The African Journal of Information Systems

Volume 8 | Issue 4 Article 2

10-1-2016

Development and Semantic Exploitation of a Relational Data Model for Service Delivery in South African Municipalities

Kgotatso Desmond Mogotlane
Vaal University of Technology, desmond.mogotlane@gmail.com

Jean Vincent Fonou Dombeu
Vaal University of Technology, fonoudombeu@gmail.com

Development and Semantic Exploitation of a Relational Data Model for Service Delivery in South African Municipalities

Cover Page Footnote

The authors would like to thanks Prof. Pillay for proofreading the manuscript

The African Journal of Information Systems

Development and Semantic Exploitation of a Relational Data Model for Service Delivery in South African Municipalities

Research Paper

Volume 8, Issue 4, October 2016, ISSN 1936-0282

Kgotatso Desmond Mogotlane
Vaal University of Technology
desmond.mogotlane@gmail.com

Jean Vincent Fonou Dombeu
Vaal University of Technology
fonoudombeu@gmail.com

(Received Dec 1, 2015, Accepted August 23, 2016)

ABSTRACT

Relational databases (RDB) are the main sources of structured data for government institutions and businesses. Since these databases are dependent on autonomous hardware and software they create problems of data integration and interoperability. Solutions have been proposed to convert RDB into ontology to enable their sharing, reuse and integration on the Semantic Web. However, the proposed methods and techniques remain highly technical and there is lack of research that focuses on the empirical application of these methods and techniques in information systems (IS) domains. This study develops and semantically exploits a relational data model of the South African Municipalities Information Systems for Service Delivery. A combination of qualitative and quantitative methods is used. The qualitative part of the research is carried out with a literature review and online search for relevant resources, whereas, the quantitative analysis was done with experiments. The research provides a case study of the empirical application of semantic web technologies for converting RDB into ontology in IS.

Keywords

South African Municipality, Semantic Web, Service Delivery, Relational Database, Ontology

INTRODUCTION

The Semantic Web is an evolution of the current World Wide Web in which data and resources are represented on the basis of their meaning rather than web links as is done in the current Internet. This provides the web with content that is understood by both humans and computers. In particular, the content of the Semantic Web is represented with ontology in such a way that computers can automatically decipher it to extract useful information for users. Therefore, ontology is the backbone of any Semantic Web application.

Gruber (1993) defined ontology as an explicit specification of a conceptualization. A conceptualization defines an abstract and simple view of a domain of interest that is being represented purposively. Examples of domains are medicine, biology, e-commerce, e-government, etc. Ontology represents the semantic content of a domain using its constituents including the concepts, objects, entities and relationships between them (Gruber, 1993). Other enabling technologies for Semantic Web are: (1) ontology editors: they are software that offer functionalities for handling ontology such as the creation of a new ontology, opening of an existing ontology, querying ontology, and so on, (2) languages for representing ontology: ontologies are created in these languages for computers to process them, and (3) languages for querying ontology: these are Structured Query Language (SQL) like languages specifically designed for querying ontology, and many more.

Semantic Web technologies are increasingly being adopted in various domains (biology, geology, e-commerce, e-government, etc.) to build distributed and intelligent web applications. One of the main concerns in Semantic Web is the semantic exploitation of data stored in relational databases. In fact, relational databases are the main sources of structured data for government institutions and businesses. This is the reason behind the widespread use of relational database management systems (RDBMS) like DB2, Oracle and Microsoft SQL Server (Lin, 2008) and many websites rely heavily on databases as sources of information (Tirmizi et al., 2008; Sequeda et al., 2012). Government institutions and businesses maintain large relational databases that are populated with important data gathered over many years; these databases are dependent on autonomous hardware and software, and create the problem of data integration and interoperability (Fatima and Rajput, 2012). Solutions have been proposed to convert RDB into ontology to enable its sharing, reuse and integration on the Semantic Web (Li et al., 2005; Laclavik, 2006; Zhang and Li, 2011; Saleh, 2011; Gherabi et al., 2012; Sedighi and Javidan, 2012; Pasha and Sattar, 2012; Jain and Singh, 2013). However, the proposed methods and techniques for converting RDB into ontology remain highly technical. Further, there is a lack of research that focuses on the empirical application of these methods and techniques in real world information systems (IS) domains to strengthen their application and adoption.

This study develops and semantically exploits a relational data model of the South African Municipalities Information Systems for Service Delivery. The research provides a case study of the empirical application of semantic web technologies for converting RDB into ontology in IS using a combination of qualitative and quantitative methods. The data was collected from a literature review and online resources downloaded from 81 municipalities' public websites. Thereafter, the business rules of the domain are analyzed and the relational data model is built. Experiments are carried out to convert the relational data model into ontology. The resulting ontology is further queried and analyzed with a set of competency questions of the domain and the users' views of their answers. The results show that Semantic Web Technologies can be effectively applied to extract and import semantic data from a Municipality RDB and provide answers to users' queries.

The rest of the paper is organized as follows. Section 2 discusses related studies. The methodology of the study is presented and applied in Section 3. Section 4 presents and discusses experimental results of the study. A conclusion ends the paper in Section 5.

RELATED WORK

In recent years, the topic of ontology construction from relational databases has been of interest to many researchers. Mapping rules of database and ontology structures are presented in (Li et al., 2005; Zhang and Li, 2011; Gherabi et al., 2012; Sedighi and Javidan, 2012; Loudhi et al., 2013; Ramathilagam and Valarmathi, 2014); the mapping rules provide formal definitions of the rules that can be applied to map database structures (entity, primary key, foreign key, etc.) to that of ontology (class, property, instance, etc.). Trinkunas and Vasilecas (2007) presented an approach for building ontology from a relational database using reverse engineering. The authors define reverse engineering rules for converting relational database into ontology. The approach is further evaluated with a software, namely, Sybase Power Designer 12.0. Another study by Banu et al. (2011) presented a method for semantically querying relational database using ontology. Transformation rules of database to ontology are defined and applied to a set of *ad hoc* tables of the Library Management System.

Although the majority of studies discussed above have used a set of *ad hoc* database tables for proof of concept, they all remain highly technical. The proposed methods have not yet been applied in the real world IS domains to empirically evaluate their effectiveness and accuracy in converting relational database into ontology. None of the studies provide a detailed collection, analysis, semantic modelling and evaluation of the business requirements of the target IS domains against the resulting ontologies.

Besides the above conceptual methods for transforming relational database into ontology, many tools and algorithms exist that enable the automatic conversion of a relational database into ontology (Laclvik, 2005; Zhou et al., 2010; Pasha and Sattar, 2012; Jain and Singh, 2013). The World Wide Web Consortium (W3C) through their RDB2RDF Working Group is also developing a direct mapping standard that focuses on translating relational database into Resource Description Framework (RDF) ontology (Gherabi et al., 2012). The problem with many of the proposed tools is that they are still at the prototype stage and are not yet fully fledged products. Furthermore, these tools have not yet been applied to real world IS domains to ascertain their performance in the automatic conversion of relational databases into ontologies.

Protégé is a free, widely used open source ontology editing platform that offers great extensibility and scalability (Alatrish, 2013). Its extensibility is due to many plug-ins developed by semantic web experts. A plug-in is a separately developed software module that adds more functionality to existing software. Examples of Protégé plug-ins include OntoLT (Buitelaar et al., 2004; Nyulas et al., 2007), DataGenie (Gennari et al., 2007), OntoBase (Yabloko, 2009), DataMaster and RONTO (Papapanagiotou et al., 2006), and SIM-DLA (Mulligann et al., 2011). OntoLT enables the extraction of ontology from text within Protégé (Buitelaar et al., 2004). SIM-DLA is a Protégé plug-in that enables the comparison of ontology concepts and their meanings through the measurement of semantic similarities (Mulligann et al., 2011).

DataMaster, DataGenie, OntoBase and RONTO are Protégé plug-ins that deal with the conversion of relational databases into ontologies. However, the RONTO plug-in is still under development and is not yet available for use in the Semantic Web community (Papapanagiotou et al., 2006). Further, due to technical challenges such as unresolved errors and bugs (Gennari et al., 2007), DataGenie functionalities were improved to create the DataMaster plug-in (Nyulas et al., 2007). In light of the above, DataMaster and OntoBase are the only plug-ins for automatic conversion of relational databases into ontologies that are currently available for use in Protégé. Consequently, they are used in this study to convert relational database to ontology.

METHODOLOGY

Both qualitative and quantitative methods are used in this study. The data is collected from a literature review and online resources downloaded from 81 South African municipalities' public websites. Thereafter, the business requirements of the domain are analyzed to build the relational data model. The requirements of the domain are further modelled with a set of competency questions (CQs) using the Tropos Methodology (Fernandes et al., 2011) and translated into executable form with the Competency Question Translation (CQT) approach (Zemmouchi-Ghomari and Ghomari, 2013). Finally, experiments are carried out to automatically construct ontology from the relational data model and to run and evaluate semantic queries against the ontology.

Data Collection

The IS domain in this study is the South African municipalities information systems for service delivery (SAMISSD). A study was carried out to understand the SAMISSD domain. The South African government, through its local and metropolitan municipalities has a constitutional obligation to provide basic services (e.g. potable water, sanitation, refuse removal, property assessments and electricity) to its citizens (RSA, 1996; ELM, 2014a, 2014b). To achieve the constitutional obligation for effective service delivery, the country is divided into 234 local and metropolitan municipalities (Table 1) to ensure that all areas in the South African are served (Koma, 2010). Municipalities have tariff policies to govern the billing of major services and consumables such as electricity, water, sewerage, and refuse removal. They are also regulated by certain laws such as the Municipal Systems Act of 2000 (RSA, 2000) to ensure that they remain constitutional when dealing with the public. The relationship between a municipality and the public can be compared to that of a service provider (municipality) and customer (public).

South African Provinces	Number of Municipalities	Number of Municipalities Covered
Gauteng	10	9 (90%)
Mpumalanga	18	9 (50%)
KwaZulu Natal	51	9 (18%)
Western Cape	25	9 (36%)
Free State	20	9 (45%)
North West	19	9 (47%)
Northern Cape	27	9 (33%)
Limpopo	25	9 (36%)
Eastern Cape	39	9 (23%)
Total	234	81 (35%)

Table 1. Summary of South African Municipalities Studied

Table 1 shows a summary of municipalities that were studied in all the 9 provinces of South Africa. A total of 9 municipalities were selected and studied per province. Overall 81 (35%) of both local and metropolitan municipalities were studied. The study consisted of a review and analysis of the municipality's tariff and property rates policies downloaded from their public websites. This enabled a thorough understanding of the knowledge domain as summarized in the following business rules.

Municipalities maintain customer accounts to bill services they provide on a monthly basis. A customer can have an account with a municipality by virtue of being a property owner and occupier of a property that receives services. Monthly payments will be made to the account and failing to pay will lead the account into arrears. Customers are allowed to make payment arrangements on accounts that are in arrears. A customer will also be able to lodge a complaint or a general query in case of poor services rendered. The following services are offered to customers: water, electricity, refuse removal, basic sewerage and property assessment. The services listed above are charged according to a tariff that is influenced by many factors that include: the category of the property, market value of the property determined after municipality property assessments, consumption, and peak and non-peak months (this specifically affects consumables like electricity). The municipality is responsible for maintaining a valuation roll that is used to capture all assessed and valued properties according to property category. Property categories are listed as residential, sectional title, business, commercial, industrial, and farm dwellings. The requirements of the domain presented above are analyzed to build a relational data model in the next subsection.

Relational Data Modelling

It is necessary to define what a relational database is before building one for the SAMISSD domain. A relational database is a data model which includes sets of relationships, attributes, and basic types (Zhang and Li, 2011). A relational database could be represented in the form of a relational database schema (Navathe, 1992). The relational database schema defines the structure of the database (Mahmood et al., 2010) and consists of the following main elements (Li et al., 2005; Zhou et al., 2010; Telnarova, 2010; Zhang and Li, 2011; Saleh, 2011):

- Relation - database table with a set of columns, rows and constraints.
- Attribute - column of a database table.
- Tuple - record or row of a database table.
- Domain - data type of a column of a database table. This is the type of values contained in a column e.g. integer values etc.
- Primary Key - a constraint placed on a column to maintain entity integrity in the table. A primary key maintains unique rows in the table.
- Foreign Key - a constraint placed on a column to maintain referential integrity. A foreign key maintains relationships among database tables.

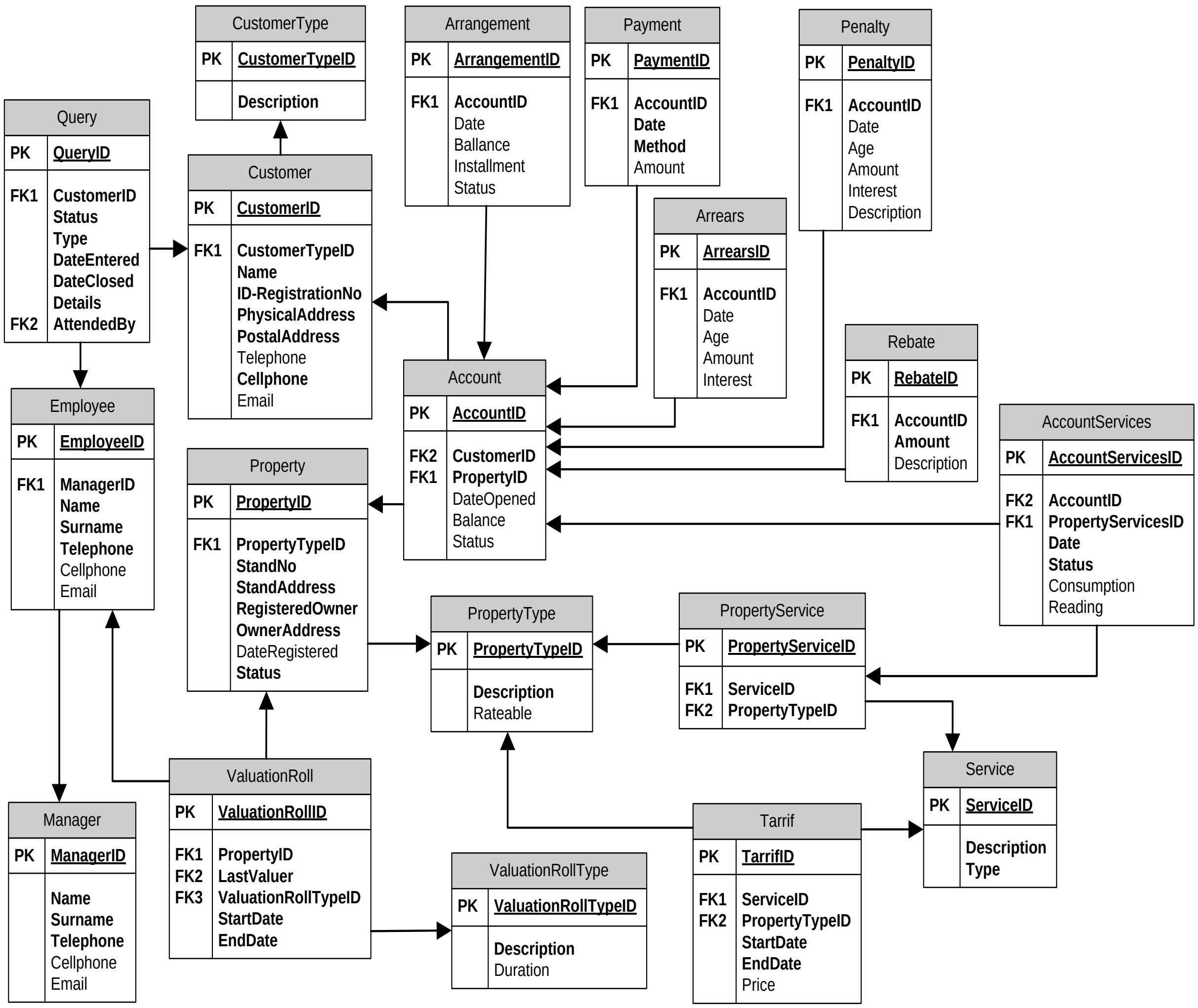

Figure 1. Relational Data Model for South African Municipalities IS for Service Delivery

A relational database can have different types of relationships between its tables. The relationships are maintained by the use of foreign keys. Consider two related tables T1 and T2. The possible relationships between the tables of the relational database are as follows:

- One to One relationship - only one row in T1 corresponds to only one row in T2.
- One to Many relationship - one row in T1 can have many corresponding rows in T2; further, a primary key in T1 will be a foreign key in T2.
- Many to Many relationships - many rows in T1 corresponds to many rows in T2; these relationships are normally resolved by the use of bridge tables.

From the business rules and scenario of the SAMISSD domain, the following entities were initially identified: Query, Administrator, Manager, Property Status, Group, Category, Query Type, Query Status, Property, Customer Group, Customer, Account Status, Account, Account Billing, Property Service, Service, Arrangement, Arrangement Status, Arrears, Payment Method and Tariff. These entities were present in the data collected from all the studied municipalities indicating that these municipalities have compatible entities. Furthermore, it was found that the entities: Services, Property, Property Type/Category, Tariff, and Customer are common to all the municipalities studied. After data comparison and further analysis, the generic relational data model of the SAMISSD domain was drawn as in Figure 1.

Recall that the aim of this study is to provide a case study of the empirical application of semantic web technologies for converting RDB into ontology in IS. The first step was the building of the relational model of the SAMISSD domain in Figure 1. At the conceptual level, the next step is the modelling and translation of the requirements of the domain into a form that can be executed against the semantic model (ontology) that will be constructed from the relational data model later in this study. The requirement modelling is done with competency questions. Competency questions (CQs) are natural language questions representing the requirements of a IS domain; CQs can be translated and executed against an ontology to extract relevant information from it.

Requirement Modelling with Competency Questions

The CQs of the SAMISSD domain are modelled with the Tropos Methodology (Fernandes et al., 2011). It is an ontology building methodology constituted of three phases:

- Early Requirements - during this phase, organisational actors, goals and their dependencies are identified. Organisational actors are like role players in the target knowledge domain. After the actors are identified, their soft and hard organisational goals are identified and modelled together with resources and organisational plans. This is a way of getting full organisational objectives.
- Late Requirements - focuses on the capturing and modelling of CQs from the information obtained in the Early Requirements phase.
- Ontology Modelling – in this phase concepts and their relationships are extracted from the set of CQs to build the domain ontology.

The ontology modelling phase of the Tropos Methodology is not applied in this study. Instead, CQs obtained in the Late Requirements phase are used to query an ontology constructed from the relational database of the SAMISSD domain. The rest of this section applies the first two phases (early and late requirements) of the Tropos Methodology to develop the CQs of the SAMISSD domain.

In the Early Requirements phase, the municipality and customer are identified as the main role players/actors in the SAMISSD domain. In fact, the municipality has a soft goal to provide effective service delivery to customers. The customers are owners of properties that fall under the jurisdiction of the municipality and have the responsibility to log queries whenever they are not satisfied with services offered by the municipality. The soft goal above then leads to three main hard goals of:

- Identifying customers' properties as destinations of services to be rendered,
- Maintaining and managing a roll of all properties under the municipality, and
- Continuously improving services offered to customers.

The three hard goals above are further broken down into four sub-goals: offering services, improving services, capturing properties and managing properties. The resource needed to fulfil the goal of improving services is the queries submitted by customers. By addressing customer queries, the municipality will be in a better position to improve service delivery processes. On the other hand, a resource needed to fulfil the goals of capturing and managing properties is a municipal valuation roll. The valuation roll is a list of all properties under the municipality's jurisdiction.

The scenario above provides an overview of the early requirements where information is collected on how the municipalities fulfil their obligations of effective service delivery to customers.

Identifiers	Competency Questions
CQ1	What are the services offered by the municipality?
CQ2	What are the types of services offered by the municipality?
CQ3	Which services are consumables in the municipality?
CQ4	Which services are basic in the municipality?
CQ5	How many customers do we have in our municipality?
CQ6	What are the names of our customers?
CQ7	What types of customers are catered for in our municipality?
CQ8	What are the overall queries in the municipality?
CQ9	What are the details and status of the current customer queries?
CQ10	What are the types of valuation rolls in the municipality?
CQ11	How much is the highly rated property within the municipality?
CQ12	What is the address of the most valued property in the municipality?
CQ13	How many properties are there in the municipality?
CQ14	How many services are offered for residential properties?
CQ15	What are the ID's of customers who make queries?
CQ16	What are the closed queries from the customers?
CQ17	What are the current open queries from the customers?

Table 2. List of Competency Questions

In the Late requirement phase of the Tropos Methodology, the organisational actors, goals and resources identified above are used to capture and model the competency questions. The resulting CQs of the SAMISSD domain are provided in Table 2. The CQs in Table 2 are encoded with identifiers as in Fernandes et al. (2011). Seventeen CQs were derived in total with identifiers from CQ1 to CQ17 (Table 2). The competency questions CQ1 to CQ4 were derived from the *offering services* goal. In fact, to succeed in offering services to customers, the municipality would be interested in keeping records of service names (CQ1) and their types (CQ2). It will also be necessary for the municipality to specifically know which services are basic (CQ4) and which ones are consumables (CQ3). Consumables are services that are billed according to the customer's usage.

The competency questions CQ5 to CQ7 focus on the *customer* as the second organisational actor and the receiver of services. In this instance, the municipality will be interested to know the number of customers, their names and types to gauge the demand for services. The competency questions CQ8 to CQ9 were derived from the *queries* resource. The municipality would need to know the overall queries and their details in order to achieve the *improve services* goal. The competency question CQ10 was derived mainly from the *valuation roll* resource. Here the municipality would need to establish the types of valuation rolls available to achieve the *capturing properties* and *managing properties* goals. The competency questions CQ11 to CQ13 were derived from the *capturing properties* and *managing* *properties* goals. To achieve these goals, the municipality would have to establish their mostly valued properties (CQ11) and their physical locations (CQ12). The municipality would also be interested to know the number of properties (CQ13) that are in their jurisdiction. The competency questions CQ14 is derived from *offering services, capturing properties and managing properties* goals; in this case the municipality would be interested in identifying specific services that are offered to residential properties. Last, the competency questions CQ15 to CQ17 were derived from customers (organisational actor) and queries (resource). Here, the municipality would be interested in (1) identifying customers (CQ15) who put in queries and (2) the details of closed (CQ16) and open (CQ17) queries from customers. This will assist in the goal to *improving services*.

The CQs in Table 2 must be translated into a format that can be executed against the ontology in an attempt to retrieve useful information from it. This is done in the following section.

Semantic Translation of Competency Questions

The competency questions in Table 2 are in the Natural Language (English) format. They need to be translated into a formal query language in order to be executed against the ontology. In this study, the Competency Question Translation (CQT) approach (Zemmouchi-Ghomari and Ghomari, 2013) is applied to obtain a set of SPARQL queries to be executed against the ontology constructed from the relational database of the SAMISSD domain. SPARQL is a SQL like language for querying ontology.

The CQT approach assumes that the user has a working knowledge of ontology, languages for representing ontology, and ontology query languages such as SPARQL. Ontology is a knowledge base system representing the common and shared vocabularies/concepts within a specific domain as well as the relationships between them (Li et al., 2005; Zhou et al., 2010; Telnarova, 2010). Typical ontology elements are concepts, relationships/properties, axioms and instances (Zhang and Li, 2011; Saleh, 2011). A concept is the basic component of ontology. The relationships/properties between concepts define how concepts are semantically related to each other in the ontology. Axioms are the statements in the ontology, i.e., the logical combinations of concepts and properties. The instances are the occurrences/values of concepts or properties in the ontology. The popular languages for the formal representation of ontology are Resource Description Framework (RDF) and Web Ontology Language (OWL). However, OWL is preferred over RDF due to the weak expressive power of the RDF language (Li et al., 2005; Jia and Yue, 2009). It is also considered to be the most advanced ontology representation language (Lemaignan et al., 2006).

The CQT approach starts with the classification of CQs into different categories according to expected answers' types. There are five types of questions including:

- Definition questions ("What is/are?" or "What does this mean?" type of questions),
- Boolean questions (question with Yes/No answers),
- Factual questions (those that search precise information),
- List questions (those that query list of entities), and
- Complex Questions ("How" and "Why" type of questions).

After the questions are categorised, their expected answers are then determined. From the answers and questions, entities are extracted and their types (whether it's a concept/class, relation/property, axiom, or instance) are identified. With this information, appropriate SPARQL queries are constructed. This process was applied on the set of CQs in Table 2 to build SPARQL queries for each. Table 3 shows a sample SPARQL queries for the competency questions CQ1, CQ9, CQ11 and CQ12 (Table 2).

Identifiers	SPARQL Queries
CQ1	SELECT? Name WHERE { ?service a Ont:Name. ?service Ont:Service.Name ?Name. }
CQ9	SELECT * WHERE { ?query a Ont:Query. ?query Ont:Query.Details ?details. ?query Ont:Query.Status ?status.}
CQ11	SELECT (MAX (?value) AS ?value) WHERE { ?prop a Ont:Property. ?prop Ont:Property.Value ?value}
CQ12	SELECT ?address WHERE { ?prop a Ont:Property. ?prop Ont:Property.Address ?address. ?prop Ont:Property.Value ?value FILTER (?value = Max(value)) }

Table 3. Sample SPARQL Queries

In Table 3, the SPARQL query for CQ1 selects (SELECTS?) all the names from the Service class (Ont:Service) in the ontology (Ont). This will result in the list of service names from the ontology. Similarly, the SPARQL query for CQ9 returns all information (SELECT*), i.e., details and status, on the query class (Ont:Query) in the ontology (Ont). With regard to CQ11, its SPARQL query selects the maximum value attribute (SELECT (MAX (?value) AS ?value) from the Property class (Ont:Property) in the ontology (Ont). The results of this query will indicate the highly rated property within the municipality. Last, the SPARQL query for CQ12 selects an address from the Property class (Ont:Property) in the ontology (Ont). The query also has a filter represented by the *FILTER* command within the *WHERE* clause. This is to filter the maximum value of a property. The whole query will result in a physical address of a highly rated property in the municipality.

Up to this point, a relational data model for the SAMISSD domain was built (Figure 1) and the business requirements of the domain modelled and translated into SPARQL semantic web query language (see sample SPARQL queries in Table 3). To achieve the aims of the study, that is, to empirically assess the effectiveness of Semantic Web technologies in transforming and extracting useful information from a relational data model of the SAMISSD domain, experiments are going to be carried out in the next section to: (1) create and populate a relational database of the SAMISSD domain based on the relational data model in Figure 1, (2) apply relevant Semantic Web technologies to automatically built ontology from the relational database, (3) run the SPARQL translations of the CQs in Table 2 on the resulting ontology in an attempt to retrieve information from it, and (5) collect views of potential end users of applications in the SAMISSD domain to analyze the outputs of SPARQL queries on the resulting ontology.

EXPERIMENTS

Computer and Software Environment

Experiments were carried out on a Notebook computer with 2 GB of random access memory (RAM) and a Windows 7 Operating System. Oracle 11g Express Edition was used as Relational Database

Management System (RDBMS). The Protégé ontology editor version 4.3 and an associated plug-in, namely, DataMaster (Nyulas et al., 2007) were used to automatically construct ontology from the Oracle database. A plug-in is a separately developed software module that adds more functionality to existing software. Protégé ontology editor and its plug-ins are available for download free of charge over the Internet. The graphical representation of the output ontology from DataMaster was done using virtualisation plug-ins including OntoGraf (Falconer, 2010) and OWLViz (Horridge, 2010). The execution of the SPARQL queries against the resulting ontology was done with the Protégé query interface.

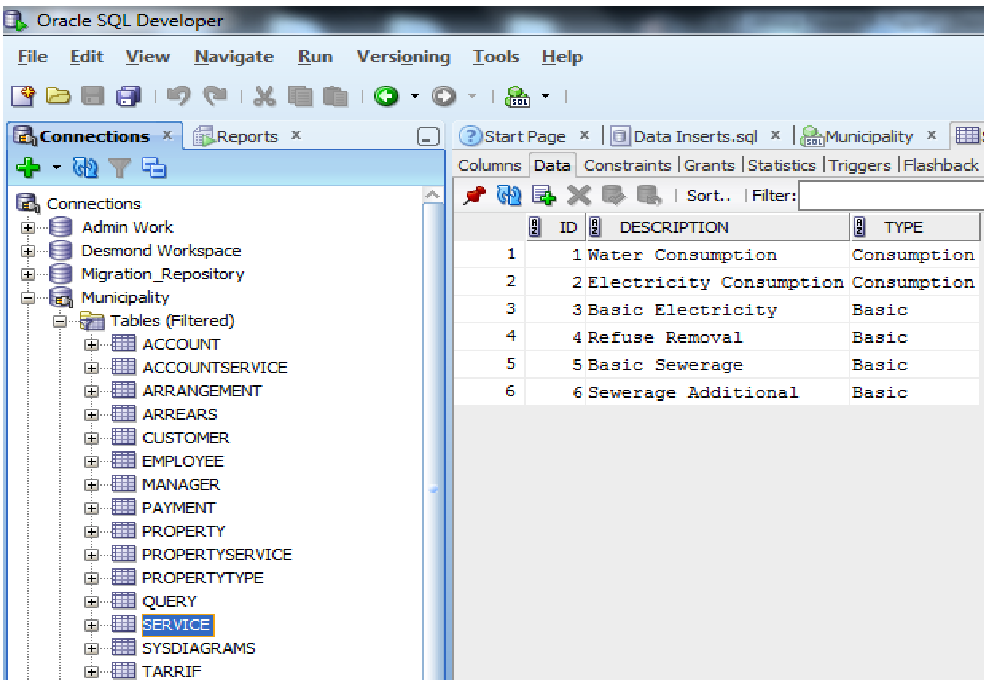

Figure 2. Screenshot of a Municipality Database in Oracle

Ontology Construction from the Oracle Database

The relational data model in Figure 1 was further implemented into a Municipality database using Oracle 11g Express Edition. Figure 2 shows a screenshot of a database in Oracle. From the screenshot, it can be seen that the database has already been populated with sample data. The left panel of Figure 2 shows a part of the database tables, whereas, the right panel depicts some records of the Service table. The relational database in Figure 2 will be converted to an ontology to allow the semantic exploitation of its records. To this end, the DataMaster plug-in (Nyulas et al., 2007) within Protégé was used to automatically construct the ontology from the Oracle database.

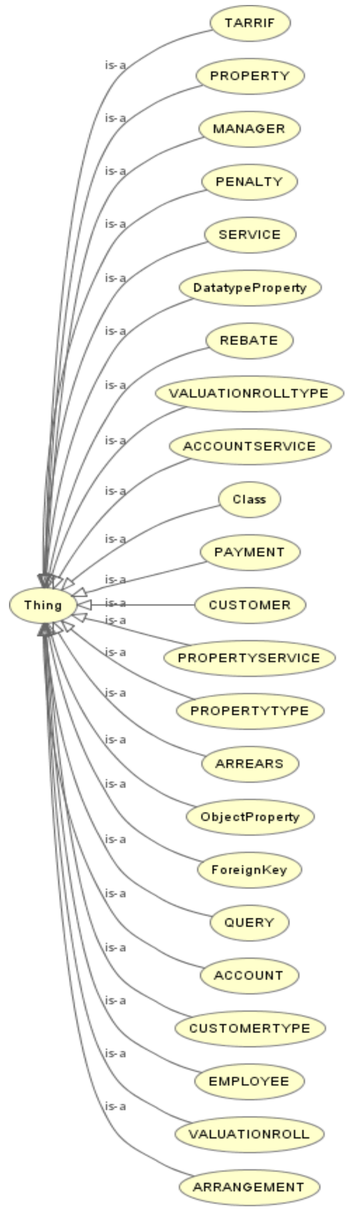

Figure 3. Inheritance Structure of Ontology Constructed with DataMaster Plug-in

Recall that the ontology editor like Protégé creates ontology in one of the semantic web ontology languages including RDF and OWL. The DataMaster Protégé plug-in constructs ontology from a relational database in OWL language. Figure 3 shows the classes of the OWL ontology constructed from the Oracle database in Figure 2 with the DataMaster plug-in. The graphical representation of classes in Figure 3 was obtained with the OWLViz virtualisation plug-in installed within Protégé. The meaning of the graph in Figure 3 is that all classes produced, inherit the default OWL class Thing; this is represented by the phrase *is-a* on the edges of the graph. It can be noticed that the classes of the ontology in Figure 3 includes all the tables of the database in Figure 2. The complete graph of the resulting ontology is shown in Figure 4; this graph was generated with the OntoGraf (Falconer, 2010) virtualisation plug-in. Figure 4 shows all the classes of the resulting ontology and the relationships between them.

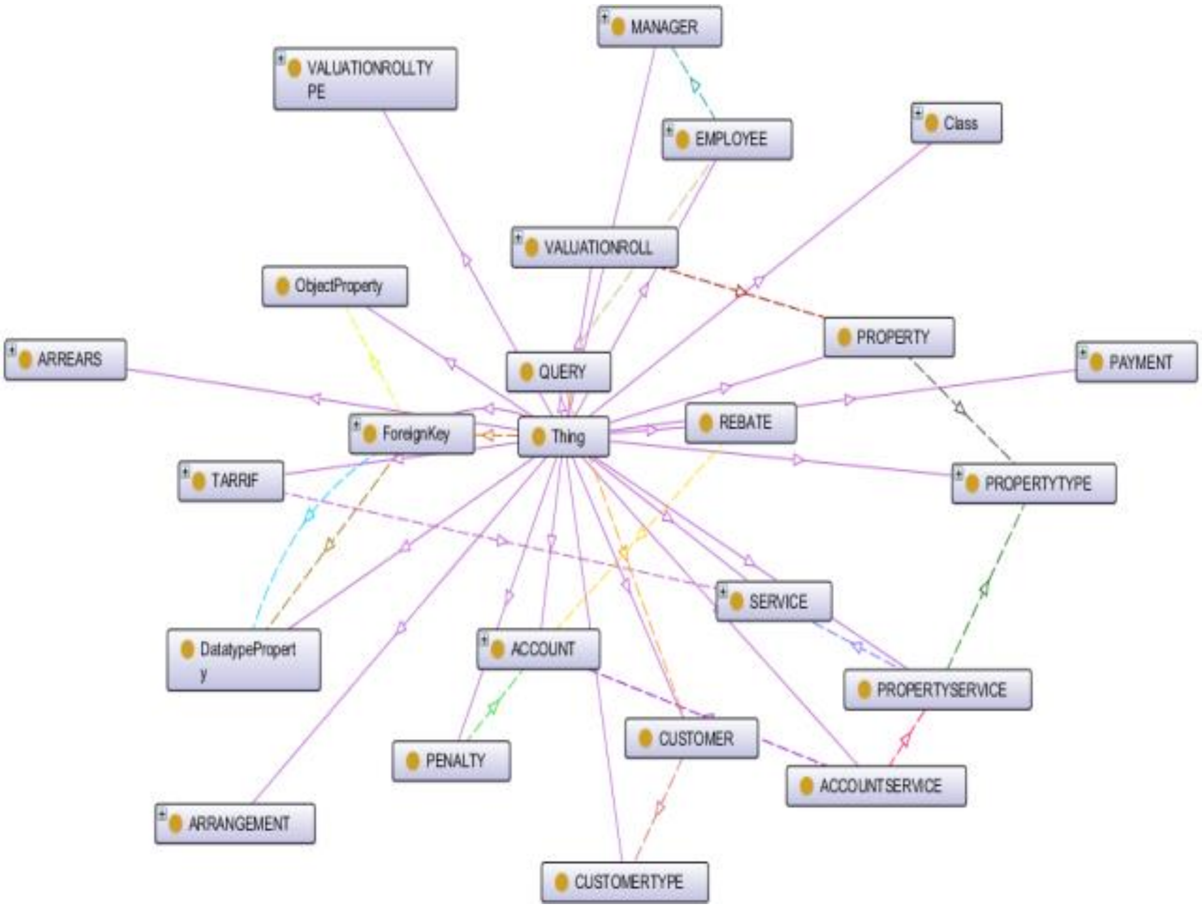

Figure 4. Ontology Constructed with DataMaster Plug-in

The Oracle database included 19 tables (see Figure 1 for all the tables). The resulting ontology (Figures 3 and 4) have 23 classes; these classes include all the tables of the database with four additional tables. This indicates that the DataMaster plug-in has accurately converted all the tables of the database into ontology classes. This observation is in line with the conceptual mapping rule of database and ontology structures proposed by Zhou et al. (2010), Zhang and Li. (2011), Gherabi et al. (2012) and Sedighi and Javidan (2012). This stipulates that all tables in the database should be mapped to classes with the same names in the ontology. A similar analysis can be done on the relationships in both the database and resulting ontology. In fact, in the relational data model in Figure 1, there are one-to-many relationships between the pair of tables (Property, PropertyType) and (Customer, CustomerType); these relationships have been successfully represented in the resulting ontology in Figure 4 by the creation of edges between the corresponding pair of classes (Property, PropertyType) and (Customer, CustomerType). Another example is the many-to-many relationship between the tables PropertyType and Service which is represented by the bridge table PropertyService in the relational data model in Figure 1; this relationship has been successfully represented in the resulting ontology in Figure 4 with the creation of edges between the pair of classes (PropertyService, Service) and (PropertyService, PropertyType). This is an indication that the DataMaster plug-in has accurately mapped the relationship between the tables of the database into semantic relationships between the classes of the resulting ontology. The results in Figures 3 and 4 provides proof that Protégé ontology editor associated with DataMaster plug-in can accurately transform a relational database into ontology. The next step towards the achievement of the aims of the study is to ascertain if the resulting ontology (Figures 3 and 4) satisfies the requirements of the SAMISSD domain. To this end, the SPARQL translations of the CQs in Table 2 was run against the OWL code of the resulting ontology in the next Subsection. Further, the outputs of the SPARQL queries will be analyzed based on end users' views of answers to the CQs in Table 2.

Semantic Query Execution and Analysis

Finally, SPARQL queries are run on the OWL code of the resulting ontology. Figure 5 shows a sample SPARQL query (CQ1 in Table 2) execution and outputs. The bottom part of Figure 5 shows the list of all the service instances in the ontology. These are the services offered by the municipality to its customers. This indicates that SPARQL semantic web queries can successfully extract useful information from a relational database (Figure 2) via an ontology (Figures 3 and 4).

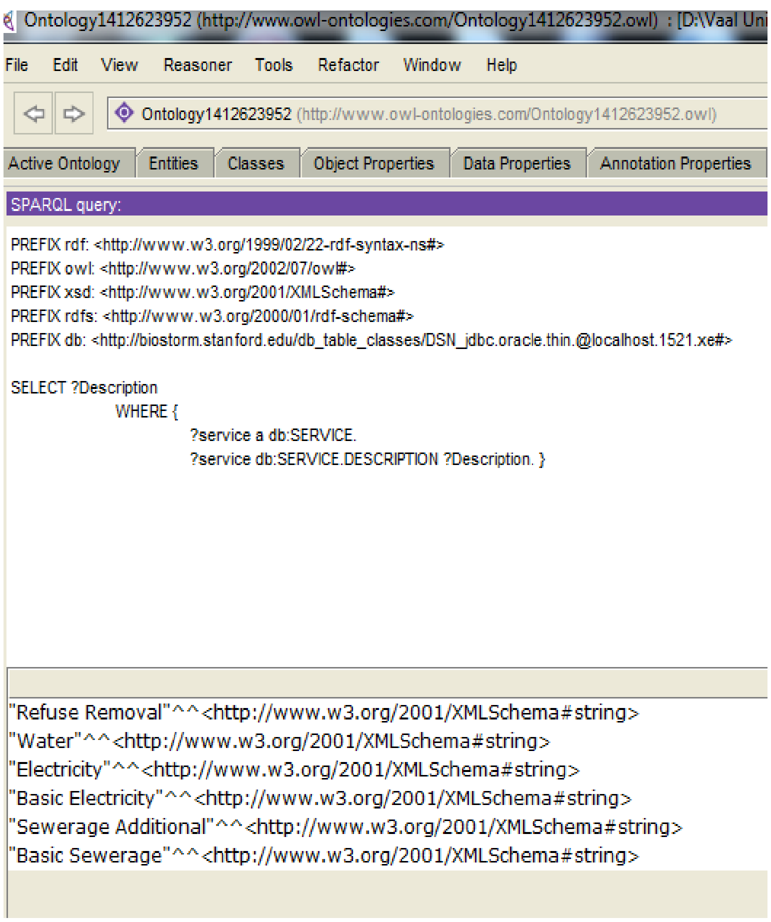

Figure 5. Sample SPARQL Query and Outputs

The outputs of SPARQL queries (see example in Figure 5) must be further analyzed against user views of the answers to the CQs. This analysis aims to ascertain whether the resulting ontology satisfies the business requirements of the domain. To this end, a technique presented in (Bezerra et al., 2013) is used in this study. In fact, Bezerra et al. (2013) proposed an algorithm which split a CQ into several tokens. Thereafter, the tokens are used to retrieve concepts or instances from the ontology. The resulting concepts and instances constitute the answer to the CQ. This approach is adopted in this study to analyze the SPARQL queries outputs against users' views of answers to CQs.

To differentiate the authors from the end users of the ontology, three individuals were invited to participate in the study. The three participants were all owners of properties in South Africa and had good knowledge of the business domain. Furthermore, the participants hold Information Technology (IT) qualifications (Master of Technology in IT, Bachelor of Technology in IT and Bachelor of Technology in Computer Systems, respectively) and had a good understanding of concepts of entity, class, term, instance, occurrence, etc. However, they had little knowledge of ontology.

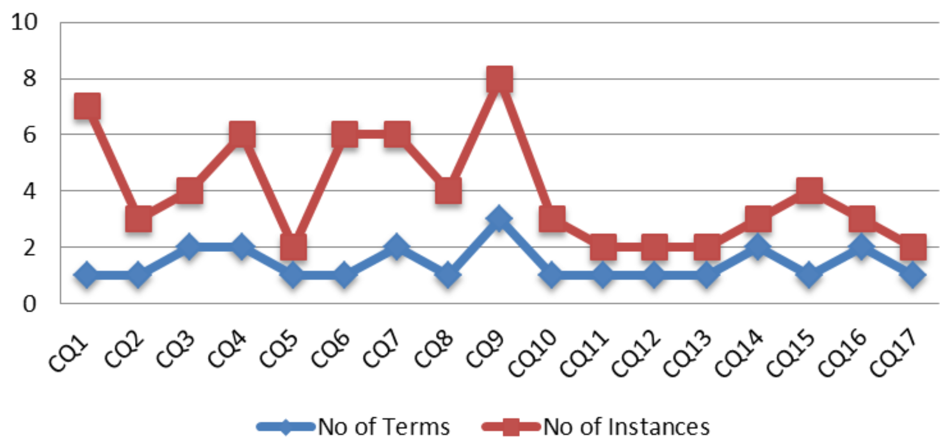

Figure 6. Chart of the Mapping of Terms of Participant 1 to SPARQL Outputs Instances

The three participants were given the list of competency questions in Table 2 and requested to select in each question the terms/entities/classes that they think could be or that the instances could be the answer to the question. The lists of CQs were collected and the selected terms/entities/classes analysed against the SPARQL queries outputs as in Figures 6, 7 and 8.

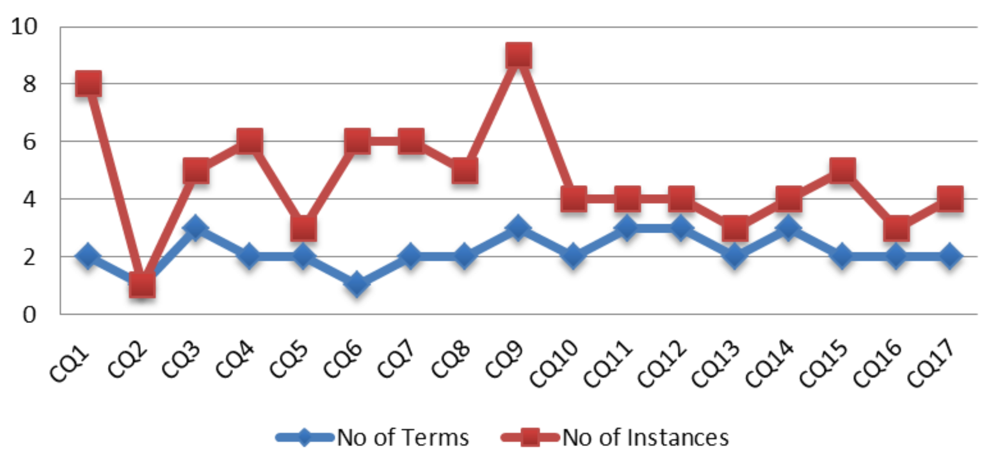

Figure 7. Chart of the Mapping of Terms of Participant 2 to SPARQL Outputs Instances

Figure 6 shows the results of the mapping of the CQs terms selected by the first participant to the outputs of the SPARQL queries. The chart in Figure 6 suggests that one term was selected per CQ by the participant for 11 CQs and two terms for five CQs (CQ3, CQ4, CQ7, CQ14 and CQ16). Further, it showed that all the selected terms had at least one instance in the SPARQL output results; in particular, the CQ9 has many instances in the output results.

Similarly, Figure 7 shows the results of the mapping of the CQs terms selected by the second participant to the outputs of the SPARQL queries. From Figure 7, it is observed that the participant selected more terms per CQ; one term was selected for the CQs CQ2 and CQ6, two terms for CQs CQ1, CQ4, CQ5, CQ7, CQ8, CQ10, CQ13, CQ15, CQ16 and CQ17, and three terms for the CQs CQ3, CQ9, CQ11, CQ12 and CQ14. Further, Figure 7 shows that all the selected terms had at least one instance in the SPARQL queries outputs, with CQ1, CQ4, CQ6, CQ7 and CQ9 having more instances in the outputs results.

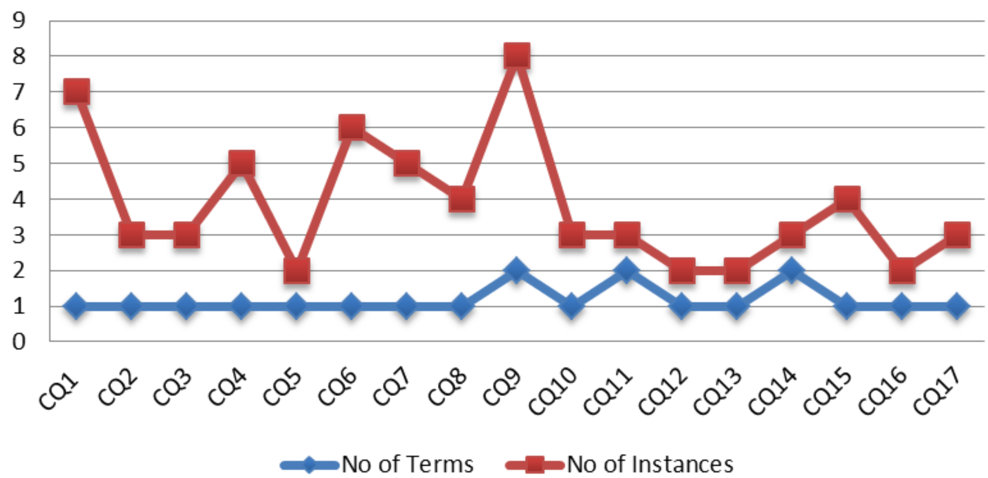

Figure 8. Chart of the Mapping of Terms of Participant 3 to SPARQL Outputs Instances

Last, the mapping of the terms chosen by the third participant to the SPARQL queries outputs is presented in Figure 8. It showed that the participant selected one term in most of the CQs except for CQ9, CQ11, and CQ14. Figure 8 also depicts that SPARQL output results included at least one instance of each selected term by the third participant with more instances for the CQs CQ1, CQ6 and CQ9.

In light of the above, user participants had different views of the answers to CQs. However, the execution of the SPARQL translations of CQs against the ontology produced outputs instances that satisfy users' expectations. This indicates that the resulting ontology constructed from the RDB of the SAMISSD domain with the DataMaster plug-in embodies domain and semantic features that appropriately capture the requirements of the SAMISSD domain.

CONCLUSION AND FUTURE WORK

This study developed and semantically exploited a relational data model of the SAMISSD domain. The business requirements of the SAMISSD domain were gathered and analyzed to build the relational data model. The relational data model was further developed as a test RDB in Oracle RDBMS. Semantic web tools including Protégé ontology editor and associated plug-ins were applied to convert the test RDB of the SAMISSD domain into ontology. The analysis of the structure of the resulting ontology showed some elements of accuracy in the conversion from RDB to ontology under Protégé. To exploit the records of the test RDB via the resulting ontology, CQs were used to represent the requirements of the

SAMISSD domain. The resulting CQs in natural language form were further translated into SPARQL semantic web query language for their execution against the ontology. The execution of the SPARQL translations of CQs showed that relevant information could effectively be extracted from a RDB via an ontology counterpart automatically constructed from it. User participants were further involved in the study to assess the potential of the resulting ontology to accurately answer the CQs. The analysis of SPARQL queries outputs showed that, although users have different views of the answers to CQs, the execution of the SPARQL translations of CQs against the ontology produced output instances that satisfied users' expectations. This provided indications that RDB can be effectively exploited on the Semantic web via ontology and semantic queries.

The future direction of the research would be to investigate the concept of semantic web services to build a full semantic web application of the SAMISSD domain that interfaces the ontology constructed from the RDB and provide appropriate semantic-driven functionalities for answering the CQs.

REFERENCES

Alatrish, E.S. (2013) Comparison of Some Ontology Editors, *Management Information Systems*, 8, 2, 18-24.

Banu, A., Fatima, S.S. and Khan, K.U.R. (2011) Semantic – Based Querying Using Ontology in Relational Database of Library Management System, *International Journal of Web and Semantic Technology (IJWesT)*, 2, 4, 21-32.

Bezerra, C., Freitas, F. and Santana, F. (2013) Evaluating Ontologies with Competency Questions, *Proceedings of the 2013 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology*, Atlanta, GA, USA, 17-20 November, 284-285.

Buitelaar, P., Olejnik, D. and Sintek, M. (2004) A Protégé Plug-in for Ontology Extraction from Text Based on Linguistic Analysis, *Proceedings of the 1st European Semantic Web Symposium (ESWS 2004)*, Heraklion, Greece, 10-12 May, 31-44.

ELM, 2014a Emfuleni Local Municipality Tariff Policy 2013/2014 Financial Year; retrieved November 15, 2015 from http://www.emfuleni.gov.za/images/docs/policies/tariff_policy201314.pdf

ELM, 2014b Emfuleni Local Municipality Property Rates Policy 2013/2014 Financial Year; retrieved November 15, 2015 from http://www.emfuleni.gov.za/images/docs/policies/property_rates_policy201314.pdf

Falconer, S. 2010 OntoGraf; retrieved November 15, 2015 from

http://protegewiki.stanford.edu/wiki/ OntoGraf

Fatima, S. and Rajput, Q. (2012) A Comparison of Algorithms to Construct Ontologies from Relational Databases, *Proceedings of the Seventh International Multi-Conference on Computing in the Global Information Technology*, Venice, Italy, 24-29 June, 232-237.

Fernandes, B.C.B., Guizzardi, R.S.S. and Guizzardi, G. (2011) Using Goal Modelling to Capture Competency Questions in Ontology-based Systems, *Journal of Information and Data Management*, 2, 3, 527-540.

Gennari, J., Nguyen, M. and Silberfein, A. (2007) DataGenie; retrieved November 15, 2015 from http://protege.cim3.net/cgi-bin/wiki.pl?DataGenie

Gherabi, N., Addakiri, K. and Bahaj, M. (2012) Mapping Relational Database into OWL Structure with Data Semantic Preservation, *International Journal of Computer Science and Information Security*, 10, 1, 42-47.

Gruber, T.R. (1993) ATtranslation Approach to Portable Ontology Specification", *Knowledge Acquisition*, 5, 2, 199-220.

Horridge, M. 2010 OWLViz; retrieved November 15, 2015 from http://protegewiki.stanford.edu/wiki/OWLViz

Jain, V. and Singh, M. (2013) A framework to Convert Relational Database to Ontology For Knowledge Database in Semantic Web, *International Journal of Scientific & Technology Research*, 2,10, 9-12.

Jia, C. and Yue, W. (2009) Rules-based Object-relational Databases Ontology Construction, *Journal of Systems Engineering and Electronics*, 20, 1, 211-215.

Koma, S.B. (2010) The state of Local Government in South Africa: Issues, trends and options, *Journal of Public Administration*, 45, 1.1, 111-120.

Laclavik, M. (2006) RDB2Onto: Relational Database Data to Ontology Individuals Mapping, *Tools for Acquisition, Organisation and Presenting of Information and Knowledge*, Navrat et al. (Eds), 86-89.

Li, M., Du, X. and Wang, S. (2005) Learning Ontology from Relational Database, *Proceedings of the Fourth International Conference on Machine Learning and Cybernetics*, Guangzhou, China, 18-21 August, 3410-3415.

Mahmood, N., Burney, A. and Ahsan, K. (2010) A logical Temporal Relational Model, *International Journal of Computer Science Issues*, 7, 1, 1-9.

Mulligann, C., Trame, J. and Krzysztof, J. (2011) Introducing the new SIM-DLA Semantic Similarity Measurement Plug-in for the Protégé Ontology Editor, *Proceedings of the 1st ACM SIGSPATIAL International Workshop on Spatial Semantics and Ontologies*, Chicago, IL, USA, 1-4 November, 17-24.

Navathe, S.B. (1992) Evolution of Data Modelling for Databases, *Communications of the ACM*, 35, 9, 112-123.

Nyulas, C., O'Connor, M. and Tu, S. (2007) DataMaster – a Plug-in for Importing Schemas and Data from Relational Databases into Protégé, *Proceedings of the 10th International Protégé Conference*, Budapest, Hungary, 15-18 July.

Papapanagiotou, P., Katsiouli, P., Tsetsos, V., Anagnostopoulos, C. and Hadjiefthymiades, S. (2006) RONTO: Relational to Ontology Schema Matching, *AIS SIGSEMIS Bulletin*, 3, 3, 32-36.

Pasha, M. and Sattar, A. (2012) Building Domain Ontologies From Relational Database Using Mapping Rules, *International Journal of Intelligent Engineering & Systems*, 5, 1, 20-27.

Ramathilagam, C. and Valarmathi, M.M. (2014) A Framework for Direct and Semantic Based Transformation from Relational Database to Ontology, *Journal of Theoretical and Applied Information Technology*, 63, 3, 814824.

RSA, 1996 Constitution of the Republic of South Africa; retrieved November 15, 2015 from http://www.thehda.co.za/uploads/images/unpan005172.pdf

RSA, 2000 Local Government Municipal Systems Act 32 of 2000; retrieved November 15, 2015 from http://www.saflii.org.za/za/legis/num_act/lgmsa2000384.pdf

Sedighi, S.M. and Javidan, R. (2012) Semantic query in a relational database using local ontology construction, *South African Journal of Science*, 108, 11/12, 1-10.

Saleh, M.E. (2011) Semantic-Based Query in Relational Database using Ontology, *Canadian Journal on Data, Information and Knowledge Engineering*, 2, 1, 1-16.

Sequeda, J.F., Marcelo, A. and Miranker, D.P. (2012) On Directly Mapping Rational Databases to RDF and OWL, *Proceedings of WWW 2012 – Session: Ontology Representation and Querying: RDF and SPARQL*, Lyon, France, 16-20 April, 649-658.

Telnarova, Z. (2010) Relational Database as a Source of Ontology Creation, *Proceedings of the International Multi-conference on Computer Science and Information Technology*, Wisla, Poland, 18-20 October, 135-139.

Tirmizi, S.H., Sequeda, J. and Miranker, D. (2008) Translating SQL Applications to the Semantic Web, *Proceedings of the 19th International Conference Database and Expert Systems (DEXA 2008)*, Turin, Italy, 1-5 September, 450-464.

Trinkunas, J. and Vasilecas, O. (2007) Building Ontology from Relational Database Using Reverse Engineering Methods, *Proceedings of the International Conference on Computer Systems and Technology (ComSysTech'07)*, Rousse, Bulgaria, 14 – 15 June, 61-66.

Zemmouchi-Ghomari, L. and Ghomari, A.R. (2013) Translating Natural Language Competency Questions into SPARQL Queries: a Case Study, *Proceedings of the First International Conference on Building and Exploring Web Based Environments*, Seville, Spain, 27 January – 1 February, 81-86.

Zhang, L. and Li, K. (2011) Automatic Generation of Ontology Based on Database, *Journal of Computational Information Systems*, 7, 4, 1148-1154.

Zhou, S., Ling, H., Han, M., and Zhang, H. (2010) Ontology Generator from Rational Database on Jena, *Computer and Information Science Technology*, 3, 2, 263-267.