From Database to Semantic Web Ontology: An Overview

Shuxin Zhao, Elizabeth Chang

Digital Ecosystems & Business Intelligence Institute, Curtin University of Technology
GPO Box U1987
Perth WA 6845, Australia
{s.zhao, e.chang}@curtin.edu.au

Abstract. This paper intends to provide an overview of automated knowledge extraction and transformation from relational databases and their related sources into Semantic Web ontologies. Issues and challenges in this area are addressed. Knowledge embedded in each part of a relational database is analysed and defined. Corresponding techniques for extracting, acquiring and transforming the knowledge are highlighted. In this paper, we classify previous approaches on this work into two types. A comparison table of the first type of approach is also given.

Keywords: database to ontology, domain ontology development, knowledge acquisition from database, knowledge discovery from application code.

1 Introduction and Motivation

The Semantic Web [1] provides explicit meaning for information and data on the web in the future. One of the goals it pursues is to allow intelligent search agents to process and integrate data from heterogeneous resources at the conceptual level. Ontologies capturing and representing various domain knowledge and information, as one of the key factors to the success of the Semantic Web, need to be well developed. However, the current state of domain ontology development for the Semantic Web is still in its infancy in terms of both quantity and quality. One of the main issues is the high development cost associated with manual knowledge acquisition for ontology construction. Acquiring domain knowledge requires many resources and is time-consuming. The knowledge acquisition process for most existing ontology construction, such as Cyc [2] and SENSUS [3], is mainly conducted on a manual basis. This has become one of the bottlenecks of the ontology development. For this reason, how to effectively acquire knowledge from available resources for ontology construction in order to reduce the knowledge acquisition effort has become a hot topic in the ontology research community [4].

While the Semantic Web is waiting for a large amount of knowledge and information represented with explicit and shared semantics, a vast quantity of relational databases contain valuable business information and implicitly embed knowledge cannot participate in the Semantic Web directly. This type of knowledge

resource is too valuable to be neglected for ontology development for three reasons: *firstly*, compared to other knowledge sources such as documents, databases are carefully designed and developed data repositories which support business process applications. The conceptual model of a relational database is analogous to an ontology model such as EER and UML. Some researchers refer to UML as a lightweight ontology [5]. Some components of the database model could be directly mapped into ontology constructs; *secondly*, these databases are well-maintained. They contain up-to-date data instances and reflect timely business information; and *thirdly*, these databases are usually deployed within a network or on the internet, which are already physically available to the Semantic Web.

On the other hand, there are increasing demands for databases to become sharable and searchable across organisational and application boundaries. This kind of request is driven by constantly increasing collaboration among organisations and by the business need for their service and product information to be searched, integrated and processed without prior negotiation. For instance, patient data from individual health care systems needs to be integrated in the health care domain in order to provide better quality health care services; and travel information needs to be shared amongst travel service providers and agents in order to provide integrated travel services regarding transportation, accommodation and so forth. Ontologies as a means for knowledge sharing provide a solution to this problem. "*Ontologies give a concise, uniform, and declarative description of semantic information, independent of the underlying syntactic representation or conceptual models of information bases*" [6]. What we need to do is to transfer implicit knowledge from databases into Semantic Web ontologies, intelligent search agents can then integrate them like human agents.

Realising the needs of both the Semantic Web and vast databases, this paper intends to provide an overview and insight into the work of transforming the knowledge from databases into the Semantic Web ontologies. The rest of this paper is organised as follows: Section 2 analyses and defines the knowledge embedded in a relational database and its related sources; then brief descriptions of corresponding techniques that are used to extract the knowledge are followed in Section 3; in Section 4, the challenges of automated knowledge extraction from the resources are discussed; then a classification of previous approaches in this area with a comparison table is given in Section 5; Section 6 concludes this paper and indicates future work.

2 Implicit Knowledge Embedded in Relational Databases

This section firstly provides a description of the basic ontology constructs specified in OWL [7] and then identifies the knowledge embedded in a relational database and its related sources that can be used for ontology development.

2.1 The Semantic Web Ontology

Ontology is defined as "*a formal, explicit specification of a shared conceptualisation*" [8]. Ontologies define basic concepts in a domain, assert properties of concepts and represent relationships among concepts in a computer-usable way such as in logic-

based language so that "*detailed, accurate, consistent, sound, and meaningful distinctions can be made among the classes, properties, and relations*" [9] by machines. OWL [7], as the W3C recommendation for the Semantic Web ontology language, has three basic concepts: class, properties of class and relationships among classes which are the target of the knowledge acquisition process from databases. The class hierarchy of an ontology is formed by the construct: '*rdfs:subClassOf*' in OWL which models the '*is-a*' type of relationships between classes. Other associations between classes are realized by defining and restricting on properties of classes such as using '*objectProperty*' or by defining class axioms.

In a relational database and its related sources, the knowledge of classes, their properties and relationships in a domain can be obtained from three parts which are the database schema, the data instances, and the applications that are built upon the underlying database.

2.2 Relational Database Schema

Relational database schema specifies the structure of the data held in a database and many business constraints. The conceptual model such as an ER model or an UML class diagram that a database schema implements describes a collection of domain concepts and their relationships which are analogous to the classes and their relationships of an ontology. For this reason, database schema has become the dominant source in previous approaches for acquiring knowledge for ontology development. Domain concepts and their properties and relationships are implied in the forms of relations, attributes, attribute data types, primary keys and foreign keys, and referential integrity constraints etc in the database schema. According to [6, 10-13] a relational schema consists of the following constructs that may be used for ontology development :

- *relations*
- *attributes of relations*
- *atomic data type of attributes*
- *constraints of attributes such as unique, not null*
- *primary keys/foreign keys*

As mentioned in [14], these represent a model of the real world appropriate to the database. One important limitation of database schema is that it restricts its attention only to these parts of the real world of direct relevance to the stored information and it is a static model of the real world. The input form of database schema is usually the logical model which can be easily generated from most of the DBMSs. Through analysis of these constructs and their correlations, a major concept frame in the domain can be obtained.

2.3 Database Instance

Database instances contain concrete and timely data and business information, and implicitly embed up-to-date domain knowledge. It is the input for creating ontology instances of the resulting ontology generated from database schema. Besides,

database instances can aid in clarifying the semantics of poorly named attributes through the analysis of data value and data correlations. Furthermore by applying data mining techniques, database instances can also be used to reveal patterns, association rules in the information. This type of previous unknown knowledge is not specified in the relational database schema and may be used to generate axioms of the ontology. One issue is worth to mention regarding to the frequent changing nature of database instances, therefore, the ontology instances need to reflect the changes in time. A process for generating ontology instances from database instances dynamically on the fly is thus desirable.

2.4 Application Source Code

Applications built on an underlying database are another important means of verifying and identifying domain knowledge. In general, a database supported application that was developed to facilitate a certain business process consists of three components: *database*, *data manipulation code* and *user interface*. The database provides a persistent data repository needed by the business processes at one end and the user interface displays the *intended meaning* of the data to the user on the other end. Data manipulation code sits in the middle, links user interface and database, carries out data manipulation of the backend database via SQL queries and pre-defined stored procedures upon user requests according to predefined business rules, and performs interpretation of data in the database to their intended meanings in user interface [15]. In this architecture, data instance in databases is only a codification of some facts [15]. It alone, without the specification of the database schema and the interpretation of application source code, does not specify any explicit meaning. On the other hand, the user interface provides users with domain agreed and user-friendly terms for the data in the database which is processed and populated by the data manipulation code. This can be used to verify and clarify the semantics of database schema. Besides, the application code itself embeds business rules which are most likely beyond the definition of the underlying database. For example, the knowledge of calculating the total amount of a customer order which is spread over more than one table may be obtained from the application code. This kind of hard coded knowledge can be extracted for axiom construction in an ontology.

The three parts of a database and its related sources are consistent in a database supported system. They represent the domain knowledge from different aspects and complement and verify one another during the process of knowledge acquisition. However, knowledge embedded in a database and its related sources rarely covers a complete range of the knowledge about a domain. Therefore, once a domain ontology is created from one input database and its related sources, it needs to incorporate knowledge obtained from other sources such as other databases or documents.

3 Techniques of Knowledge Acquisition from Relational Database

The knowledge embedded in each part of a relational database has its own structure, syntax and characteristics. Thus, techniques used to extract the knowledge are based

on the input form of each part of the relational database. We can only provide a highlight of some commonly used techniques in this area due to the page limit.

3.1 Database Reverse Engineering

Database reverse engineering aims to recover the data model of an existing database in order to apply the data model to a new application setting. An EER model or an UML model is the commonly used target of the reverse engineering process in previous approaches. Based on [16, 17], we define database reverse engineering as:

"The process of analyzing a specific database implementation and to perform concept abstraction in order to reconstitute the data asset of an existing system and apply it to a different context"

Through database reverse engineering on the logical model of an existing database, relations can be classified as base relation, dependent relation and composite relations [10, 13], which may be indicators of a concept, specialisation of a concept, or a relationship between concepts. There are three types of correlations (key correlation, attribute correlations and data correlation) of an input database that may be used to identify relationships among concepts and to decide on whether a concept or attribute should be derived. Key correlations, i.e. primary keys and foreign key, are the primary means of identifying relationships among concepts [6, 10-13]. Analysing attribute correlation including attributes equity, overlap, inclusion and disjoint and the like, across relations [10] can help identify different types of relationships among concepts including subtype of a concept; and finally, perform data correlation analysis on instant data may also assist to verify uncertain concepts, attributes and relationships.

3.2 Mapping Technique

The mapping technique is usually used in conjunction with reverse engineering in the context of developing an ontology from databases. It maps the data model derived from the reverse engineering process or the logical model to an ontology language such as F-logic [18], RDF [19] or OWL [7] by specifying the corresponding counterparts between two languages. The mapping is usually performed through a set of predefined mapping rules which specifies the semantics of the mapping between the two models. For example, some atomic data types in relational schema may be mapped to XML schema 'datatypes'.

3.3 Data Mining

The emergence of data mining techniques is resulted from the need to transfer huge amounts of available data into useful and meaningful information and knowledge. It is also known as Knowledge Discovery in Databases (KDD) [20]. Data mining is "*to apply data analysis in order to discover previously unknown, useful patterns and relationships in large data sets*" [21]. Data mining can be used for marketing, fraud

detecting, and terrorist detecting and for intelligent e-agents gathering and associating information in an information-rich environment such as the Semantic Web. The goal of data mining can be prediction or description [20]. By applying data mining on database instances, one may discover 'human-interpretable patterns describing the data' or may use existing information to make reasonable prediction regarding future activities [20, 21]. Some commonly used data mining methods include classification, association, clustering and sequence and path analysis and so forth. In database to ontology context, axioms of the resulting ontology may be created from the knowledge discovered from data mining on database instances.

3.4 Information and Knowledge Extraction

Information Extraction (IE) and Knowledge Extraction (KE) are emerging research areas over the past decade as a consequence of the dramatically increasing volume of electronic text including plain text and semi-structured text. IE transforms input text into information that is more readily digested and analysed. "*It isolates relevant text fragments, extracts relevant information from the fragments, and then pieces together the targeted information in a coherent framework*" [22]. IE has yielded NLP and machine learning techniques to extract useful information from text. IE and KE techniques may be used to extract useful information and knowledge from application source code. Approaches for extracting knowledge from web pages have been proposed by automatically detecting extraction rules, useful patterns [23, 24].

3.5 Application Reverse Engineering

Application reverse engineering aims to comprehend legacy systems for system maintenance and reconstruction purposes in the instance that the analysis and design documents are not available. Reverse engineering research on application code level analysis has been successful since early 1990s [17]. It has produced the capabilities for decomposing a system into subsystem, concept synthesis, program slicing and dicing, and analysing static and dynamic dependencies and the like[17]. Current trends of application reverse engineering include the object-oriented approach, component-base approach and the incremental approach [25].

As we can see from the above description, the techniques used to extract knowledge from different parts of a database and its related sources are divergent. Each of the techniques has their own focus. It is important to effectively integrate and combine them into a consistent framework and apply it to the context of automated knowledge acquisition from a database and its related sources.

4 Challenges in Automated Database to Ontology

Knowledge acquisition from a relational database and its related sources in an automated fashion presents many challenges. The first challenge comes from the vagueness of the input source of a database. The semantics of data is not explicitly

defined in the relational schema. Knowledge of a specific domain is user-oriented and ad hoc. Necessary assumptions on the original design model of the database and its related sources must be made and user intervention or user verification cannot be avoided during or after the knowledge acquisition process. Therefore, fully automated knowledge acquisition is almost impossible. *Secondly*, knowledge embedded in the input database and its related sources is incomplete as databases are usually designed to support a certain business process or to solve a particular problem in a domain. They most likely only cover a certain scope of the domain knowledge. Therefore, knowledge acquired from this input should be enriched by other knowledge sources in order to represent full domain knowledge. The input forms of other knowledge sources can be divergent in many aspects such as their physical presentation, structure, coding languages and so on. To automatically obtain integrated knowledge from these heterogeneous sources requires many joint efforts from each of those areas. *Thirdly*, there exist many issues regarding the implementation of a running database in the real world. This includes database redundancy, poor naming of database relationships and attributes, and poor database design which did not follow good design principles and so forth. This kind of problem is hard to foresee for each individual database implementation. *Finally*, knowledge acquisition from a database system is an interdisciplinary research area. The automated knowledge acquisition process from databases involves many interrelated disciplines and requires a synthesis of techniques from different areas in order to discover the knowledge accurately without information loss. One of the key issues to be solved is that of how to integrate the output from different techniques on different inputs of a database system.

5 Classification of Previous Approaches on Database to Ontology

The previous approaches on database to ontologies can be grouped into two types. The first type of approach generates an ontology model from an input database model through reverse engineering, mapping techniques and other techniques, then create ontological instances from database instances based on the previously generated ontology. This group of approaches includes [6, 10, 11, 26-28]. Key correlations are a major means of identifying relationships between concepts. Attributes correlation [10] and the use of application source code such as HTML forms [27] have begun to be examined and combined into the database reverse engineering process while data correlation is rarely examined. In Kashyap [6], more than one database of the same domain is used to extract the domain knowledge. This approach also utilised a domain specific thesaurus which contains standardised vocabularies and used user queries to refine the ontology model generated from database schemas. An overview of these approaches is shown in Table 1. As we can observe from the overview table, there is no approach that has examined the knowledge embedded in a relational database in its full dimensions. In addition, some approaches need to be tested with real world examples.

The second type of approach proposed mapping languages that directly map database into Semantic Web data syntax such as OWL [7] without analysing database