Freie Universität Bozen
Libera Università di Bolzano
Free University of Bozen · Bolzano

Faculty of Computer Science
Free University of Bozen-Bolzano

RESEARCH AND STUDY PLAN

Lina Lubyte

XXII cycle

supervised by

Sergio Tessaris

December 2007

RESEARCH AND STUDY PLAN

Abstract

This document specifies the research area of the doctoral work, determines the questions to be examined and describes possible approaches for answering the latter. Particularly, the main focus of the current research is reuse of relational sources in the context of semantic-based access to information. This problem is tackled in two phases: *(i)* extracting semantics hidden in the relational sources by wrapping them by means of an ontology, *(ii)* studying the methodology for semantic enrichment of such ontologies.

1 Introduction and motivations

The use of a conceptual model (or an ontology) to describe relational data sources has been shown to be extremely useful to overcome many important data access problems. In particular, the notion of accessing information through the navigation of an ontology modeling the domain of an information system – which can be seen as a conceptual schema – only recently has gained interest in knowledge representation research. The ontology defines a vocabulary which is richer than the logical schema of the underlying data. Thus, the user would prefer to query the database using the rich vocabulary of the ontology. Moreover, this allows users (or applications) to formulate queries also in the case of schema mismatch, since the information system is accessed using the common and agreed-upon ontology vocabulary [15]. On the other hand, so-called *intentional navigation* can help a less skilled user during the initial step of query formulation, thus overcoming problems related with schema comprehension and so enabling him/her to easily formulate meaningful queries (e.g., Query Tool [11]).

To date, the task of wrapping relational data sources by means of an ontology is mainly done manually. However, large quantity of structured information sources available on the Web, as well as multiple databases in enterprises require automatic support for the conceptualization of the domain. Within this research area, we identify two key tasks that we discuss next.

Suppose the goal is to design an ontology that is to be used as a unifying view in enterprise integration. Typically, there will be a number of database systems that are used in the enterprise to be modelled. Instead of constructing the ontology from scratch, ontology designer wants an automatic support for converting the database schema into an initial ontology (i.e., support for ontology bootstrapping). Thus, it is desirable to derive ontology from the database schema together with a set of views that connect the ontology with the database, so that queries over the ontology can be answered by using the data in the database. Once a core ontology is obtained, ontology designer adds more details to a part of the ontology that has not yet been sufficiently described. An end user then uses the enriched ontology for formulating his/her queries over the enriched ontology and an application is expected to return the answer (obviously using sophisticated query answering techniques, such as query rewriting [10]). However, in some cases, the final result is not as expected (see next section for an example). To avoid unexpected results in such applications when using the enriched ontology, the ontology engineer must have control of the consequences of his/her modifications.

Motivated by the above scenario, we aim at understanding which is the best way of building such an infrastructure by providing a uniform access to the data sources in a coherent and user-oriented way. We see the problem in two phases: eliciting the semantics of data hidden in the structure of relational sources by wrapping them by means of an ontology; and understanding the methodology for semantic enrichment of such ontology, i.e., when those enrichments are meaningful and give sensible results.

The rest of the document is structured as follows. In the next Section we describe the problem by giving specific examples explaining the matter. Section 3 surveys the work relevant to our research. In Section 4 we describe the approach proposed and results achieved so far. Finally, we list additional activities of the past year in Section 5.

2 Problem description

In this section we define the problem of reusing relational sources in the context of semantic-based access to information. We do so by illustrating the matter with relevant examples. We first describe the problem of extracting ontologies from relational sources. Then, we introduce the specific scenario of semantic enrichment of such ontologies.

Let us consider a real-world database schema that implements CERIF[1] (Common European Research Information Format) 2000 Full Data Model – the standard EU recommendation as a tool to harmonize databases on research projects. The schema is strongly structured as it contains 123 tables. A major difficulty with such a database is to understand the meaning of the data. This lack of understanding hampers the effective utilization of data and also reduces the likelihood that maintenance activities can be performed correctly. Such an understanding can only be achieved by representing the semantics hidden in the relational source with its conceptual view, i.e., an ontology. However, the task of wrapping relational data sources by means of an ontology is mainly done manually. Moreover, with schemas like CERIF (having more than 100 tables), the manual process is time-consuming and error-prone. Therefore, as a first part of the doctoral work, we aim at defining a framework for automatically extracting from a relational database an ontology that is to be used as a conceptual view over the data. The semantic mapping between the database schema and its conceptualisation is captured by associating views over the data source to elements of the extracted ontology. In such a setting, queries over the ontology can be answered by using the data in the database.

Having the ontology derived from database schema, a pertinent task is to enrich it by adding more details to a part of the ontology that has not yet been sufficiently described. When such an operation is performed, it is important for ontology engineer to be aware of the consequences of such modifications. In particular, when the enriched ontology is used for querying the information residing in the database, it is plausible that unexpected answers are returned.

We illustrate the problem with an example shown in Figure 1. Please note that in this example the derived ontology is represented with DL *DL-Lite* [9]. We use Entity-Relationship (ER) notation for a clear pictorial representation, and we do not discuss in this document how common ER constructs can be expressed with *DL-Lite* (see [9] for details). At the lowest level of this figure we have an input schema of the database, providing the actual data, whereas

[1] http://cordis.europa.eu/cerif/src/toolkit.htm

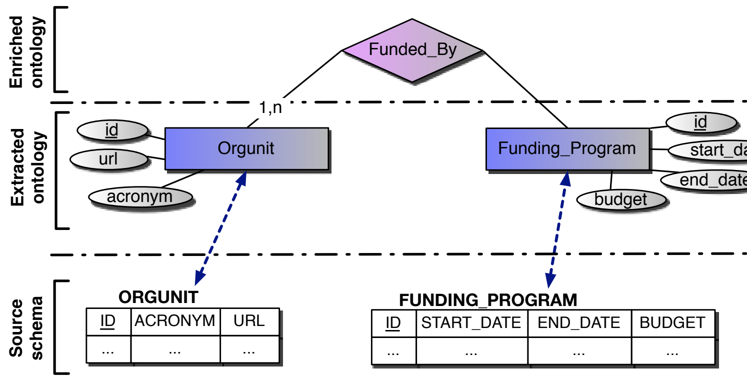

Figure 1: An example scenario

the middle part represents the ontology automatically generated from the input database schema.

Assume we have a database schema with two relations ORGUNIT and FUNDING_PROGRAM (taken from CERIF schema). Since no foreign keys are present in none of the two relations, with the ontology extraction procedure we obtain two corresponding entities, *Orgunit* and *Funding_Program*, not related to each other via any relationship. The associated mappings (views over the data source) are depicted with dashed arrows. When the ontology engineer sees the derived ontology, he/she decides that a relationship, say *Funded_By*, between the two entities exists. Now, suppose a user, looking at the enriched ontology (the lowest part is hidden from the user), wants to know the pairs of organisational units and funding programmes by which those organisational units are funded, i.e., he/she poses a query

$$q(x, y) \leftarrow \mathit{Funded\_By}(x, y).$$

It is immediate to see that the evaluation of the above query gives an empty answer, since during query reformulation step none of the inclusion assertions will be applicable to the atom $\mathit{Funded\_By}(x, y)$. However, consider the situation when ontology engineer adds mandatory participation constraint for the entity *Orgunit*, i.e., he/she adds the inclusion assertion $\mathit{Orgunit} \sqsubseteq \exists \mathit{Funded\_By}$. Then, given a query

$$q(x) \leftarrow \mathit{Funded\_By}(x, \_),$$

a new query is obtained

$$q'(x) \leftarrow \mathit{Orgunit}(x),$$

thus returning the set of all organisational units.

Following observations above, as a second part of this work, we aim at studying such a phenomenon. In other words, the goal is to characterize the methodology for semantic enrichment of ontologies, guaranteeing meaningful and sensible results. Currently there are no general guidelines of how to avoid or solve the latter situations. As a consequence, those are dealt with manually to date. Therefore, we are concerned with giving guidelines for knowledge engineers such that this problem could be minimized.

3 Literature overview

In this section we review work that is related to our research on reusing relational sources in the context of semantic based access to information. We start by surveying the work related to the problem of ontology extraction, and in particular we emphasize on the field of database reverse engineering (Section 3.1). Since the problem of eliciting data semantics from relational sources has been carried out from the beginning of the doctoral work, we also give insights of how our work differs from the state of the art approaches.

The study of the second problem – that of semantic enrichment of ontologies – is instead at a rather early stage. In general, the problem of accessing information through enriched ontologies has not been tackled in the literature. However, as a first attempt to approach the problem, we rely on *query reformulation* techniques, which we describe in Section 3.2. Finally, we survey in Section 3.3 the results on query and view languages that *admit view-based rewritings* (i.e., has Beth definability property), which we might consider as an alternative approach to the problem.

3.1 Database reverse engineering

The first focus of the doctoral work is that of deriving from a relational database schema its corresponding conceptual view. In this process we integrate and enhance many aspects of *database reverse engineering* (DBRE) [16], which we survey next.

There are various reasons for applying DBRE: maintenance and redesign (it is much more accurate to create a new schema by modifying an old one rather than starting from scratch), data migration (with the success of object-orientation, an interest in reverse engineering of relational database into object-oriented specifications has been risen), integration of databases (integration process is often performed at the conceptual level). As a consequence, DBRE is defined as a process of recovering a conceptual model that represents the meaning of the logical schema by examining an existing database system to identify the database contents and their interrelationships. Approaches to recovering a conceptual schema from a relational database have appeared in the literature over the years [27, 13, 3, 1, 5]. Four main sources have been explored for finding evidence to construct a conceptual schema from a logical database: the structures and integrity constraints of the database schema, the application programs that access the database, the data instances stored in the database, and the users and designers. Moreover, because reverse engineering of relational databases is a complex task, all existing approaches are conditioned by a set of restrictive assumptions, namely, relational schemas are supposed to be highly normalized (3NF, BCNF), semantic information about the schema is available (e.g., keys and foreign keys, inclusion and exclusion dependencies), etc. In particular, [27] use schema structures and constraints by considering an input relational database being in BCNF. In Andersson's work [3] information about functional dependencies, keys and inclusion dependencies are deduced by looking into data manipulation statements that can be extracted from the application code. The approaches in [13] analyze not only the database schema, but also data instances, while the work in [1] relies only on data instances in order to identify candidate keys, to locate foreign keys and to decide on the appropriate links between the given relations. Finally, the method of [19] decides the correct object types in the conceptual schemas by interacting with users. Even though there are similarities between this area and the problem we are tackling, there are however important differences. In particular, DBRE approaches firstly produce just a pictorial representation of a conceptual model, without showing how it links to the database schema, and are thus used for "documenting" the database. Our approach, instead, is tailored for the direct use of the extracted ontology – that of accessing information mediated by the ontology, where the views generated during the extraction process that connect the derived ontology with the data sources play a crucial role. Secondly, most of DBRE methods are informal and do not specify the quality of the results. On the contrary, we provide formal results showing that the extracted ontology represents all information sources and does not represent any extra information not present in the sources.

Furthermore, the recent call for a Semantic Web arose several approaches in bringing together relational databases and ontologies, particularly tailored to migrating data-intensive Web pages into the ontology-based Semantic Web. Among them we mention [5], where the author analyzes key, data and attribute correlations (as well as their combination) to first transform the 3NF database schema into an ontology and then migrate the data. Despite that the techniques of [5] used for transforming the schema into an ontology are close to our extraction procedure, the scenarios in which the latter and our approach are used differ and result in the distinctions already described above. The work by Y. An et al. [2] proposes an algorithm for automatic mapping between relations and ontologies, when given as input simple correspondences from attributes of relations to datatype properties of classes in an ontology. Unlike our approach, it requires a target existing ontology onto which the relations are mapped to. Finally, the approach of [21] extracts the schema information of the data source and converts it into an ontology. However, this technique extracts only the structural information about the ontology, so the constraints are not taken into account.

3.2 Query reformulation

Query reformulation is first of all at the heart of query processing in data integration [22]: the query over the global schema has to be reformulated in terms of a set of queries over the sources. Note that in our framework, the views generated by the ontology extraction algorithm correspond to a set of global-as-view (GAV) mappings[2]. It is easy to see that in GAV data integration systems, when integrity constraints are not present in the global schema, query processing can be based on a simple *unfolding* strategy. Specifically, when we have a query q over the alphabet $\mathcal{A}_{\mathcal{G}}$ of the global schema, every element of $\mathcal{A}_{\mathcal{G}}$ is substituted with the corresponding query over the sources, and the resulting query is then evaluated at the sources. However, when the language used for expressing the global schema allows for integrity constraints, query processing in GAV systems becomes more complex. Indeed, in this case, integrity constraints can be used to overcome *incompleteness* of data at the sources [22]. A fundamental tool for dealing with constraints at the global schema is the use of *chase* of a conjunctive query [26, 20]. Intuitively, given a conjunctive query, its conjuncts are "frozen" and seen as facts in a database, where each variable is associated to a distinct value. Since this collection of facts in general does not satisfy the integrity constraints, the idea is to convert the initial facts into a new set of facts constituting a database that satisfies the dependencies, possibly by collapsing facts or adding new facts. The work in [8] adopts this notion and presents an algorithm, based on encoding the information about the conceptual schema into a rewriting of the query, that computes the consistent answers to queries posed on EER schemata, where the data are incomplete w.r.t. the schema.

Query reformulation method for *DL-Lite* family of DLs [10] follows the same idea of using ontology constraints (TBox) as means to overcome incompleteness at the data sources (ABox) in a different way: given a conjunctive query q over the knowledge base (KB), the assertions of the TBox are compiled into the query itself, thus obtaining a new query q'. The union of such conjunctive queries are then evaluated over the ABox of KB, as if the ABox were a simple relational database. Since the size of each q' does not depend on the ABox, the data complexity of the whole query answering algorithm is polynomial [9].

3.3 Determinacy and rewriting of queries using views

The question of whether a given set of queries on a database (i.e., views) can be used to answer another query arises in many different contexts (for a survey see [17]). This can be formulated at several levels. The first attempt might be syntactic or language specific: query Q can be answered by the views $\mathcal{V}$ if there exists a query Q' in language L in which the $\mathcal{V}$ predicates are the only primitive symbols and Q and Q' are logically equivalent assuming the view definitions. In this case, it is said that there exists an L-rewriting of Q in terms of $\mathcal{V}$. The second formulation might be semantic or information theoretic: query Q can be answered by the views $\mathcal{V}$ if $\mathcal{V}(D_1) = \mathcal{V}(D_2)$ guarantees that $Q(D_1) = Q(D_2)$, for all database instances D_1 and D_2 [31, 28]. In this case, it is said that Q is *determined* by the views $\mathcal{V}$, which intuitively means that $\mathcal{V}$ provides enough information to uniquely determine the answer to Q. If there exists an L-rewriting of Q in terms of $\mathcal{V}$, then Q is also determined by $\mathcal{V}$. The

[2]For this reason we do not review here query processing in LAV data integration systems.

other direction may not hold and clearly depends on the choice of the rewriting language. Thus, in this research program, the main question is: given a query Q and views $\mathcal{V}$ in L for which it is known that Q is determined by $\mathcal{V}$; what is the smallest query language extending L in which Q can be rewritten in terms of $\mathcal{V}$; in particular, is L itself sufficient? If so, it is said that L *admits view-based rewritings*, which is, of course, the desired situation: the rewriting is not more difficult than the views and the queries [28]. However, conjunctive queries do not admit view based rewritings, as known from [30]. For first order (FO) queries and view definitions over possibly infinite databases, the answer is yes, as follows from old results of Beth (definability theorem) [7] (different formulations of Beth's property are surveyd in [18]) and Craig (interpolation theorem) [14, 12]. For finite databases, FO is again not sufficient [31]. But M. Marx in [28] has shown that the *packed fragment* [4] of FO logic admits view-based rewritings, so do its restrictions to (unions of) conjunctive queries. Moreover, it is decidable whether a query is determined by a set of views for queries and views expressed in the packed fragment.

The relevance of determinacy and rewriting to our work is driven with two facts. First, when added ontology terms are determined by and can be rewritten to the database terms, the answers to queries over the enriched ontology are guaranteed to be non-empty. Second, the property of a (fragment of a) language to admit view-based rewritings is crucial in devising the rewriting of queries over the enriched ontology terms into the queries over the database terms.

4 What has been done so far

In this section we provide a brief description of the proposed approach and results achieved so far. Please note that the problem of extracting ontologies from relational data sources has been carried out from the beginning of the doctoral work, thus, in this section we concentrate on the former. The problem of accessing the data sources using the enriched ontology is at the initial stage, and we only provide initial insights for approaching the matter.

Ontology extraction. First, an automatic procedure for extracting an ontology with mappings (i.e., set of views) from a relational database schema was developed, that was first described in [24], and in more detail in [25]. The core extraction algorithm uses database reverse engineering techniques (see Section 3.1). Specifically, the heuristics underlying the ontology extraction process are based on ideas of standard relational schema design and normalisation. In fact, we assume that the relational source is in third normal form. To represent the extracted ontology, we use an ontology language, rather that a graphical notation, in order to provide a precise formal semantics. In particular, we adopt a variant of *DLR-Lite* [10] description logic because of its nice computational properties, and ability to express the mostly used modelling constraints.

Second, the developed technique was evaluated using the relative *information capacities* of the source and target schemata. Our developed ontology extraction

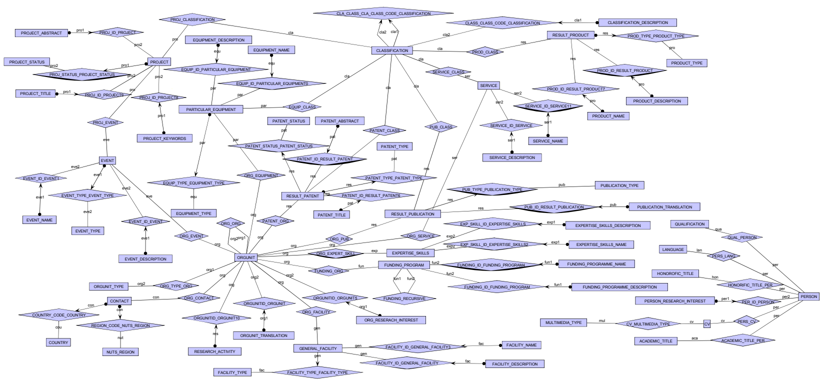

Figure 2: An ontology extracted from CERIF database schema

technique can be seen as a *schema transformation*, as defined in [29]. It has been shown that the ontology extraction procedure is an *equivalence preserving schema transformation* (the actual proof can be found in [25]). This result shows that queries expressed over the extracted ontology can be evaluated by simply expanding the generated views. However, this is no longer true in the case when the ontology is going to be modified. Then, more sophisticated query answering techniques must be adopted in order to guarantee completeness (e.g., query rewriting [10]). Moreover, the ontology modification phenomenon, as discussed in Section 2 must be addressed.

Third, the prototype ontology extraction tool[3] was developed and tested with a real-world database schema CERIF. Figure 2 shows the ontology extracted from CERIF database schema.

Finally, the following contributions for TONES project[4] were submitted: section "Ontology Extraction from DB Schemas" to D13 deliverable [6], a prototype ontology extraction tool to D15 deliverable ("Software Tools for Ontology Design and Maintenance") and a prototype for TONES demonstration session.

Ontology enrichment. As a starting point to approach the problem, we have chosen to rely on query rewriting techniques, and in particular on query reformulation algorithm for *DL-Lite* family of DLs [10]. We have devised an algorithm that, given a term and a TBox of an (already enriched) ontology, return if this term gives an empty answer when evaluating it over the data sources. Note that it is enough to check whether a given term can be rewritten by applying to it an appropriate inclusion dependency and then to verify that a view for the rewritten term exists.

As an example, consider the conceptual schema displayed in Figure 1 of

[3]The demo is available at https://lamj.inf.unibz.it/ontoext/OntoSynt.html.
[4]http://www.tonesproject.org/

Section 2. The *DL-Lite* TBox for this schema is the following:

(1) $\exists Funded\_By \sqsubseteq Orgunit$

(2) $\exists Funded\_By^- \sqsubseteq Funding\_Program$

(3) $Orgunit \sqsubseteq \exists Funded\_By$

Now, consider a query

$$q(x) \leftarrow Funded\_By(x, \_),$$

asking for organisational units that are funded by some funding programme. There is no mapping associated to the term *Funded_By*. However, by applying query reformulation algorithm for *DL-Lite* (the actual algorithm can be found in [10]), $Funded\_By(x, \_)$ atom is applicable to the inclusion assertion (3), and new query is obtained

$$q'(x) \leftarrow Orgunit(x).$$

Orgunit term is linked to ORGUNIT table and thus the set of all organisational units is returned.

In parallel, we are also studying Beth's definability property for queries and views (inspired by [28]), but we do not have any concrete results to date.

5 Other activities of the past year

Published papers

- Poster paper at DL 2007 [24];
- Paper at Ph.D. workshop in CIKM, PIKM 2007 [23].

Attended conferences and workshops

- DL 2007: 20th Description Logics Workshop, Bressanone, June, 2007
- VLDB 2007: 33rd International Conference on Very Large Databases, Vienna, September, 2007
- SWDB-ODBIS07: Joint ODBIS & SWDB Workshop on Semantic Web, Ontologies, Databases (colocated with VLDB 2007), Vienna, September, 2007
- CIKM 2007: 16th International Conference on Information and Knowledge Management, Lisbon, November, 2007
- PIKM 2007: 1st Ph.D. Workshop in CIKM, Lisbon, November, 2007

Attended summer schools

- ESSLLI 2007: 19th European Summer School on Logic, Language and Information, Dublin, Ireland, August 6-17, 2007
- ACAI 2007: Logic for Artificial Intelligence, Leuven, Belgium, August 20-28, 2007
- EDBT 2007: 8th EDBT Summer School on Database Technologies for Novel Applications, Bozen-Bolzano, September 3-7, 2007

Presented talks

- Talk at KRDB PhD Students workshop, Bozen-Bolzano, May, 2007
- Invited Seminar at Bruno Kessler Foundation - IRST, Trento, July, 2007
- Paper presentation at PIKM 2007, Lisbon, November, 2007

References

[1] R. Alhajj. Extracting an extended entity-relationship model from a legacy relational database. *Information Systems*, 26(6):597–618, 2003.

[2] Y. An, A. Borgida, and J. Mylopoulos. Inferring complex semantic mappings between relational tables and ontologies from simple correspondences. In *ODBASE'05*, pages 1152–1169, 2005.

[3] M. Andersson. Extracting an entity-relationship schema from a relational database through reverse engineering. In *Proc. of Int. Conf. on Entity-Relationship Approach (ER'94)*, pages 403–419, 1994.

[4] H. Andreka, J. van Benthem, and I. Nemeti. Modal languages and bounded fragments of predicate logic. *Journal of Philosophical Logic*, 27(3):217–274, 1998.

[5] Irina Astrova. Reverse engineering of relational databases to ontologies. In *ESWS*, pages 327–341, 2004.

[6] F. Baader, R. Bernardi, D. Calvanese, A. Cali, B. Cuenca Grau, M. Garcia, G. de Giacomo, A. Kaplunova, O. Kutz, D. Lembo, M. Lenzerini, L. Lubyte, C. Lutz, M. Milicic, R. Moller, B. Parsia, R. Rosati, U. Sattler, B. Sertkaya, S. Tessaris, C. Thorne, and A.-Y. Turhan. Techniques for ontology design and maintenance. Deliverable D13, TONES EU-IST STREP FP6-7603, January 2007.

[7] E. Beth. On padoa's method in the theory of definition. *Indagationes Mathematicae*, 15:330–339, 1953.

[8] A. Cali. Querying imcomplete data with logic porgrams: Er strikes back. In *ER*, pages 245–260, 2007.

[9] D. Calvanese, G. De Giacomo, D. Lembo, M. Lenzerini, and R. Rosati. Dl-lite: Tractable description logics for ontologies. In *Proc. of the 20th Nat. Conf. on Artificial Intelligence (AAAI 2005)*, pages 602–607, 2005.

[10] D. Calvanese, G. De Giacomo, D. Lembo, M. Lenzerini, and R. Rosati. Data complexity of query answering in description logics. In *Proc. of KR 2006*, pages 260–270, 2006.

[11] Tiziana Catarci, Paolo Dongilli, Tania Di Mascio, Enrico Franconi, Giuseppe Santucci, and Sergio Tessaris. An ontology based visual tool for query formulation support. In *Proc. of the 16th Eur. Conf. on Artificial Intelligence (ECAI 2004)*, 2004.

[12] C. Chang and H. Keisler. *Model Theory.* North-Holland, 1973.

[13] R. H. L. Chiang, T. M. Barron, and V. C. Storey. Reverse engineering of relational databases: extraction of an eer model from a relational database. *Data and Knowledge Engineering*, 12(2):107–142, 1994.

[14] W. Craig. Three uses of the ferbrand-gentzen theorem in relation to model theory and proof theory. *Journal of Symbolic Logic*, 22:269–285, 1957.

[15] E. Franconi, S. Tessaris, B. C. Grau, B. Suntisrivaraporn, C. Lutz, R. Moller, and D. Lembo. Revised ontology task handbook. Deliverable D07, TONES EU-IST STREP FP6-7603, August 2006.

[16] J. L. Hainaut. Database reverse engineering: models, techniques and strategies. In *Proc. of the 10th Conference on ER Approach*, 1998.

[17] A. Y. Halevy. Answering queries using views: A survey. *VLDB Journal*, 10(4):270–294, 2001.

[18] E. Hoogland. Algebraic characterizations of various beth definability properties. *Studia Logica*, 65(1):91–112, 2000.

[19] P. Johannesson. A method for transforming relational schemas into conceptual schemas. In *Proc. of the Int. Conf. on Data Engineering (ICDE'94)*, pages 190–201, 1994.

[20] D. S. Johnson and A. C. Klug. Testing containment of conjunctive queries under functional and inclusion dependencies. *J. of Computer and System Sciences*, 28(1):167–189, 1984.

[21] C. P. Laborda and S. Conrad. Bringing relational data into the semantic web using sparql and relational.owl. In *Proc. of the 22nd Int. Conf. on Data Engineering Workshops (ICDEW'06)*, pages 55–62, 2006.

[22] M. Lenzerini. Data integration: A theoretical perspective. In *Proc. of PODS02*, pages 233–346, 2002.

[23] L. Lubyte. Reusing relational sources for semantic information access. In A. Varde and J. Pei, editors, *Proc. of the First Ph.D. Workshop in CIKM, PIKM 2007, Sixteenth ACM Conference on Information and Knowledge Management, CIKM 2007*, pages 9–16, 2007.

[24] L. Lubyte and S. Tessaris. Extracting ontologies from relational databases. In *Proc. of the 20th Int. Workshop on Description Logics (DL'07)*, 2007.

[25] L. Lubyte and S. Tessaris. Extracting ontologies from relational databases. Technical report, KRDB group – Free University of Bozen-Bolzano, 2007. Available at http://www.inf.unibz.it/krdb/pub/TR/KRDB07-4.pdf.

[26] D. Maier, A. O. Mendelzon, and Y. Sagiv. Testing implications of data dependencies. *ACM Transactions on Database Systems (TODS)*, 4:455–469, 1979.

[27] V. M. Markowitz and J. A. Makowsky. Identifying extended entity-relationship object structures in relational schemas. *IEEE Transactions on Software Engineering*, 16(8):777–790, 1990.

[28] M. Marx. Queries determined by views: Pack your views. In *PODS*, 2007.

[29] R. J. Miller, Yannis E. Ioannidis, and Raghu Ramakrishnan. The use of information capacity in schema integration and translation. In *Proc. of VLDB'93*, pages 120–133, San Francisco, CA, USA, 1993. Morgan Kaufmann Publishers Inc.

[30] A. Nash, L. Segoufin, and V. Vianu. Determinacy and rewriting of conjunctive queries using views: a progress report. In *Proc. of ICDT*, 2007.

[31] L. Segoufin and V. Vianu. Views and queries: Determinacy and rewriting. In *PODS*, pages 49–60, 2005.