Environment. Technology. Resources, Rezekne, Latvia
Proceedings of the 11th International Scientific and Practical Conference. Volume II, 24-29

Ontology-Based System Development for Medical Database Access

Henrihs Gorskis, Ludmila Aleksejeva, Inese Polaka
Riga Technical University, Faculty of Computer Science and Information Technology

Abstract. Medical research is a complex multi-disciplinary task involving specialists from different fields and professions, not only medical professionals. Medical databases are structured by information technology experts, but the contents must be tailored to the medical field. When the medical staff defines the information they use, terminology from their particular field of expertise is employed. This leads to misunderstandings between the maintainers and developers of information technology solutions, and the users of those solutions. When the time comes that a user, who is a medical professional, requires very specific data from the database, the chance of obtaining the data incorrectly is very high. By defining specific concepts and relationships between the data, in an explicit shared specification, some of the above problems can be avoided. The developed ontology-based data access system, described in this paper, provides a tool to store, manage and use definitions of common terminology and their mappings to the database. It is also capable of reasoning about the relationships between terms and indicates inconsistencies of term definitions, if any are present. By defining these interconnected terms in the ontology and by working through the system, all experts and software tools, who use the data, are able to use and reuse these terms to obtain data in a reliable and predefined way. This paper discusses the development and implementation of the ontology-based data access system, the ontology describing the medical data and the data mapping system, linking data from the database to concepts and virtual ontology individuals.

Keywords: Database analysis, intelligent system development, ontology.

I. INTRODUCTION

There are cases when an information system can be built using standardized blocks and known approaches. In the case of a small online store or a similar field, the technology and development steps can be known even before development starts. In some other cases, minimal analysis of the field is required and after finalizing the users' specifications of the information system, development can begin. However, there are fields where development is much more difficult. Specifications cannot be fully determined beforehand or can change over time. Medical research, or any research can be such a field. In the case of medical research, the procurer of the system may envisage the need to catalogue some medical procedures and a fixed pipeline of analysis for the research. However, the procedures, the order in which the procedures are commissioned, the types of analysis, the desired participants and many other aspects may change over time. There are no fixed approaches for such situations. Furthermore, any knowledge about the domain may exist only within the currently used system in a derived state. Any knowledge that went into the development may be found in the documentation. However, the documentation exists only in the form of informal text meant to be read by humans. Frequently, only the original developer truly knows how the system works.

The uses of the data structures in the existing system are only truly known to the developer. They might also have changed over time. Understanding the significance of database columns may not be sufficient to understand the data. Some columns might not be used anymore. Some values might be outdated. New structures replace the old ones, while still maintaining some aspect of the old system. This all leads to confusion and makes maintenance and improving of the system increasingly difficult.

Another aspect to be considered is that databases for scientific research may be accessed not only by administration personal, but also by researchers to analyse the data. These requests for data may be very different. Different researchers are interested in different data. This means that standard reports or data extraction solutions may not be possible to implement, since the requests for data are constantly changing. Providing all the data without structure and context will lead to misunderstandings, or the inability of the researcher to do anything useful with the data. Providing a standardized report solution may work for a limited time and only for a certain repeating task.

That all necessitates using knowledge in the system. When knowledge about the domain and the system's inner working is embedded into the system itself, it becomes accessible and usable by both different human users, software agents, and different modules of the information system itself. This paper

ISSN 1691-5402

http://dx.doi.org/10.17770/etr2017vol2.2572

describes the development of an ontology-based system for extending an existing ontology-less information system, for the medical field. This is done by adding ontology reasoning capabilities to information access. The ontology allows for the storage of knowledge about the field to make it possible to retrieve data from a complex relational database in a simple and intuitive way. Information retrieval becomes paired with reasoning.

II. EXISTING SOLUTIONS

Using ontology as a knowledge extension is not a new idea; it is employed in many different fields [1] - [3]. The use of related semantic meaning to mostly plain data offers additional opportunities desired by both system users and developers. Using ontologies in the medical field has gained popularity in recent years [4] – [7]. This is partly due to medical information being often complex in structure and using complex terminology. There already exist multiple solutions for accessing databases using ontologies or other semantic technologies [8]. Although, ontologies are offering two levels of descriptions, the TBox and the ABox, it is apparent that separating these levels can be beneficial. This can be achieved by using the ontology only to describe terminology and higher conceptual relations and applying it to data, stored separately. However, we find that the existing solutions are lacking in certain aspects, mostly due to the way the knowledge is stored [9]. Many of these solutions require that the database is built from ground up to the specifications of the solutions. These solutions propose a triple store database. These are specific databases meant for the storage of subject-predicate-object triples. These triples are the smallest unit used to describe the concepts of the ontology. Although, these solutions are capable of storing and retrieving information using ontology knowledge, they are very difficult for the task of adding knowledge-based support for an existing software solution. Restructuring the existing database in this way is often not viable. This is due to all previously developed solutions being tuned to the database technology and data structures. Also, converting a database into a network of related data, as is the case with triples, the retrieval of data, as it is possible in relational databases, becomes more complex and resource intensive. This would raise the usage of resource not only for any new software solutions, but also for any existing solutions.

Another shortcoming is the complexity of retrieving and using the knowledge. The current way of retrieving knowledge from an ontology is to use SPARQL queries over the RDF data [10]. Having an ontology describing the data can offer additional advantages over classical data description solutions. However, if the complexity of correctly obtaining data increases with the addition of ontology, it is less likely to be used by the average user. SPARQL queries are not simpler than SQL queries, in fact they are in many aspects more complex. In order to make the addition of ontology to an existing solution an improvement, we prioritize ease-of-use and ease-of-integration.

The developed solution described in this paper is able to connect and use an existing database to obtain data on which to perform induction and deduction reasoning, without making any changes to the database or requiring any additional compliancy. The developed solution extracts data using simple SQL queries, making it no different from any other software module accessing it. Many existing solutions are based around the usage of the language and conformity to all linguistic aspects of ontologies. Many solutions will put their conformity to RDF at the centre of the solutions. This is due to the most popular ontology language OWL being and extension of RDF and RDF schema. The proposed solution aims at making it easy for the user and developer to use ontology specific capabilities, by concentrating on what makes an ontology an ontology, instead of viewing an ontology as an additional meta-layer to RDF. The descriptive and reasoning capabilities of an ontology are interesting in themselves and could be used to perform tasks in a stand-alone way. Ontologies describe the class layer of individuals, relations between them and the attributes of properties themselves. This can be used to create usable descriptions without making use of lower level structures. However, this requires that the ontology used with the system is created in a certain way. The development of the system is tuned to its use.

III. DEVELOPMENT OF THE SYSTEM

System development starts with the analysis of the field aimed at creating an ontology for it. There are multiple types of ontologies. Some ontologies are very abstract and describe high-level concepts, such as time and what a place is. There are more specific ontologies describing a certain domain or field of interest. At the lowest level, there are application ontologies, defined for a certain application only. In the case of the developed system, the ontology is a combination of a domain ontology with database specific mappings and based on the structure of the database. Since the primary use for the ontology will be to access the database and all information is obtained from the database, forming the ontology around the structure and content of the database is not only unavoidable, but also an opportunity to shape the ontology in a useful way. A downside to this is that the ontology is not a pure domain ontology. All concepts are viewed through the context of the database. If some unit of information exists in the database and describes some real-world object, some concept describing such units must exist in the ontology. On the other hand, if some concept exists in the ontology, but there is no data in the database,

which could be used to instantiate an object of this concept, the existence of such concept in the ontology can be questioned.

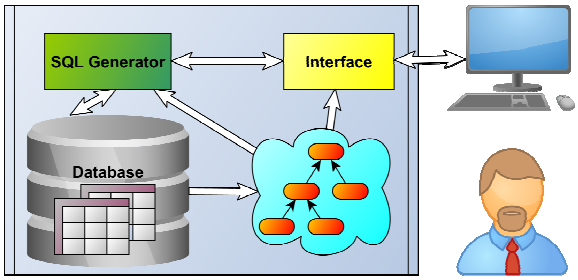

Fig. 1. System structure

Having developed the ontology, the system is capable of reading and constructing a memory model of the ontology. In this case, a new solution has been built from scratch. It reads an ontology described in OWL/XML and constructs a concept-network based ontology in its memory.

Figure 1 shows the systems schema. The system consists of the pre-existing database, the ontology, an interface module and a SQL generator. The database is analysed and an ontology is created. The created ontology is used in both the SQL generator and the interface module for selecting required concepts. The selected concept together with the definitions provided in the ontology are used by the SQL generator to obtain data. All related concepts of the obtained database records are found and provided to the user through the interface.

A. Database analysis

The first step in developing an intelligent system for an already existing solution is to analyse the inner workings of the existing solution. In this case, there was an already established database for the project. The database contains all the data relevant to the project. The database is structured around a central table containing data about participants in the medical study. Participants are defined as people who at some point agreed to participate in the study. This table contains the names, addresses and other personal information about these people. It also contains fields classifying them as different kinds of participants. One field describes their status. They can be active participants or can have been excluded from the study, for some reason. Another field describes their subtype. Participants are divided into the main group and a control group. Further still, since the study had multiple stages, the participants were further divided into participants of the current main study and participants of the previous pilot study.

When the participants first joined the study, they were given questionnaires. Each questionnaire had its own database table containing the respondents' answers. Each respondent will have either 1 or no corresponding record in each of these tables. If the respondent was excluded from the study, there will be a record for the respondent in the main table, without a corresponding questionnaire record.

Other tables describe medical procedures. Most procedures are divided into two tables. One table holds records for a distinct procedure at a specific date. Another table will hold multiple records related to the procedure. There are usually records for obtained samples (blood samples, biopsies) and some data about medical results if the samples have been analysed.

It is important to understand how the database table is used and what data it describes. It is also important to conduct a more technical analysis of the data contained in the database. Direct access to the database was not given, and only views were used for data extraction. Since all tables were accessed through table views, some important definitions were lost. Since the developed system should make use of the views instead of the tables directly, almost all information about the database structure had to be obtained from the accessible data. By using methods described in an earlier paper [11], it was possible to obtain information about distinct values and key pairs. A meta description of the database was created. In the case of a well-maintained database, such step would not be required. It would be possible to obtain such data from the definitions and the meta data of the database.

The result of the analysis is that an ontology describing the content of the database must be able to reference certain data values and relationships between tables. Each database record can be viewed as an individual of the table class. The meaning of the table class is not necessarily readily available. The only thing that can be said for sure, is that a database table is a grouping concept for its records. For example, the database of the described project has the participants table. However, it would be wrong to equate the concept of the database table to the concept of a participant. This is due to the table having records which are not real participants. Therefore, the concept of a real participant would be a sub concept of the concept describing the database sable.

B. Ontology building

Once the database is well understood, the ontology for the system can be built. There exist many different approaches to building ontologies. These approaches are described in many papers [12]. When an ontology is being created from a specific domain, often terminology is analysed. There is no wrong way to create an ontology, however, there is no guarantee of the ontology being usable, or usable for the specific task. By basing ontology development directly on a database, and for the task of ontology-based data access, the likelihood of the concepts being usable and used in the task is higher. By knowing what data are stored in the database, what groups are described by the data and understanding

the meaning of the data, it is possible to define concepts describing these groups.

The ontology is first filled with concepts describing records from tables. For each table, a corresponding table-concept is created. As described earlier, these concepts have a very specific use. They do not necessarily describe the thing they were named after and it would also be wrong to use these concepts as descriptions of database technology. Instead, these concepts are used as grouping concept for the records and are also used as mappings. When the SQL generator encounters a concept, having a database table name, it knows that this table will be used in the query. Many times, an ontology based on a database will start to describe the database more, than describing the field. Often ontologies based on databases will automatically contain generated concepts and properties based on tables and fields. This should not be considered correct for the purposes of this system. In order to create the needed link to the database, fields will be addressed, but only the most necessary fields, not all of them. This means that the knowledge engineer defines the mapping to the database as needed, instead of using all possible mappings. This is helpful to keep the ontology small enough to be understandable by humans.

For the database, the system has been developed for, the following ontology concepts have been defined. Based on the status, concepts for active participants, excluded participants, decided participants, main study participants and pilot study participants have been defined. These concepts are given a definition based on a database value. For example, the participants of the main study are defined as records of the participants table with a value of “Main” in the data attribute correlating to the table field holding this information.

Database tables can be found in the names of the special class concepts. Field names can be found in the data properties of the ontology. They are used to define complex classes. The third kind of mapping between the ontology and the database are object properties. Object properties describe relations between individuals. In a database, these are primary and foreign key mappings. In the ontology, special object properties are defined. In order to handle mapping, annotations are used. A special annotation property is defined to indicate database mapping. Using these annotation properties, the database tables and fields, participating in the relationship, are defined.

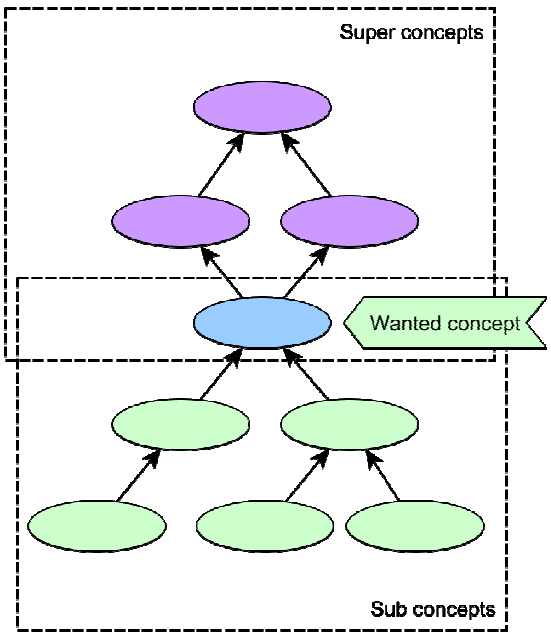

Fig. 2. Related concepts

The purpose of the system is to extend knowledge. Therefore, it is not enough to have these base types. They are only used to define mappings. The real knowledge has to be added to the ontology as sub and super classes of the mapping classes and properties.

C. Development of ontology reasoner

The developed ontology can now be used to access the database, but before that can happen, a reasoning system must be developed capable of using concepts, their descriptions and relations to perform the necessary reasoning for correct access to the database. Ontology reasoning at its core is based on extending the existing hierarchy of concepts. Each concept (class, object property, data property, datatype) is assigned some position in a pre-established hierarchy. This hierarchy can be extended when analysing complex concepts. The developed solution searches for special cases and adds new hierarchical relations, based on these cases.

Once the hierarchy has been fully extended, it can be used for classification. When a new instance of a concept is added to the ontology, complex concepts can be used to determine, whether this new instance can be classified as any of them. No additional reasoning is required, since the hierarchy already connects the new instance to all other concepts. The only additional reasoning required is to check if the individual belongs to multiple internally disjoint classes. If this happens, the ontology is inconsistent and must be checked.

D.SQL generator

The SQL generator is a direct extension to the reasoner. The developed reasoner makes it possible to traverse the relationships between concepts. It is necessary for the SQL generator, so that it may use only the most necessary tables and fields. It must generate a SQL query to obtain the needed data, without extracting the entire database.

As described before, multiple concepts and data properties are directly based on database tables and fields. Some object properties describe table key restrictions. Generation of SQL statements is directly related to the search for these concepts and properties. When a concept is selected (Figure 2), the concept hierarchy is used to traverse it in the search for these values. To determine the necessary fields

and table to classify any data with the selected class, only the subclasses need to be considered. If a database record can be classified as any of the subclasses (including the selected class), the record belongs to the selected class. If it cannot be classified as any of the subclasses, the relationship is unknown, at best. By traversing all subclasses the SQL generator looks for tables and fields referenced in the names of the concepts and relations (in the case of complex concepts). They are added to a list of distinct tables and fields. This is the smallest set of data, which has to be obtained for positive classification. Next, all super classes of the select concept should also be considered. By doing the same operation of gathering database references from the super classes, additional tables and fields may be found. The purpose of these new tables and fields is not positive classification. They are needed to test consistency. Any data that have to belong to the selected class must satisfy the inherited restrictions of the super concepts. In order to keep the set of selected data as small as possible, addition of other tables is restricted in the second step.

E. Concept selection interface

The user is presented with an interface for the selection of the desired concepts. By traversing the ontology only named classes are selected to be shown to the user. The user may select multiple concept and an "AND", "OR", "NOT" relationships and grouping between these concepts. Working together, the ontology reasoner and SQL generator can extract the smallest amount of data corresponding to these ontology concepts. The user query represents a new complex concept, which is a combination of existing concepts. The reasoner can determine the new concepts position within the existing concept hierarchy. The SQL generator obtains the data, and reasoning is performed. After the classification is done, the data may be presented to the user. In addition to selecting the desired data, all related classifications are also shown to the user. This makes the data more descriptive.

It is difficult to visualize this data. There may be multiple object relations between tables. If provided in table form, some additional approaches for visualization may need to be used.

IV. CONCLUSIONS

The described system can add ontology knowledge to an existing system or solution, because it is built in addition to it and does not require any changes to the established order. It works by mapping database tables and field to ontology concepts. The developed system provides an additional way of accessing the data in the database. Instead of using SQL queries, combinations of existing concepts can be used. Data access becomes more intuitive. This can also be helpful to the developer of the system. Knowledge about the type of information stored in the database is also added to the system. Having an ontology, the concepts important to the system have to be stated explicitly. When problems with development or the functionality arise, it is possible to investigate the definitions of the concepts to recall their meaning. In case of insufficient definitions, the definitions of concepts can be extended and improved. Improvements, in these cases, can be performed by changing the ontology and not program code. This can be beneficial to development. The described system enhances the capabilities of the system by adding new possibilities in a none-intrusive manner.

REFERENCES

[1] O. Daramola, M. Adigun and C. Ayo, *Building an Ontology-Based Framework for Tourism Recommendation Services*: Information and Communication Technologies in Tourism 2009: Proceedings of the International Conference in Amsterdam, The Netherlands, 2009, 135-147, doi: 10.1007/978-3-211-93971-0_12

[2] M. del Mar Roldán García, J. García-Nieto, J. F. Aldana-Montes, *An ontology-based data integration approach for web analytics in e-commerce*, Expert Systems with Applications, Volume 63, 30 November 2016, Pages 20-34, ISSN 0957-4174, http://dx.doi.org/10.1016/j.eswa.2016.06.034.

[3] D. Calvanese, P. Liuzzo, A. Mosca, J. Remesal, M. Rezk, G. Rull, *Ontology-based data integration in EPNet: Production and distribution of food during the Roman Empire*, Engineering Applications of Artificial Intelligence, Volume 51, May 2016, Pages 212-229, ISSN 0952-1976, http://dx.doi.org/10.1016/j.engappai.2016.01.005.

[4] R. Dieng-Kuntz, D. Minier, M. Růžička, F. Corby, O. Corby, L. Alamarguy, *Building and using a medical ontology for knowledge management and cooperative work in a health care network*, Computers in Biology and Medicine, Volume 36, Issues 7–8, July–August 2006, Pages 871-892, ISSN 0010-4825, http://dx.doi.org/10.1016/j.compbiomed.2005.04.015.

[5] J. D. Cameron, A. Ramaprasad, T. Syn, *An ontology of and roadmap for mHealth research*, International Journal of Medical Informatics, Volume 100, April 2017, Pages 16-25, ISSN 1386-5056, http://dx.doi.org/10.1016/j.ijmedinf.2017.01.007.

[6] A. Sigov, V. Baranyuk, V. Nechaev, O. Smirnova, A. Melikhov, *Approach for Forming the Bionic Ontology*, Procedia Computer Science, Volume 103, 2017, Pages 495-498, ISSN 1877-0509, http://dx.doi.org/10.1016/j.procs.2017.01.033.

[7] A. Kumar, Y. L. Yip, B. Smith, P. Grenon, *Bridging the gap between medical and bioinformatics: An ontological case study in colon carcinoma*, Computers in Biology and Medicine, Volume 36, Issues 7–8, July–August 2006, Pages 694-711, ISSN 0010-4825, http://dx.doi.org/10.1016/j.compbiomed.2005.07.001.

[8] D. Calvanese, B. Cogrel, S. Komla-Ebri, R. Kontchakov, D. Lanti, M. Rezk, M. Rodriguez-Muro, G. Xiao, *Ontop: Answering SPARQL queries over relational databases* (2017) Semantic Web, 8 (3), pp. 471-487.

[9] H. Gorskis, A. Borisovs, *Storing an OWL 2 Ontology in a Relational Database Structure*. In: Environment. Technology. Resources : Proceedings of the 10th International Scientific and Practical Conference, Latvia, Rezekne, 18-20 June, 2015. Rezekne: Rezeknes Augstskola, 2015, pp.71-75. ISBN 978-9984-44-173-3. ISSN 1691-5402. e-ISSN 2256-070X. Available from: doi:10.17770/etr2015vol3.168

[10] S. Chuprina, I. Postanogov, O. Nasraoui, *Ontology Based Data Access Methods to Teach Students to Transform Traditional Information Systems and Simplify Decision Making Process*, Procedia Computer Science, Volume 80,

2016, Pages 1801-1811, ISSN 1877-0509, http://dx.doi.org/10.1016/j.procs.2016.05.458.

[11] H. Gorskis, L. Aleksejeva, I. Poļaka, *Database Analysis for Ontology Learning.* Procedia Computer Science, 2016, Vol.102, pp.113-120. ISSN 1877-0509. Available from: doi:10.1016/j.procs.2016.09.377

[12] N. F. Noy and D. L. McGuinness, *Ontology Development 101: A Guide to Creating Your First Ontology*. Stanford Knowledge Systems Laboratory Technical Report KSL-01-05 and Stanford Medical Informatics Technical Report SMI-2001-0880, March 2001.