, e.g., [Abiteboul and Duschka, 1998; Calvanese *et al.*, 2000a]). In the former approach, we are given a query Q, a set of view definitions characterising the actual data, and a set of (conceptual) constraints – all over the conceptual vocabulary – and the goal is to reformulate the query into an expression, the rewriting, that refers only to the views, and provides the answer to Q. Typically, the rewriting is formulated in the same language used for the query and the views. In the latter approach, besides Q, the view definitions and the constraints, we are also given the extensions of the (materialised) views. The goal is to compute the set of tuples that are implied by these extensions, i.e., the set of tuples that are in the answer set of Q in all the databases that are consistent with the views and the constraints.

This framework can be used to characterise several aspects of document management. In query optimisation, view-based query processing is relevant because using the views may speed up query processing. In data integration, the views represent the only information sources accessible to answer a query. A data warehouse can be seen as a set of materialised views, and, therefore, query processing reduces to view-based query answering. Finally, since the views provide partial knowledge on the database, view-based query processing can be seen as a special case query answering with incomplete information.

In this project, we are interested in the problem of *reasoning* with the views and the queries, rather than evaluating them. In particular, we consider the following *query management* tasks, based on the basic query consistency and containment reasoning tasks [Calvanese *et al.*, 1998b; Horrocks *et al.*, 1999a].

- **Query validation.** Incoherent queries - i.e., queries that can not return any value as answer, given their inconsistent meaning with respect to the schema - are detected before they are evaluated against the data.

- **Query organisation.** Data exploration may involve a great amount of queries, possibly submitted by different group of persons, in different periods of time, for different purposes. The system can organise the set of queries in a hierarchy, such that it is possible to retrieve already submitted similar or equivalent queries, together with the cached results. This is relevant if the queries need a substantial amount of time to be processed, or if the users associate comments or observations to the queries or to the answers.

- **Query generalisation.** In many situations, the query, even if it is consistent, can return an empty answer, since there is no actual document in the database satisfying it. In such cases, it is reasonable to generalise the query until a non empty answer is obtained; the complete query lattice is the obvious space where such generalisations can be searched for.

- **Query refinement.** Queries can be specified through an iterative refinement process supported by the complete description lattice for the queries. This process is useful for data exploration tasks. The user may specify his/her request using generic terms; after the query classification, which makes explicit the meaning and the specificity of the query itself and of the terms composing the query, the user may refine some terms of the query or introduce new terms, and iterate the process.

- **Intensional navigation.** Users may explore and discover new generic facts without querying the whole document base, but by giving an explicit meaning to the queries through classification. The system has the ability of answering a query with synthetic concepts representing the general characteristics of the data that satisfy it, as opposed to answering with long sequences of detailed data. Moreover, if the query is classified in a taxonomy of descriptions and queries already computed and indexing the answers, then it can be processed with respect to the indexed objects only, rather than with respect to the whole data base. For this reason, an intensional answer can be considered complementary to the standard one, allowing a preliminary phase where the user can refine its query and exactly mark the boundary of what she/he is looking for.

For the purpose of this project, we have identified a semantic characterisation of a view centred framework, where views (queries) are disjunction of conjunctive queries – i.e., non-recursive datalog queries – and the conceptual schema is the one mentioned in the previous Section. It is important to devise a methodology for query management, and to study the various tasks which are useful to support the methodology. In particular, we are concentrating ourselves on intensional navigation, which is probably the most advanced and interesting task. Intensional navigation can help a less skilled user during the initial step of query formulation, thus solving the critical aspect that only very skilled users have available sufficient knowledge about the schema of the stored information and, therefore, are able to formulate significant queries.

We have already started to design and implement an innovative tool for query management which exploits the knowledge given by a conceptual schema. The tool supports a user in formulating a query to a database with the help of the rich ontological information from the conceptual schema. The query makes use of a graphical representation, and the answer could be an intensional one. The tool uses in the background the basic query containment inference engine – based on DL technology – which is being currently developed within our Department [Horrocks *et al.*, 1999a].

References

[Abiteboul and Duschka, 1998] S. Abiteboul and O. Duschka. Complexity of answering queries using materialised views. In *Proc. of the 17th ACM SIGACT SIGMOD SIGART Symp. on Principles of Database Systems (PODS'98)*, pages 254–265, 1998.

[Artale and Franconi,] A. Artale and E. Franconi. On the expressiveness and complexity of temporal conceptual modelling. Submitted.

[Artale and Franconi, 1999] A. Artale and E. Franconi. Temporal ER modeling with description logics. In *Proc. of the International Conference on Conceptual Modeling (ER'99)*. Springer-Verlag, November 1999.

[Beeri *et al.*, 1997] C. Beeri, A. Y. Levy, and M.-C. Rousset. Rewriting queries using views in description logics. In *Proc. of the 16th ACM SIGACT SIGMOD SIGART Symp. on Principles of Database Systems (PODS'97)*, pages 99–108, 1997.

[Borgida, 1995] A. Borgida. Description logics in data management. *TKDE*, 7(5):671–682, 1995.

[Bresciani and Franconi, 1996] P. Bresciani and E. Franconi. Description logics for information access. In *Proceedings of the AI*IA 1996 Workshop on Access, Extraction and Integration of Knowledge*, Napoli, September 1996.

[Bresciani *et al.*, 2000] Paolo Bresciani, Michele Nori, and Nicola Pedot. A knowledge based paradigm for querying databases. In *Proc. of DEXA-00*, pages 794–804, 2000.

[Calvanese *et al.*, 1998a] D. Calvanese, G. De Giacomo, M. Lenzerini, D. Nardi, and R. Rosati. Description logic framework for information integration. In *Proceedings of the 6th International Conference on the Principles of Knowledge Representation and Reasoning (KR-98)*, pages 2–13. Morgan Kaufmann, 1998.

[Calvanese *et al.*, 1998b] D. Calvanese, Giuseppe De Giacomo, and M. Lenzerini. On the decidability of query containment under constraints. In *Proc. of the 17th ACM SIGACT SIGMOD SIGART Sym. on Principles of Database Systems (PODS'98)*, pages 149–158, 1998.

[Calvanese *et al.*, 1998c] D. Calvanese, M. Lenzerini, and D. Nardi. Description logics for conceptual data modeling. In J. Chomicki and G. Saake, editors, *Logics for Databases and Information Systems*. Kluwer, 1998.

[Calvanese *et al.*, 1999] D. Calvanese, M. Lenzerini, and D. Nardi. Unifying class-based representation formalisms. *J. of Artificial Intelligence Research*, 11:199–240, 1999.

[Calvanese *et al.*, 2000a] D. Calvanese, G. De Giacomo, and M. Lenzerini. Answering queries using views over description logics knowledge bases. In *Proc. of the 16th Nat. Conf. on Artificial Intelligence (AAAI 2000)*, 2000.

[Calvanese *et al.*, 2000b] D. Calvanese, G. De Giacomo, M. Lenzerini, and Moshe Y. Vardi. View-based query processing and constraint satisfaction. In *Proc. of the 15th IEEE Sym. on Logic in Computer Science (LICS 2000)*, 2000.

[Fensel *et al.*, 2000] D. Fensel, I. Horrocks, F. van Harmelen, S. Decker, M. Erdmann, and M. Klein. OIL in a nutshell. In *Proceedings of the European Knowledge Acquisition Conference (EKAW-2000)*, Lecture Notes In Artificial Intelligence. Springer-Verlag, 2000.

[Franconi and Kifer, 1999] E. Franconi and M. Kifer, editors. *Proceedings of the 6th International Workshop on Knowledge Representation meets Databases (KRDB'99)*. Linköping University Technical Report, July 1999. Also electronically available as CEUR Publication, Vol. 21, RWTH Aachen, Germany.

[Franconi and Ng, 2000] E. Franconi and G. Ng. The ICOM tool for intelligent conceptual modelling. In *Proc. of the 7^{th} International Workshop on Knowledge Representation meets Databases (KRDB'2000)*, 2000.

[Franconi and Sattler, 1999] E. Franconi and U. Sattler. A data warehouse conceptual data model for multidimensional aggregation. In *Proceedings of the Workshop on Design and Management of Data Warehouses (DMDW'99)*, 1999.

[Franconi *et al.*, 1999] E. Franconi, F. Baader, U. Sattler, and P. Vassiliadis. Multidimensional data models and aggregation. In M. Jarke, M. Lenzerini, Y. Vassilious, and P. Vassiliadis, editors, *Fundamentals of Data Warehousing*, chapter 5, pages 87–106. Springer-Verlag, 1999.

[Franconi *et al.*, 2000] E. Franconi, F. Grandi, and F. Mandreoli. A semantic approach for schema evolution and versioning in object-oriented databases. In *Proc. of the 6th International Conference on Rules and Objects in Databases (DOOD 2000)*, 2000.

[Franconi, 1997] E. Franconi. Software asset classification and retrieval with description logics. In *1997 International Workshop on Description Logics (DL'97)*, September 1997.

[Horrocks *et al.*, 1999a] I. Horrocks, U. Sattler, S. Tessaris, and S. Tobies. Query containment using a DLR ABox. LTCS-Report 99-15, LuFG Theoretical Computer Science, RWTH Aachen, Germany, 1999.

[Horrocks *et al.*, 1999b] Ian Horrocks, Ulrike Sattler, and Stephan Tobies. Practical reasoning for expressive description logics. In H. Ganzinger, D. McAllester, and A. Voronkov, editors, *Proceedings of the 6th International Conference on Logic for Programming and Automated Reasoning (LPAR'99)*, number 1705 in Lecture Notes in Artificial Intelligence, pages 161–180. Springer-Verlag, 1999.

[Horrocks, 1999] Ian Horrocks. FaCT and iFaCT. In *Proceedings of the International Workshop on Description Logics (DL'99)*, pages 133–135, 1999.

[Jarke *et al.*, 2000] M. Jarke, V. Quix, D. Calvanese, M. Lenzerini, E. Franconi, S. Ligoudistiano, P. Vassiliadis, and Yannis Vassiliou. Concept based design of data warehouses: The DWQ demonstrators. In *2000 ACM SIGMOD International Conference on Management of Data*, May 2000.

[Stock *et al.*, 1993] O. Stock, G. Carenini, F. Cecconi, E. Franconi, A. Lavelli, B. Magnini, F. Pianesi, M. Ponzi, V. Samek-Lodovici, and C. Strapparava. AlFresco: Enjoying the combination of natural language processing and hypermedia for information exploration. In Mark Maybury, editor, *Intelligent Multimedia Interfaces*, chapter 9. AAAI Press / The MIT Press, 1993. Also at IJCAI'91.

[Ullman, 1997] J. D. Ullman. Information integration using logical views. In *Proc. of the 6th Int. Conf on Database Theory (ICDT'97)*, pages 19–40, 1997.