
We recommend you cite the published version.
The publisher’s URL is: https://doi.org/10.1016/j.aci.2017.07.003

Refereed: Yes

(no note)

Disclaimer

UWE has obtained warranties from all depositors as to their title in the material deposited and as to their right to deposit such material.

UWE makes no representation or warranties of commercial utility, title, or fitness for a particular purpose or any other warranty, express or implied in respect of any material deposited.

UWE makes no representation that the use of the materials will not infringe any patent, copyright, trademark or other property or proprietary rights.

UWE accepts no liability for any infringement of intellectual property rights in any material deposited but will remove such material from public view pending investigation in the event of an allegation of any such infringement.

PLEASE SCROLL DOWN FOR TEXT.
The use of Ontologies for Effective Knowledge Modelling and Information Retrieval

Kamran Munir, M. Sheraz Anjum

PII: S2210-8327(17)30064-9
DOI: http://dx.doi.org/10.1016/j.aci.2017.07.003
Reference: ACI 77

To appear in: Applied Computing and Informatics

Please cite this article as: Munir, K., Sheraz Anjum, M., The use of Ontologies for Effective Knowledge Modelling and Information Retrieval, Applied Computing and Informatics (2017), doi: http://dx.doi.org/10.1016/j.aci.2017.07.003

This is a PDF file of an unedited manuscript that has been accepted for publication. As a service to our customers we are providing this early version of the manuscript. The manuscript will undergo copyediting, typesetting, and review of the resulting proof before it is published in its final form. Please note that during the production process errors may be discovered which could affect the content, and all legal disclaimers that apply to the journal pertain.
The use of Ontologies for Effective Knowledge Modelling and Information Retrieval

Kamran Munir*
Department of Computer Science and Creative Technologies, University of the West of England, BS16 1QY, Bristol, United Kingdom.

M. Sheraz Anjum
School of Electrical Engineering and Computer Science, National University of Sciences and Technology (NUST), H-12, Islamabad.

Abstract
The dramatic increase in the use of knowledge discovery applications requires end users to write complex database search requests to retrieve information. Such users are not only expected to grasp the structural complexity of complex databases but also the semantic relationships between data stored in databases. In order to overcome such difficulties, researchers have been focusing on knowledge representation and interactive query generation through ontologies, with particular emphasis on improving the interface between data and search requests in order to bring the result sets closer to users research requirements. This paper discusses ontology-based information retrieval approaches and techniques by taking into consideration the aspects of ontology modelling, processing and the translation of ontological knowledge into database search requests. It also extensively compares the existing ontology-to-database transformation and mapping approaches in terms of loss of data and semantics, structural mapping and domain knowledge applicability. The research outcomes, recommendations and future challenges presented in this paper can bridge the gap between ontology and relational models to generate precise search requests using ontologies. Moreover, the comparison presented between various ontology-based information retrieval, database-to-ontology transformations and ontology-to-database mappings approaches provides a reference for enhancing the searching capabilities of massively loaded information management systems.

Keywords: information systems, ontology, domain knowledge, database, information retrieval, knowledge management

1. Introduction
In information management systems, structured query formulation languages are one means of retrieving information. Writing structured queries is a powerful method to access data since it allows end-users to formulate complex database queries by learning specialised query languages. However, query formulation with the exception of a few visual query generation and refinement approaches remains appreciatively difficult for the various levels of systems users. In recent years information retrieval has turned out to be more complicated with the increased use of data mining, decision support and business analytics applications. Consequently, researchers focus has been on approaches that include visual database interfaces [1] and interactive query generation through graphs [2] and [3], with a particular emphasis on providing interactive natural language interfaces to support query generation. Recently, semantic-based approaches using domain ontologies have been adapted for data modelling and information retrieval. Ontology-based information retrieval, for example as in [4], [5] and [6] mainly aim at improving the interface between data and search requests in order to bring the result sets closer to the users research requirements. In general, an ontology represents a shared, agreed and detailed model (or set of concepts) of a certain problem domain [7]. One major advantage of using a domain ontology is its ability to define a semantic model of the data combined with the associated domain knowledge. Ontologies can also be used to define links between different types of semantic knowledge. Thus, ontologies can be used in formulating some data searching strategies.

This paper discusses ontology-based information retrieval approaches by taking into consideration the aspects of:
(a) ontology generation from database schema(s);
(b) processing of domain knowledge to represent it as ontological knowledge; and
(c) the translation of such ontological knowledge into relational database queries.
Moreover, it provides a comparison between ontology-to-database transformation and mapping approaches in terms of:
loss of data and semantics;
structural mapping; and
domain knowledge applicability.

The outcomes presented in this paper can be beneficial in bridging the gap between ontology and relational models while attempting to generate precise search requests from ontology expressions. Moreover, the comparison presented between various ontology-based information retrieval, database-to-ontology transformations and ontology-to-database mappings tools/approaches provides a reference for enhancing the
searching capabilities of massively loaded information and management systems [8].

After having introduced the motivation and context, the remainder of this paper is divided into the following sections. Section 2 introduces ontologies and domain knowledge representations. Section 3 reviews the state of the art in ontology-based database information retrieval. Section 4 discusses our findings in relation to ontology-based information retrieval. Section 5 reviews the state of the art database schema to ontology schema transformation and ontology-to-database mapping approaches in terms of loss of data and semantics, structural mapping and domain knowledge applicability. Section 6 provides a discussion and highlights a possibility of combining database-to-ontology transformation and ontology-to-database mappings approaches for relational query formulation. Finally, Section 7 outlines the future challenges and possible research directions towards using ontologies for information retrieval from information and Big data management systems.

2. Ontologies and Knowledge Representation

Over the past few years, many ontology development and query languages have been developed and this is still a continuing effort. While building an ontology-based system, it first requires deciding which ontology language is to be used in a given context. Numerous ontology languages were developed in the last few years. Most of these are based on the eXtensible Markup Language (XML) [9] which enables them to be machine interpretable [10]. Notable examples are the Resource Description Framework (RDF) and RDF Schema [11], the DARPA Agent Markup Language and the Ontology Inference Layer (DAML+OIL) [12], and the Ontology Web language (OWL) [13] and OWL2 [14]. In order to use ontologies for query formulation it is important to evaluate them in terms of their expressive power, tools and reasoning support in order to decide which ontology language is best suitable for this task. Most developments in the latest ontology languages are influenced by the RDF/RDFS XML based language rules and ranking structure. OWL has three sublanguages: OWL-Lite, OWL-DL and OWL-Full: it was developed on top of RDF and DAML+OIL [15] providing gradually more expressiveness power. The Semantic Web Rule Language (SWRL) [16] adds rules to OWL-DL (to increase its expressiveness) and is a combination of the OWL-DL and sublanguages of the Rule Markup Language such as First Order Language. The basic idea of SWRL is to extend OWL-DL. It is simple and has tight integration with the existing OWL language; this can be considered as a key characteristic of SWRL. Recently, OWL2 [14] has been developed on the existing structure to OWL (OWL1) i.e. all the building blocks of OWL2 are present in OWL1; therefore all OWL1 ontologies remain valid OWL2 ontologies [14]. There are three new profiles of OWL 2: (1) OWL 2-EL, (2) OWL 2-QL, and (3) OWL 2-RL. These profiles are syntactic subsets of OWL 2 constructs. Selection between these profiles depends on the reasoning tasks and ontology structure. A comparison between RDF(s), OWL-1 and OWL-2 showing the possible uses of knowledge representation concepts to formulate ontology based relational database queries is presented in Table 1. In summary, both OWL and RDF have many common features, but OWL is a stronger language with greater machine interpretability than RDF. Moreover, OWL comes with a larger vocabulary and a stronger syntax than RDF, which can be used to define complex ontology concept restrictions and subsequently to formulate ontology based relational database queries.

3. Ontology-based Information Retrieval

This section reviews the state of the art in ontology-based database information retrieval. Here, a historical overview of information retrieval approaches is first presented, followed by a detailed analysis of existing ontology-based query systems and data search strategies in relation to three different key aspects that guided the review of such work. These three aspects are: (1) ontology assisted visual or interactive query formulation; (2) ontology based information linking approaches (also known as keyword search); and (3) ontology based query refinement (including query enrichment).

3.1. Information Retrieval from a Historical Perspective

Database information retrieval is the search for information in databases. The need for effective methods to automate information retrieval has grown in importance because of the significant increase in the amount of both structured and unstructured information embodied in information sources. Over the years, many visual information retrieval approaches came into existence which aim to reduce the end users effort while interacting with databases. These approaches intend to extract information from databases using visual tools. Such approaches include form-based [17], query by example (QBE) [18] or query by template (QBT) [19] etc. These approaches work for basic relational database queries, primarily because tabular structure of the database fits well with the tabular skeletons used in query interfaces. However, such approaches...
do not help in semantic data retrieval nor do they provide any query formulation support to generate complex queries.

To overcome the shortcomings identified above, further implementation improvements were advocated. One example is QUICK (Universal Interface with Conceptual Knowledge) [18] that focuses on automating query formulation by exploiting ER conceptual schema design knowledge. Unfortunately, in the real world the ER model has been used primarily for database design and they often do not store domain knowledge. Therefore, the ER based query formulation approaches cannot provide a reliable method to depend on its comprehensiveness in expressing low-level query constraints. More recently, several ontology languages with properly specified semantics have been developed. Several ontology-based approaches have been reported in the literature that can provide intelligent query formulation services for relational databases. Such approaches are reviewed in the following sections.

3.2. Ontology-based Query Formulation Approaches

Ontology-based Visual or Interactive query formulation systems are query systems for databases that use visual representations to express related data requests. These systems adapt ontologies for database query generation in order to improve the effectiveness of the human-computer communication. In recent years, many such systems have been reported in the literature (e.g., TAMBIS [20], GROLL [21], SEWASIE [22], Ontogator [23], OntoViews [24], OntoQF [25], VISAGE [26], Smarth [27], Semantic-based [28] and many others). In most of these ontology based visual query formulation systems, the search queries are performed using an ontology browser that visualises the ontology as a tree. The actual search is done via concept selection through a visual tree or through keywords annotated by the visual ontology concepts.

The TAMBIS system [20] supports the specialisation or generalisation of the base or filler ontology concepts to build database specific queries interactively. Here the data in the databases are stored (linked) as instances of ontology concepts. This approach can be applied to resolve integration problems, where all information sources have the same schema or provide nearly the same view of a domain. Another similar approach based on ontological graph pattern queries is presented in GRQL [21] and KnowledgeSifter [29]. GRQL relies on the full power of the RDF/S data model and provides a GUI for building queries based on ontology navigation. In this approach, queries are constructed by graphically navigating through individual RDF/S classes and property definitions. In SEWASIE (Semantic Webs and AgentS in Integrated Economies) [22], the principles of designing and developing an ontology-based query interface are presented. The query interface of SEWASIE supports the user in formulating a query through an iterative refinement process supported by ontology navigation where in the query formulation process, a user can specify a request using generic terms, can refine some terms of a query or can introduce new terms, and can iterate the process if needed.

In OntoQF [25] OWL-DL ontologies have been used for information retrieval by automatically generating relational database queries using pre-stored domain knowledge. In comparison to other existing approaches, one of the main features of OntoQF approach is that it uses a combination of both database-to-ontology transformation and mappings to enable the automatic query formulation process, which helps in generating precise database queries. Overall, OntoQF uses a two-phase approach. In the first pre-processing phase, domain ontology is generated from relational schema and related mappings are defined which links the domain ontology concepts to relational entities/columns and vice versa [30]. Moreover, the domain experts can specify statements as ontology statements using a visual ontology query editor. OntoQF rules for ontology-based relational query formulation suggest that for such query formulation the generated domain ontology does not require the definition of datatype ranges [31] or specific constraints that are expressed in the database schema. Moreover, the domain knowledge is to be expressed in terms of OWL-DL assertions as concept restrictions, which need to be consistent with the respective domain ontology schema. In the second translation phase, the OntoQF engine translates ontology statements into the corresponding relational query statements. OntoQF’s approach is suitable for those systems or data mining applications that aim to keep all data at the original location(s) and use domain ontology for knowledge-based information retrieval [25].

The system presented in [32] provides interactive database query generation through non-directed graphs supporting natural languages. The ontology language used in this system is based on the RDF structure. In order to construct queries, query terms are suggested to a user in a natural language from a predefined vocabulary. In a report of the EU Translational Research and Patient Safety in Europe (TRANSFoRm) [33], a query and data extraction workbench has been presented. The TRANSFoRm query formulation workbench software tool provides interfaces to author, store and deploy queries of clinical data in order to identify subjects for clinical studies. Moreover, TRANSFoRm query formulation workbench enables users to define criteria groups flexibly, with most catering for complex queries with combinations of operators.

The search method of ontology-based image retrieval systems such as Ontogator [23], and OntoViews [24] are examples of a concept-based multi-facet search using RDFS ontologies. In a multi-facet search, multiple distinct views are augmented to data created via ontology projection[24]. OntoViews, supports semantic auto-completion of a query [24]. It uses a keyword search mechanism for ontology navigation. The search keywords are linked directly to ontology classes. A user search request is processed as a multi-facet search and results are delivered in a web browser. Once a single interesting instance (at least) has been found, additional information can be retrieved via ontology browsing.

Effective information retrieval is becoming more challenging with the increase in the use of Multimedia databases, which are usually bigger than traditional databases. In [34] a semantic search engine for multimedia databases namely CROEQs is presented that works as both ontology-based query translator

Please cite this article as:
and text based search engine. In relation to the use of ontologies for the provision of intelligent and accurate search engines Kunmei Wen [27] proposed Smartch, which is an ontology-based search engine. In this approach, a ranking method is proposed while searching for concepts, instances and the relationships between them. In Smartch, the end-users’ search is performed by keywords. Once results are retrieved end users’ can use the graphical user interface of Smartch to view all instances of an ontology concept, view relationship between two entities and view all instances of a user defined concept. Table 2 presents a comparison between some of the major ontology-based query formulation tools and approaches.

3.3. Ontology-based Information Linking Approaches

The work carried out in the European TONES project [44] provides relational database access through ontologies. In this approach, data access is enabled by defining links between ontology concepts and relational data. This ontology-to-database mapping mechanism enables a designer to link a data source to an OWL-Lite ontology. While defining mappings, the designer needs to take into account that an ad-hoc identifier should denote each concept instance so that instance values cannot be confused with data items in the data source. Queries are formulated by consulting ontology-to-database mapping rules, but this rule derivation process is carried out manually by ontology and database experts [44]. Another ontology-based information linking approach with similar techniques, but for query refinement purposes, is presented in [45] and [46]. This approach stores concepts from a data source as part of the ontology and links actual data with ontology concepts. The query answers are improved by using the semantic knowledge expressed in an ontology. Database queries are transformed by using is-a, part-of and sync-of relationships between ontology concepts.

The work carried out in designing ontology-based interactive information retrieval interfaces [47] provides an ontology-based web information retrieval system. This approach works as an interactive information retrieval system where end-users are guided through an ontology (OWL-based) driven graphical interface to define the search criteria. This work mainly addresses the problem of “where to start in the usage of an ontology-based IR interface”; that is, which elements of the ontology should be provided to the user to begin the search specification [47]. Accordingly, a user first selects a relevant domain in order to start building a query. The interface then provides a number of search entry points along with their descriptions. Once the user selects the desired ontology elements, web information elements are retrieved by following the static ontology-to-web links.

In the SemanticLIFE project [48], a front-end approach guides the users in generating data requests. The SemanticLIFE system integrates multiple data sources and stores them in an ontological repository. The Virtual Query component of the SemanticLIFE system allows semantic query writing on the ontological RDF-based repository. Users are provided with an overview about the system data through a Virtual Data component which stores the extracted metadata of the data sources in the form of an ontology. The approach provides a query engine, which recommends the query patterns according to the users’ querying context. Since it is based on a common ontology mapped from the local data source ontologies, this approach can refine users’ queries and create sub-queries over the local data sources.

3.4. Ontology-based Query Refinement Approaches

Ontology-based query refinement approaches aim at enabling end-users to make an improved formulated query. These approaches attempt to improve information retrieval by replacing or adding extra terms into an initial query. Most of the existing query refinement approaches include both query rewriting and expansion operations. Using these approaches end-users are provided interaction with candidate expansion terms based on concept hierarchies which stem naturally from the developed domain ontologies and associated ontological schema. This section discusses these ontology-based query refinement techniques that have been introduced over the past few years such as Thesaurus Ontology Navigation [49] and [50], Ambiguity-Driven [51] and [52], Information-Need Driven [53] etc.

Query expansion implementations (e.g. [49] and [50]) use thesaurus ontology navigation for query expansion. These approaches use the WordNet ontology (http://wordnet.princeton.edu) for query expansion and adapt basic keyword search mechanisms using keywords, which are identified in the ontology for a matching concept. Another approach based on this thesaurus ontology navigation approach is the Knowledge Sifter [29]. Knowledge Sifter is a scalable agent-based system that supports access to heterogeneous information sources and relies on the agents technology for query refinement. In this approach, a user query formulation agent supports user query specification to access multiple ontologies using an integrated conceptual model expressed in the OWL. This user query formulation agent also consults the ontology agent to refine or to generalise a query based on the semantics provided by the ontology.

In QuOnto [54] and MASTRO [55], the query answering process is performed through query rewriting. Both of the MASTRO and QuOnto systems adapt a similar ontology based query answering service [55]. In these approaches, end-user queries are first reformulated on the basis of ontological intensional knowledge, and then they are evaluated by a database engine using a means of predefined mappings. Database views are defined for ontology concepts and roles using SQL queries are specified in ontology-to-database mapping declarations. In [56] an ontology-based tool to convert a natural language query into nRQL query has been proposed. To achieve the conversion, first a pre-populated dictionary is used to search the synonyms of query terms. If no matching records are found then the ontology search is performed, which results in extracting a sequence of entities represented in form of triples. Finally, nRQL query is generated based on the resultant information. The ontology-based query refinement approaches such as the Step-By-Step Query Refinement [52], examine query ambiguity in relation to both structural and
Table 2: Comparison of ontology-based query formulation tools/approaches

<table>
<thead>
<tr>
<th>Tools/Approaches</th>
<th>Query support by semantic clause</th>
<th>Query support by text</th>
<th>Replication of data not required in ontology</th>
<th>Supports multimedia database</th>
<th>Heterogeneous data-sources support</th>
<th>Natural language query</th>
</tr>
</thead>
<tbody>
<tr>
<td>CROEQS [34]</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>GRQL [21]</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ontogator [23]</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>OntoQF [25]</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>OntoViews [24]</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Smarch [27]</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SEWASIE [22]</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TAMBI [4] and [20]</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TRANSForM [33]</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>VISAGE [26]</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ontology and Natural Language [35]</td>
<td></td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>OPTIQUE [36]</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>KIRA [37]</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ATHENA [38]</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Using ontology SPARQL [39]</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Pay-As-You-Go Method [40]</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ontop [41] and [42]</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Querying via OWL 2 QL [43]</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

4. Discussion: Ontology-based Information Retrieval

In Section 2, benefits, weaknesses, power and effectiveness of current mature ontology development languages in relation to query formulation have been highlighted, and it has been noted that OWL has greater support for expressing semantics when compared to RDF and RDFS. According to the literature review of ontology-based information retrieval in Section 3, it appeared that the focus has been on (1) visual or interactive query formulation; (2) information linking approaches; and (3) query refinement approaches. Ontology based visual or interactive database query formulation systems use visual representations to express the search criteria. Most of these systems are based on the RDF structure and support the specialisation or generalisation of the base or filler ontology concepts in order to build database specific queries interactively. However, it may be concluded that much of this work has been (e.g., [57], [21], [29] etc.) towards interactive query generation through non-directed graphs. Other approaches (e.g., [44], [46], [58], [48] etc.) store all data from a data-source as part of the ontology (as ontology instances) or link it directly to ontology concepts. But, often it is not practically feasible to store all data as part of a certain domain ontology, especially for systems with large amounts of data. Data that are stored as part of the ontology often need to be loaded in memory to perform the Select query operations. Moreover, this may become both a complex and time consuming activity to directly link all database instances to associated ontological concepts. Furthermore, it appears that a limited number of query formulation approaches reviewed in the literature review build on the assertion capabilities of ontologies. Thus, it can clearly be deduced that these approaches need to be extended to include further details, such as what needs to be included in the ontology from database along with the domain knowledge needed to initiate the query formulation process, to enable ontology-based query formulation based on the ontology semantic and assertion capabilities.

As discussed before, a domain ontology could be used for representing domain metadata with related semantics extracted from the relational database schema. To achieve this, first there is the need:

1. to identify the extent to which domain metadata and relationships from a relational database can be transformed into a domain ontology schema, and
2. to identify a systemic approach to transform this selected domain metadata and relationships into the domain ontology schema.

Please cite this article as:
Moreover, a description logic based knowledge representation formalism is well suited for modelling domain knowledge (also called assertional knowledge) in a domain ontology in terms of concepts and properties. However it may not be necessary or even possible to use all of OWL's description logic constructs for formulating queries. Therefore, there is a need to identify OWL constructs that can be utilised to specify domain specific knowledge as concept restrictions for the purpose of formulating relational database queries. In addition, there are significant differences between OWL ontology statement constructs and relational query statement constructs. Furthermore, the OWL concept restrictions could be either simple or complex potentially involving many conditions. As a consequence, ontology-driven relational query formulation is not only dependent on the underlying relational database schema structure, but also on the translation of individual and different combinations of ontology statement constructs into relational query statement constructs.

An ontology guided (relational) query formulation process also needs to take into consideration aspects of ontology modelling, processing and integration of domain knowledge based on the underlying relational database models and mapping to ontological queries to a relational database schema. The following sections first review the state of the art in this field and build a case to inform the usability of combining both ontology-to-database transformation and mapping approaches for relational database query formulation.

5. Approaches to Perform Database-to-Ontology Transformations and to Define Ontology-to-Database Mappings

In order to specify the schema structure of a domain ontology for relational query formulation, one requirement is to represent the domain metadata along with the semantic relationships in the underlying relational database schema into the ontology schema. Thus, to represent a relational data model in an ontology model, the transformation of the relational model into the ontology model remains an essential requirement in order to achieve correct transformations. This is because an ontology generally contains the definition of the concepts and their relationships for a given domain, as well as the domain rules (e.g., cardinality, disjointness etc.) that restrict the semantics of concepts and the conceptual relationships in a specific conceptualisation of a particular application domain. In contrast a relational data model, on the contrary, represents a specific conceptualisation of a particular application domain.

R2O is an extension of recent approaches such as D2R-MAP [60], extended D2R [61], R2O [62], VisAVis [63] and in [64]. These mapping approaches relate each construct in the relational database schema to a construct in the ontology schema and ignore unrelated constructs between them. The R2O (Relational to Ontology) [62] approach, for example, is an extensible and declarative language to describe mappings between relational database schemas and ontologies implemented in RDF(S) or OWL. The most important aspect of this approach is the use of the database schema and the ontology without adaptation. Also, this approach defines a declarative specification of the mappings between its modelling components. The ontology-to-database mappings are defined as a set of mapping elements that relate a relational DB schema to an ontology schema. This means that database tables, columns, primary and foreign keys, etc., are related to domain ontology concepts, attributes, relationships, etc.

R2O is an extension of recent approaches such as D2R-MAP [60] and extended D2R [61], which lack expressiveness in terms of writing complex mapping statements and are not considered fully declarative [62]. As a consequence, the R2O mapping language has been considered sufficiently expressive to cope with complex mapping situations arising from low similarity between ontology and database models [62]. The mapping definitions generated by R2O are not intended to be generated manually, and therefore they cannot be read or updated without using its specific GUI. Such an ontology-to-
database mapping approach is not sufficient to be used for ontology creation for query formulation.

An approach for automatic database to ontology mapping that satisfies both information and query preservation properties of semantic mapping is presented in [74]. Here, information preservation is the ability to recreate original database from mapping results and query preservation is the ability to translate each relational query over a relational database into an equivalent semantic query over resulting RDF graph. D2RQ [69] extracts the contents of a relational database to an RDF graph as per the mappings specified in a mapping language which is also expressed in RDF. Another tool that is heavily influenced by D2RQ [69] is Automapper [75]. Automapper creates an OWL ontology through SPARQL [76] to describe a relational schema. The feature that separates Automapper from other ontology mapping approaches is that the generated OWL ontology is also enhanced with SWRL [16] rules to express the constraints, such as the primary key or attribute datatype restrictions.

In [77] a three phased approach to extract ontology from a relational database is presented. In the first phase, ontology TBox is written using relational schema by generating classes from both referenced and referencing columns. Here, the referencing column is defined as a subclass of the one related to the referenced column. In the second phase, ABox is written using DB values, and in the final phase, reasoning is performed to extend the ontology.

In [78] an automated ontology construction approach is presented that considers relational schema. In addition to trivial table to class transformations, the proposed mappings in this approach include some additional rules in relation to a table key column(s) and other constraints. These include map foreign key column to ontology object property and non-foreign key column to ontology datatype property. Moreover, it maps primary key and unique constraint to ontology InverseFunctionalProperty, and NULL constraint to minCardinality constraint with minCardinality of 1 [78]. Thus, this approach can be useful for the transformation of both relational database structure and its constraints to ontology.

Table 3 presents a comparison between some of the major database to ontology mapping tools and approaches discussed in this section.

### 5.2. Database-to-Ontology Transformation Approaches

The database-to-ontology transformation approaches assume that only a relational database exists while an ontology is produced from the relational database by applying certain transformation rules [73]. Such approaches include learning ontologies from relational databases [65], mappings from relational to OWL model [66], relational databases to OWL ontology [67] and rule-based transformation of SQL relational databases to OWL ontologies [84]. Most of these approaches result in specifying an ontology that has the same flat structure (i.e. classes and instances) as the original relational database (i.e. relations and columns). These approaches utilise automatic or semi-automatic transformations of relational databases to ontologies. Here the transformation is based on a set of rules specifying how to transform the constructs of a relational model to develop an ontology with the relational model as its domain. The basic transformations adopted in these approaches are very similar, where the constructs of a relational model such as tables, columns, datatypes and constraints are transformed in an ontology model as classes, instances, properties and constraints, respectively. In such a scheme, a table is represented as a class, unless all its columns are foreign keys to other tables. The reference column of a foreign key is represented as an object property. A column is represented as a datatype property accompanied by a maximum cardinality of one, unless it is a foreign key. The column constraint Unique is represented as an inverse functional property. Similarly, Not Null is represented as a minimum cardinality of one. A primary key is represented as both an inverse functional property and a minimum cardinality of one. Finally, a row is represented as an ontology instance.

Another database-to-ontology transformation tool, RDB2ONT [85] can be used for generating OWL ontologies from relational database systems. RDB2ONT describes a formal algorithm that uses relational database metadata and structural constraints to construct an OWL ontology whilst preserving the structural constraints of the underlying database [85]. The RDB2ONT tool has two major components: the OWL Builder and the OWL Writer. The OWL Builder extracts metadata and structural constraints from a relational database system and builds a model. This model is then used to generate an OWL ontology describing the underlying database. This approach is less complex when compared to D2R-MAP[60], D2R [69] or R2O [22] which are based on creating a new complicated mapping language. This is mainly due to its simplicity and easy of configuration for defining concrete data mappings.

Triplify [86] is an RDF extraction tool from relational schemas, which uses SQL queries to select subsets of the database schema and maps them to ontology classes and properties. The advantages of the Triplify tool include predefined configurations for the schemas and the use of easily understandable SQL queries for the mapping representation. Another such tool is SquirrelRDF [87], which uses predefined mappings to extract database contents as RDF triples. Its use is fairly simple, but it does not offer the wide range of features when compared to D2RQ [69] as explained earlier. Later, Tether [88] has provided some refinements and solved several shortcomings of the native translation of a relational database to RDF approach. Most of these refinements; such as to reduce the size of the RDF graph, were specific to cultural heritage domain.

In OntoQF [25] an ontology is generated from a relational database (model) that stores the domain metadata, semantics and related domain knowledge. The major differences of this approach with other approaches is that in OntoQF (1) the transactional data remains at the original data source(s); (2) it can handle the database relations up to third normal form; and (3) the resultant transformation describes the original database relationships [25]. Moreover, in OntoQF, the ontological descriptions are based on domain metadata.
Table 3: Comparison of database to ontology mapping tools/techniques

<table>
<thead>
<tr>
<th>DB to Ontology Mapping Tools / Approaches</th>
<th>Ontology to DB Mapping Support</th>
<th>Mappings defined using Ontology</th>
<th>Mappings defined using XML</th>
<th>Maps instances ontology</th>
<th>DB to</th>
</tr>
</thead>
<tbody>
<tr>
<td>D2R-MAP [60]</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>R2O [62]</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>VisAVis [63]</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>OntoQF [30]</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Extended D2R [70]</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>D2OMapper [68]</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Mapping DB to RDF and OWL [74]</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>RDB2OWL [79]</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>BOOTOX [80]</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>RDF Graph from using SPARQL [81]</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>RBA [82]</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>R²BA [83]</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
</tbody>
</table>

objects; and consequently, it supports changes and extensions to the underlying database schema. In [89], modelling of health service knowledge is done in terms of health service ontology, where the health service ontology is defined from the perspective of concept hierarchy and ontology concepts. Moreover, the matchmaking algorithm in this approach is powered by UMLS ontology. In [90] graph-based knowledge representation model has been used in query answering; in particular, a conceptual graph model is used instead of knowledge representation formalisms as used in [25].

Table 4 presents a feature comparison between some of the major database to ontology transformation tools and approaches discussed in this section.

5.3. Ontology-to-Conceptual Data Model

Ontologies allow an interaction between data held in different formats and can potentially be used as the basis to guide and validate models of particular domains. For example, a considerable amount of work has been reported which aims to transform ontologies to conceptual data models (expressed, for example, in UML or in ER) in [98], [99] and [72]. This section reviews the key relevant and mature ontology-to-database mapping approaches.

According to El-Ghalayini et. al. [98], a domain ontology can be mapped to a domain conceptual data model (CDM). In this research, several mapping rules have been proposed that guide the transformation from a given domain ontology to a corresponding conceptual schema. Another approach to building ontologies from relational databases using reverse engineering is presented in [72]. Unlike RDB2ONT[85] and RDB2Onto[71], this approach does not directly transform a relational database into an ontology; rather it uses an entity relationship model to reverse engineer a domain ontology. In this approach, OWL-DL is used as an ontology representation language and ER for data modelling. The graph transformation language, as in [100], is used for node and edge addition into an ER model. The node addition operation is used to introduce new objects into the ER model and the edge addition operation to build relationships between ER objects.

Recently, an approach to transform a domain ontology into a relational database has been presented [99] based on an algorithm embedded in the OWL2DB [101]. In this approach, OWL documents are parsed in order to generate the corresponding Data Definition Language (DDL) scripts. During the parsing and data transformation process, the system first transforms ontology classes into database table definitions; the next steps are transformations of object, datatype properties and constraints into complete DDL statements, and finally, the database is filled with class instances. In order to transform ontology classes into relational database tables, the approach uses a breadth-first search on the hierarchical levels of the ontology classes [101]. As a result, one table in a relational database is created for each class in an ontology with a one-to-one relationship between classes and their subclasses. The OWL object properties are transformed into table relationships, which again uses breadth-first search. Depending upon the local cardinality of class properties, one-to-many or many-to-many relationships between tables are created. This approach can transform all OWL-Lite syntax but only part of OWL-DL syntax.

6. Discussion: Database-to-Ontology Transformation and Mappings Approaches for Relational Query Formulation

An ontology generally contains the definition of the concepts and their relationships for a given domain, as well as the assertions and domain rules (e.g. cardinality, disjointness etc.) that restrict the semantics of concepts and the conceptual relationships in a specific conceptualisation of a particular
Database to Ontology Transformation Tools and/or Approaches

<table>
<thead>
<tr>
<th>Database to Ontology Transformation Tools and/or Approaches</th>
<th>DB to ontology transformation support</th>
<th>Output ontology in OWL format</th>
<th>Ontology extension by reasoning</th>
<th>Preserve structural constraints</th>
</tr>
</thead>
<tbody>
<tr>
<td>RDB2Onto [71]</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>RDB2ONT [85]</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Learning ontology from relational schema [78]</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Generating OWL ontology from RDB [77]</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Learning ontology from RDB [65]</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>DB2OWL [91]</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>OntoQF [25]</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Mapping RDB to OWL structure [92]</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Transforming of RDB to OWL [93]</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>RDB2RDF [94]</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Mapping RDB into Ontology [95] and [96]</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>A systematic mapping via reverse engineering [97]</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>

Table 4: Comparison of database to ontology transformation tools and techniques

application domain. A relational data model, on the contrary, represents the structure and integrity of the data elements of the in principle, single specific enterprise application(s) by which it is being used. A relational database is a collection of relations (tables) and a relation consists of a relational schema and a relation instance. The relation schema in a database consists of the schemas for the relations, whereas the relation instances are sets of tuples, also called records. Moreover, each tuple is a row and all rows have the same number of fields (columns). After reviewing such differences and similarities between ontological and relational data models it has been noted that ontologies are semantically richer than database (conceptual) schemas, because conceptual data models only aim at establishing a link between users and domain requirements, and describe a logical structure of the data. An ontology can also be used to specify domain knowledge of a specific domain of interest. Moreover, ontologies can play a significant role in information system development and have the ability to represent conceptual data models using ontological theories, for example as reported by [98] and [30]. Therefore, existing relational data models can be used to create relational (conceptual) schemas.

Ontology-to-database mapping approaches assume the existence of both a relational database and a domain ontology. However, the database-to-ontology transformation approaches assume that only a relational database exists and an ontology is generated by applying database-to-ontology transformation rules. According to the database-to-ontology transformation approaches as discussed in this paper, the process of ontology construction from relational databases involves analysing the database schemas to determine the database-to-ontology transformation dependencies. This analysis helps in determining the relational entities that may be transformed into ontology concepts. It also helps to group together or separate in occasions the information specified in a relational database table and to determine relationships between different tables. However, the database-to-ontology transformation rules are solely dependent on application requirements. Most of the existing database-to-ontology transformation approaches do not provide an exact representation of the domain-metadata in an ontology and also do not enable the generation of the respective database relations. Therefore, such transformation approaches do not particularly aid in the process of specifying concept restrictions, and also when generating complex database queries. In order to overcome these shortcomings and to use ontology definitions for database information retrieval, a combination of both database-to-ontology transformation and ontology-to-database mappings [25] may be utilised, which could be an interesting future challenge for the research community. More areas of future research are outlined in the following section.

7. Future Opportunities and Challenges

Most of the existing approaches are based on using single domain ontology for generating relational database queries. Using such approach, database domain metadata and semantics can be transformed into a domain ontology schema with domain knowledge added as concept restrictions. It is possible to adapt the same approach for multiple databases that are geographically distributed, but conceptually related with respect to a common subject area; for example, a database that manages patients treatment records and another that manages patients family history, where the domain-subject is Patients. In such cases, an integrated ontology needs to be developed to capture the common as well as the different domain metadata and related semantics of the underlying distributed databases. Moreover, the existing query formulation frameworks should be extended to develop an approach to integrate related domain

Please cite this article as:
knowledge in order to obtain a unified domain ontology that captures all of the domain concepts.

Moreover, in the past the focus has been on visual or interactive query formulation, information linking and query refinement approaches. Most of these ontology based visual or interactive database query formulation systems either use visual representations to express the search criteria or they are based on the ontology structure and support the specialisation or generalisation of ontology concepts in order to build database specific queries interactively. Other approaches store all data from a data-source as part of the ontology or link it to ontology concepts. However, it appeared that, with the exception of few approaches, there is still a lack of knowledge-driven query formulation approaches that build on the assertion capabilities of OWL-DL, such as OWL-2. Moreover, there is a need to extend the existence approaches to answer the questions like: What needs to be included in the ontology from database along with the domain knowledge needed to initiate the query formulation process, to enable ontology-based query formulation based on the OWL-DL semantic and assertion capabilities. Plus how such domain knowledge can be automatically evolved and extended within the existing domain ontology? Furthermore, this work may be extended to further domains; and in particular, for enhancing the searching capabilities of massively loaded information management systems such as national statistical survey portals and context-aware environments that mobile devices are part of.

As per the literature review presented in this paper, it also appeared that currently there is very limited tool support available for the direct specification and manipulation of ontology domain knowledge for query formulation. In most of the presented systems; domain experts, with the help of knowledge engineers express domain knowledge in terms of ontology statements that include the definition of property constraints for concepts and individuals. In this way, whenever related knowledge about real world concepts change or when new knowledge is added into the domain ontology, the related ontology file is reloaded into the ontology server. In order to enable domain experts to directly specify and manipulate domain knowledge, it would be an interesting future work to empower the ontology knowledge component of such systems with tool support for the direct (on the fly) specification and manipulation of related domain knowledge in the ontology server. However, leaving ontology knowledge specifications completely for domain experts, who usually do not have description logic experience, may end-up in defining inconsistent domain knowledge specifications from real world knowledge, and hence this needs to be controlled.

There are also various challenges associated with the management of large data sets (so-called Big data) include structuring, search, analysis, visualisation e.g. as detailed in [102] and [103]. Due to the large and complex data sets it becomes difficult to process stored information using traditional data management tools and processing applications. Currently there is little research done on investigating efficient ways to process these large data sets to get benefits such as:

search performance, creation of reference data and enable reasoning. One of the best possible ways to achieve this is by building ontological knowledge base for Big data; and that is to (a) define a semantic model of data, (b) specify domain knowledge, and (c) define links between different types of semantic knowledge. Hence, ontologies can be used to discover information from Big Data. Based on these clear prospects, future research efforts can be towards an investigation of the feasibility to use ontological knowledge base for the specification of Big data meta data as a foundation to efficiently discover useful information for analysis. Such research efforts will need to find answers for research questions such as: (1) To what extent the meta data from Big data (NoSQL DB) can be extracted and aggregated? and (2) How can the extracted Big data meta data along with domain knowledge be represented to be the foundation of knowledge discovery?

References


Please cite this article as:

The use of Ontologies for Effective Knowledge Modelling and Information Retrieval

Kamran Munir*

Department of Computer Science and Creative Technologies, University of the West of England (UWE), BS16 1QY, Bristol, United Kingdom.

M. Sheraz Anjum

School of Electrical Engineering and Computer Science, National University of Sciences and Technology (NUST), H-12, Islamabad.

Abstract

The dramatic increase in the use of knowledge discovery applications requires end users to write complex database search requests to retrieve information. Such users are not only expected to grasp the structural complexity of complex databases but also the semantic relationships between data stored in databases. In order to overcome such difficulties, researchers have been focusing on knowledge representation and interactive query generation through ontologies, with particular emphasis on improving the interface between data and search requests in order to bring the result sets closer to users research requirements. This paper discusses ontology-based information retrieval approaches and techniques by taking into consideration the aspects of ontology modelling, processing and the translation of ontological knowledge into database search requests. It also extensively compares the existing ontology-to-database transformation and mapping approaches in terms of loss of data and semantics, structural mapping and domain knowledge applicability. The research outcomes, recommendations and future challenges presented in this paper can bridge the gap between ontology and relational models to generate precise search requests using ontologies. Moreover, the comparison presented between various ontology-based information retrieval, database-to-ontology transformations and ontology-to-database mappings approaches provides a reference for enhancing the searching capabilities of massively loaded information management systems.

Keywords: information systems, ontology, domain knowledge, database, information retrieval, knowledge management

1. Author's biography: Kamran Munir

Dr Kamran Munir has BSc, M.Sc. and Ph.D. in Computer Science from United Kingdom. His primary research interests include Information Science, Big data, cloud computing, distributed data processing and knowledge management. Dr. Munir has contributed to the various European Commission projects: Asia Link STAFF (2004-2006), Health-e-Child (2006-2010), neuGRID (2011-2014) and neuGRID4You (2013-2016) in which he led the Joint Research Area. Dr Munir’s role also includes the production and leadership of UK and international postgraduate computer science degree courses such as Big Data and Cloud Computing, and frequent collaborations with students and graduates, including the conduct of collaborative work and supervision of MPhil and PhD theses.

2. Author's biography: M. Sheraz Anjum

M. Sheraz Anjum has obtained MS degree from National University of Sciences and Technology (NUST) and currently doing PhD in Computer Science. His research interests are in the areas of knowledge management, relational

*This is the corresponding author

Email address: Kamran2.Munir@uwe.ac.uk (Kamran Munir)
databases and Big data.