Semantic Invoice Processing

Luis M. Escobar-Vega [a,*], Víctor H. Zaldívar-Carrillo [b] and Ivan Villalon-Turrubiates [b]

[a] *Doctoral Program in Engineering Sciences, ITESO Universidad Jesuita de Guadalajara, Periférico Sur Manuel Gómez Morín # 8585, Tlaquepaque, Mexico 45604*
E-mail: ng700765@iteso.mx.

[b] *Department of Electronics, Systems and Informatics, ITESO Universidad Jesuita de Guadalajara, Periférico Sur Manuel Gómez Morín # 8585, Tlaquepaque, Mexico 45604*
E-mail: {victorhugo}{villalon}@iteso.mx

Abstract. This work highlights how to transform information from invoice documents to semantic models, as an implementation of ontology modeling. The migration from printed paper to digital documents in the Mexican Government Offices in the last few years has brought significant opportunities for the usage of information technologies and applications. However, when changing digital document information into knowledge, there are still many gaps to be filled. This work proposes a solution to some issues regarding ontology modeling, specifically when mapping a document that follows some XML schema to an ontology under the OWL standard. The main contribution of this work is to provide new interpretations of the XML terms in the context of OWL, so that the XML Schema Definition (XSD) structures can be mapped into more complex OWL structures. A software tool developed to test and validate the information extraction strategies proposed is presented here.

Keywords: organizational knowledge, knowledge management, semantic technology, semantic web, information extraction, ontology modelling

1. Introduction

The latest changes in the tax management platform in Mexico has opened some opportunities to overcome the technological lag in this country. The amount of semantic information made available by the Mexican Government for open use [1] will allow the introduction of semantic tools that have already been implemented in other countries [2], [3].

The *Internet Digital Fiscal Receipt* (CFDI from its name in Spanish) is the current model of electronic invoice valid in Mexico since January 2011. This type of receipt, that uses standards regulated by the government fiscal agency in Mexico, is constituted as a digital document in XML [4] that has the following characteristics:

1. Integrity: the information contained in a CFDI cannot be manipulated nor modified without being detected.
2. Authenticity: the identity of the generator of the receipt can be verified through its Digital Certified Seal.
3. Unique: each and every CFDI has attached a registered identifier given by an Approved Certification Supplier that transforms the receipt into the link between its addressee and the government.
4. Verifiable: the person emitting the CFDI could not deny having emitted it. CFDI is obligatory to be used in every commercial or business operation.

The CFDI brings opportunities to the companies that develop commercial applications by facilitating, interacting, and accessing semi-structured information, which makes it possible to develop systems

[*]The authors would like to thank the Instituto Tecnológico y de Estudios Superiores de Occidente (ITESO) of Mexico for the resources provided for this research. Also, the main author would like to thank the National Council of Science and Technology (CONACYT) of Mexico for the sponsoring of this research by the scholarship number 399053.

to manage the commercial knowledge, information search engines, electronic commerce platforms, information management agents, and knowledge managers, to mention but a few. The bottom line will be in favor of the final users by enabling them to get better understanding about their businesses. However, to reach this aim, it is required to transform the information into knowledge. In the late 90s, Tim Berners-Lee was concerned about the fact that information itself in the web was not enough to make the computers understand the knowledge that was being generated. Even though the HTML documents can be linked through hyperlinks, they are isolated documents, which makes it complex to share information [5]. This may be a gap that causes serious problems in accessing and processing the available information; especially in searching for information, presenting information, and electronic commerce [6].

One might think that the migration from CFDI to semantic documents could facilitate the creation of complex tools that lead to a deeper understanding of the information as knowledge, and the regular user (without technical skills) would have a tool very simple to use. However, there are some gaps to be solved before using these semantic documents. It is necessary to model the knowledge in a proper way. The CFDI are structures that are nested, but, additional information (implied) is required to successfully transform them into a semantic model.

In this work, the information is differentiated according to its abstraction degree. That is, there are some words of popular use that are commonly found in dictionaries, encyclopedias, and that are taken as concise concepts or ideas; on the other hand, there are some other words such as names of persons, products, streets, among others that are considered as assertions of the concepts. The latter are also known as individuals. The CFDI, as a semi-structured document, is composed of both types of information. In Description Logic, this is defined as TBox (terminological) for naming the concepts and ABox (assertional) for naming the instances. For example,

$$Man \equiv Person \sqcap Male$$

where a male can be defined as a male person by writing this declaration. Similarly,

$$Male \sqcap Person(PABLO)$$

```
<?xml version="1.0" encoding="utf-8"?>
<cfdi:Comprobante
        xsi:schemaLocation=
"http://www.sat.gob.mx/cfd/3
http://www.sat.gob.mx/./cfdv32.xsd" folio="20474"
xmlns:cfdi="http://www.sat.gob.mx/cfd/3"
xmlns:xsi="http://www.w3.org/2001/XMLSchema...">
 <cfdi:Emisor>
   ...
 </cfdi:Emisor>
 <cfdi:Receptor>
   ...
 </cfdi:Receptor>

 <cfdi:Conceptos>
  <cfdi:Concepto unidad="CAPSULAS" importe="244"
   cantidad="1.0" descripcion="VIBRAMICINA 100MG"
         valorUnitario="244.00" />
  <cfdi:Concepto unidad="BOTELLA"
   importe="137.93" cantidad="1.0"
        descripcion="CLORUTO 500M"
        valorUnitario="137.93" />
  <cfdi:Concepto unidad="TABLETAS" importe="84.5"
  cantidad="1.0" descripcion="SEDEPRON 250MG 10"
        valorUnitario="84.50" />
 </cfdi:Conceptos>
</cfdi:Comprobante>
```

Fig. 1. CFDI document: electronic invoice with levels of nesting.

states that the individual PABLO is a male person. Given the above definition of man, one can derive from this assertion that PABLO is an instance of the concept Man. [7].

In the CFDI, it is no frequent that the TBoxes change, unless there is a new rule that demands to modify the current elements (this just happens when the congressmen reach consensus on the need of new policies on the fiscal processes). On the other hand, the ABoxes suffer alterations that are not necessarily related to the TBoxes, e.g. An invoice has descriptions of products, always using natural language; e.g. “Aspirin tablets 10 mg. for Infants”, if this text is read, It can be observed that the description is about some medicine, the amount of active formula, and the recommended user (for children). Modeling of these types of chains is what makes the ontology modeling complex, especially when the modeling pattern is being created dynamically. Due to the complexity of analysis, this document excludes the transformations of this kind of information and just takes them as informative chains.

Here, there are presented some of the strategies that are taken to transform the information of the CFDI documents, describing the considerations and the different components used to implement and reach a solution. The final aim is to build a software tool that can extract and transform the information of several CFDI into an ontology that follows the OWL standard. This ontology will be used as the base for a QAS that uses the knowledge represented to provide some decision maker with insights about its business in a more user-friendly way.

2. CFDI documents in XML

The XML used to describe the CFDI is composed of information about the purchase done by a client. The data is clustered around the basic concepts of the sale e.g. The “Client” element contains information from the customer, such as the address. The “Provider” is referred to the one who closes the sales (or offers the service), and the “Concept” element is a list where a complete description of the products/services of the sale is included. An example of a CFDI document is shown in figure 1. The challenge comes because the structure of XML schema often contains implicit assumptions about taxonomy and relationships. However, for the intended meaning (i.e. the semantics) to work, it is required that the resources be explicit in a manner that are understood by computers [7]. This makes it difficult to implement a completely general and automated mechanism to transform XML schemas into OWL ontologies.

XML provides a language for describing the structure of information to support automated processing [8]. The XML Schema Definition (XSD) language contains type and element definitions that describe characteristics of well-formed elements and attributes of XML documents [9]. However, the XSD language does not express semantics [10] and so it creates difficulties for semantic web technologies. The RDF standard [11] was created to represent information structures. Its syntax is defined by data structures that represent graphs that are sets of subject-predicate-object triplets, where the elements may be URIs, blank nodes, or data typed literals. The main objective of the RDF is to express descriptions of resources [6].

On the other hand, the OWL [12] specification was developed to enable semantics. It extends RDF with terminology to express ontologies using description logic (DL), a decidable fragment of first order logic [7]. OWL DL ontologies are a subset of OWL that satisfies the expressive requirements of a description logic.

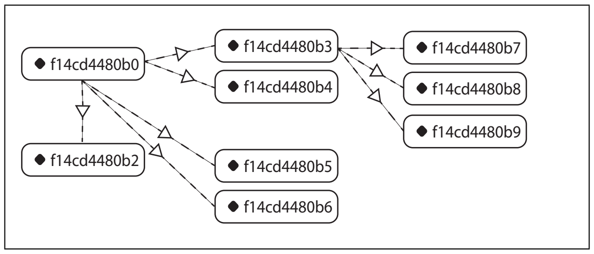


Fig. 2. XML nodes linked to OWL elements: individual relation results (*protégé*).

3. Transforming a CFDI to an OWL

The invoice results from a purchase made by a client. The CFDI represents this invoice and it is composed of an issuer and an addressee, the description of the sold items, and a description of the taxes applied in the purchase. The CFDI, just as all the XML, is handled by an XSD schema, where it is defined which fields, lengths, and types of fields are required. The element “Concept” is the root and contains the other elements; it might be nested depending on other concepts.

Just like the CFDI, the ontology is described in an XML format; this facilitates the transformation of the documents into ontologies. Both use schemes and name spaces. The CFDI uses the schemas from SHCP (*Secretaría de Hacienda y Crédito Público, Mexican Government*), meanwhile the OWL uses those from W3C/Semantics. However, this may have some complications while analyzing the information in a semantic form, e.g. a logic reasoner could not use a CFDI directly, since in the first instance, it needs that the information contained in the document be explicit, both concepts and relationships. Figure 2 shows what is expected: every XML node is linked with an OWL element and SPARQL query is used in a simple way to test it.

3.1. Mapping Strategy

There has been some approaches to transform XML into OWL/RDF, most of them are based on linear mappings. Redefer [13] is a framework that demonstrates how to map transformations, and it is able to work in environments such as Digital Right Management (DRM); on the other hand, Ontomalizer [14] is a tool to improve metadata enrichment and semantic processing for biomedical documents. There are other frameworks from open domain that also can be used in mapping transformations: OpenRefine [15] is a tool to transform data from one format into another, includ-

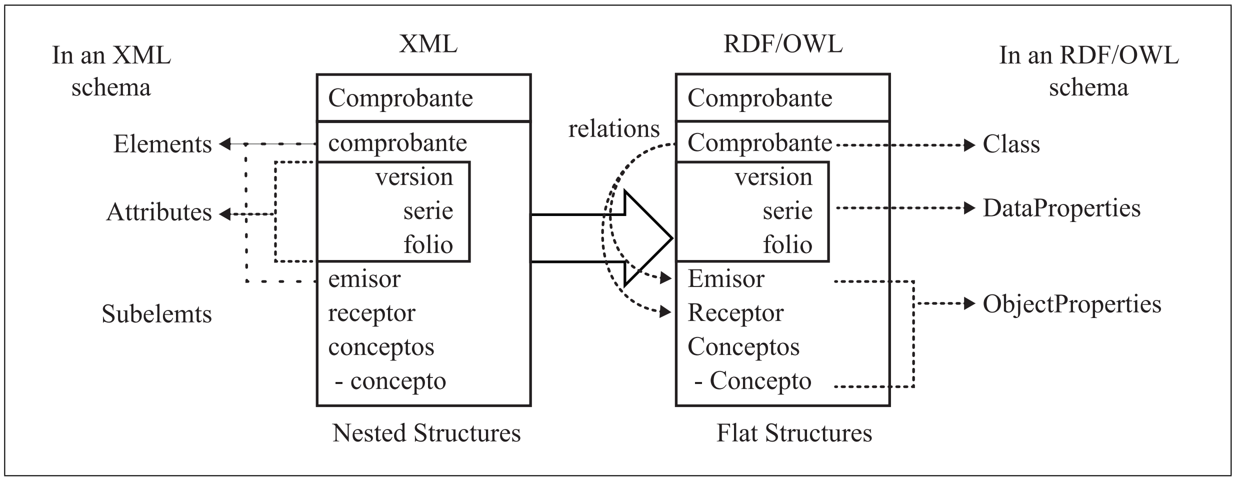


Fig. 3. Semantic transformation: XML is parsing to RDF structure. The names of the concepts are written in Spanish just as they come in the original XSD schema

ing XML or RDF, and TopBraid [16] that recently included the ability to convert XSDs and associate XML files to RDF/OWL.

Ferdinand's work [17] was one of the first approaches that proposed the use of conversion rules to deal with the mapping of the concepts in the CFDI documents. Its use could be applied to just one document as well as to groups of XML documents creating a more complex ontology. The proposal described in this work considers some topics as the one from mapping concepts of reasoning support for web engineering, and also Ferdinand's work (for mapping). However, many of the proposals here stated are quite new. The detention of conceptual elements, not only the concepts mapping but also their recognition, it is also proposed in a new ontological modeling using a horizontal structure that give more consistency to the nodes or relations in the ontology. The synthetic URIs were also introduced to avoid the losing of non schematic information. These ones are features that permitted that the quantity of concepts and relations of the resulting ontologies were improved.

Figure 3 shows the complete panorama of this transformation. It is assumed that the files schema of an XML document can be interpreted as conceptual ontologies, that is, an XSD file should be transformed into an OWL file without any complication in a linear way. It happens exactly the same when transforming XML files into RDF files. However, this only works in the theoretical field. The first difficulty comes up when the document contains several namespaces (NS). An iterative process was implemented to transform every single NS that was being found. However, when the complex schema contains a large quantity of references, the recursive process can be delayed or collapsed. In order to avoid this problem, it was defined a maximum level of iterations and created synthetic namespaces that help to define the scope of the transformation of the referred NS. In this way, the recursiveness levels are limited and the time of the transformation is reduced by improving the transformed NS. An OWL file cache of the transformations was also started; this accelerates the process, especially when the information uses references that are public domain, such as encyclopedias and geographic references, among others.

Steps for mapping an ontology:

1. Create a group file with the transformation rules based on standard OWL.
2. Obtain the NS from the XML and transform them into an ontology.
3. Define a recursive level.
4. Determine the policies/rules for the generation of synthetic NS.
5. Measure quality level of the generated ontologies.
6. Execute a SPARQL and DL queries to validate the ontology.

3.2. XSLT Transformations

In a first approach, XSLT technology was used [18] that provides support to several forms to make the transformation. This was made through transformation rules, many of them using XQuery and XPath data model. The OWL 1.1 standard [19] was chosen. Every single father element was taken as an OWL class. This happened as well to every attribute, where each one was joined with a class through the properties.

3.3. Recurrence (Mapping Complexity)

Recurrence works in a linear sense; however, when the cycles are not homogeneous, that is, when a son loop uses data from a father loop, the complexity starts to be high in the script and this could be even more complicated when the levels of nesting are higher than three, due to the exponential growing. Figure. 1 shows an electronic invoice that has about six levels of nesting. The transformations using XSLT worked well on simple XML that had a few elements and a low diversity in the references; however, the maintenance of the XSLT models is complex, especially when the diversity of the structures becomes larger. This requires the development of a new model where the adaptation of the rules with the number of the used structures is more dynamic. A hierarchical structures processor based on XML formats was developed. It was simple, though, since there are many developed components that deal with these tasks. Perhaps, their only constraints could be related to their over-usage when the elements have many internal cycles.

3.4. Ontological Looping

One feature of the ontologies is their ability to be shared and related [6]. When a document is transformed, it could be necessary to go beyond the limits of the document to complete the models. This can lead to search for other sources and to process them; however, this could be expensive for the transformer (especially if the resources are not available at the moment.)

3.5. Duplicity in the XML Elements

One of the logics conflicts that are more common occurs when the information that is contained in XML elements is repetitive. For example, in the invoicing of a store, the emitted invoices always have the same information regarding the seller; if these cases were omitted, many identical instances would be generated from the same store that semantically would be treated as different.

By incrementing the number of transformed invoices, it makes sense that new assertions make the graph grow. Some elements of the invoices contain information that begins to be repetitive, e.g. when there are several purchases from the same customer, the name, ID, and the address could have been previously registered, and there would be two or more similar nodes in the same graph unlinked. The SAT schema does not define these types of issues, in other words, they do not mention anything about unrepeatability and untiety.

Unrepeatable and Linkables. To identify this node, an additional compliment to the SAT XSD was defined, where the node specification can be defined as unrepeatable. This enables a HASH code with the node and its attributes, this ensures consistency between the links from different CFDIs.

In this way, there is a graph distributed with more number of links and quality, and it avoids the unlinked vertical growth.

Inferences and linked entities, this document did not reach to enter inside the entities inferences; however, at the end of the transformations, it was noticed that with more relations between entities, the queries were less complex, reducing searches of patterns through Regex patterns.

3.6. Improving the Ontological Model

Up to this point, the Ontological Model has been mentioned just in the individual transformation of the CFDIs. An effect in the ontological structure was found, the one that starts its formation in the moment a second invoice is created. It was detected that in the first results of the mapping process the pattern of the model followed a vertical tendency. Figure 4 shows this effect. This brought up negative consequences, especially due to the queries becoming more complex when using regex patterns in their statements. When the number of triplets (links among concepts) was lower, the effectiveness of the inference was reduced and the final results were really weakened.

To avoid this, some adjustments were made: a conceptual validation was implemented by applying the HASH algorithm, which avoided the repetition, and helped to coagulate similar concepts. However, this was not a definitive solution. There are some attributes within the elements of the XML that can not be hashables, such as dates, addresses, balances, and the like. Some of these attributes could be handled just through heuristics. Figure 5 shows the result with the adjustments included. It can be noted that there is more cohesion of the nodes and less redundancy of the information.

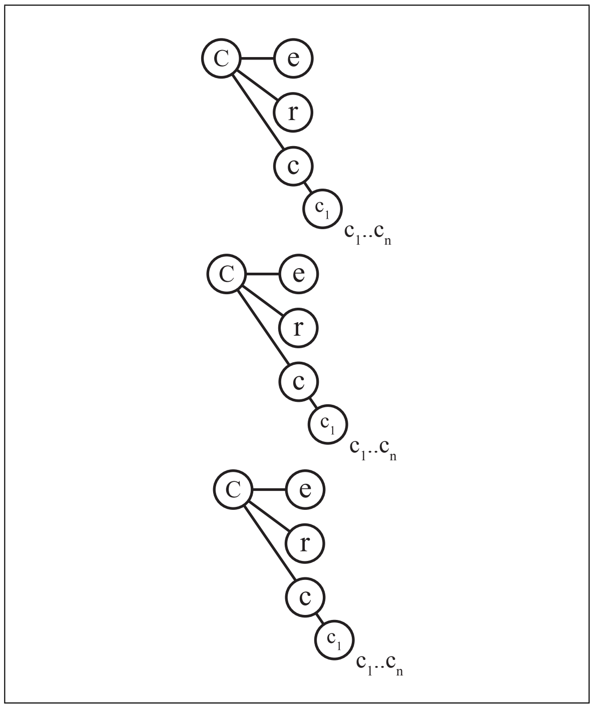


Fig. 4. Ontological Model: The vertical tendency in a graph.

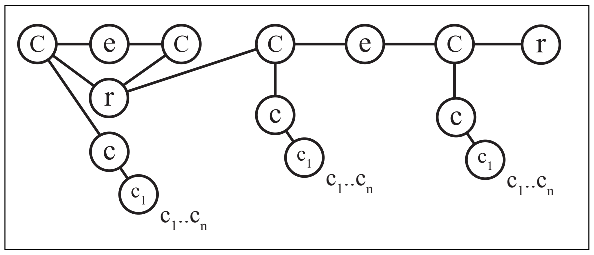


Fig. 5. Ontological Model: The horizontal tendency in a graph.

3.7. *OWL Frameworks*

Initially Jena framework was selected [20] because it is the open source community option, and recently, Apache foundation has adopted it once HP [21] let it open. The architecture of this framework is based on RDF models that from time to time have been modified to make them adequate to the recent OWL standards; however, the updating has not been completed. According with its documentation, they are not fully compatible with the last version of OWL 2 but supported only some characteristics. The main issues come up when the modeling of the individuals of a class are started. The Jena models, though they are configured to use OWL DL, do not create the model according to the OWL 2. They recur to the clustering of RDF kind classes but they do not use the owl:NamedIndividual labels. This may cause performance failures, especially when the ontologies start growing. Owlapi [22] is one of the most robust frameworks nowadays. This model treats an ontology as a set of axioms rather than as a set of triplets. It follows the guidelines of the OWL specifications, and due to this, it is more adequate to write OWL 2 ontologies.

3.8. *Conceptualization of the Information*

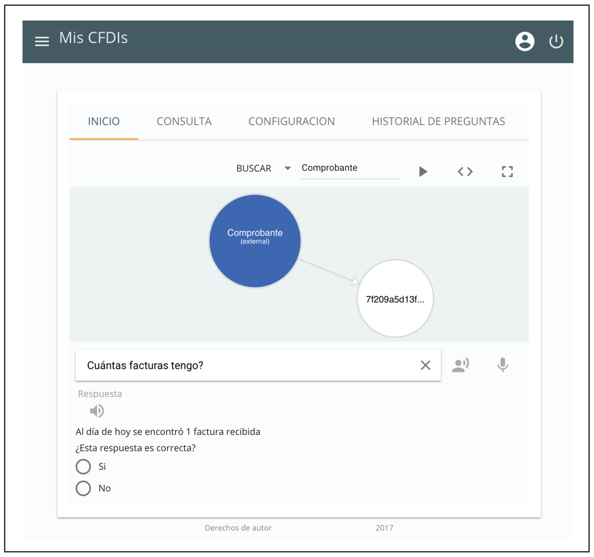
Another challenge that was found was to determine the conceptualization level that the transformed document should reach. It was determined that the conceptualization level reflects the logical processing capacity that the document could have. A description of a strategy to deal with the conceptualization of the XML files is in the following section.

4. Ontoparser

To verify the previously described methodology, the Ontoparser software was developed, which allows to manage the transformations of XML schemas and XML data to RDF/OWL automatically. Ontoparser works in a web application where the clients upload CFDI of the purchases/sales that they have made. The user can create a repository to save the information of the invoices that will be transformed into graphs. It is important to show the user the size of resource/space that is available. Ontoparser takes as a measurement unit the number of triplets that are in a repository. It can be stated that the higher the amount of triplets are in a graph the more processing resources are needed and the more expensive the service becomes.

Once the reception of the XML or Zip is finished, if there are several XMLs, a process in background is executed to make the transformation task. The user can know how gradually the graph is processing the documents separately. It is not necessary that the task is finalized for the user to make any consult about the documents that are being uploaded; it might happen that some information be omitted because it is still not processed. In this version, it is not possible to delete or update the CFDI individually just yet; it is only allowed to delete the whole graph.

Ontoparser uses different storages to save the ontologies. Fuseki TDB [23] is used by default to store the information of a standard user. It has good performance and can hold large amounts of datasets. How-



Cuántas facturas tengo?

Al día de hoy se encontró 1 factura recibida

¿Esta respuesta es correcta?

Fig. 6. Ontoparser: Resulting graph from an user query.

ever, this is a tool to be used in safe zones, without strong security restrictions. But, if the information is sensitive and some security standards are required, Ontoparser is ready to use storages as Virtuoso [24], AllegroGraph [25], and Oracle Graph 12c [26], that offer these characteristics. Figure 6 shows a resulting graph from a QUERY in natural language. A user asks about how many invoices have been received today (Ontoparser can even listen to the voice of the user). Ontoparser shows a graph with the information attached to a short answer in audio.

All the API's can be called from the REST services. Basically, there are two main services to transform an XML document into an RDF, XML to RDF, and XSD to OWL. To transform an XML into RDF, the XML must have well-defined schemes that are used in every node, attribute, or value in the body of the structure.

However, if an XML does not have the schemas well defined, the references will be created in a synthetic manner, mainly with the knowitive.com namespace.

4.1. *Creating an RDF/OWL DL Output from an XML (XML to RDF/OWL)*

It is possible to transform more than one XML if they are concatenated (using ";") with the URIs in the entry string (the OWL output will hold all the information of the URIs received). It is also possible to choose the notation of the OWL output, by default RDF/XML is set, even though there are some other formats that are possible to be chosen, such as

```
/*Sparql name:getFrequentItems*/
PREFIX cfdi: <http://www.sat.gob.mx/cfd/3#>
SELECT ?descripcion(COUNT(?descripcion)AS ?Nit)
WHERE
{ ?concepto cfdi:Descripcion ?descripcion }
GROUP BY ?descripcion
ORDER BY DESC(?Nit)
LIMIT 5

/*Sparql name:getExpensesLastInvoice*/
PREFIX cfdi: <http://www.sat.gob.mx/cfd/3#>
SELECT ?monto
WHERE
 {?Comprobante cfdi:Total ?monto;
     cfdi:Fecha ?ultimafecha
    {SELECT(MAX(?fechaFactura) AS ?ultimafecha)
     WHERE
      {?Comprobante cfdi:Fecha ?fechaFactura}
    }
 }

/*Sparql name:getTotalAmountInvoices*/
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX cfdi: <http://www.sat.gob.mx/cfd/3#>
SELECT (SUM(xsd:float(?importe)) AS ?total)
WHERE
 { ?Comprobante cfdi:Total ?importe }
```

Fig. 7. Testing ontology: Three different Sparql queries for testing the new invoice ontology.

RDF/XML-ABBREV, N-TRIPLE or N3. In the last version, the VOWL format [27] was added. VOWL provides a JSON notation that facilitates the rendering of the ontology in a GRAPH.

The outcome of the transformation can be either a URI type (that means, the file is set/written in a storage and then is assigned a URI) or a body type where the service request contains the complete OWL file.

Additionally, the ontological framework used for the transformation can be chosen, Jena or OWLAPI.

4.2. *XSD to OWL*

This function is similar to that one previously described, but its objective is just to transform a schema from XML into an ontology or TBox. The XMLtoRDF parser module is called internally when in the XML file a reference to a schema is found. It uses the same parameters as those for the previous service.

4.3. *Outcome/Tests and Results*

The tests was made with different sizes of XMLs. It was arbitrarily classified them as Simple when they are no longer than 40 nodes, and Complex when they are longer than the simple ones, or when they use

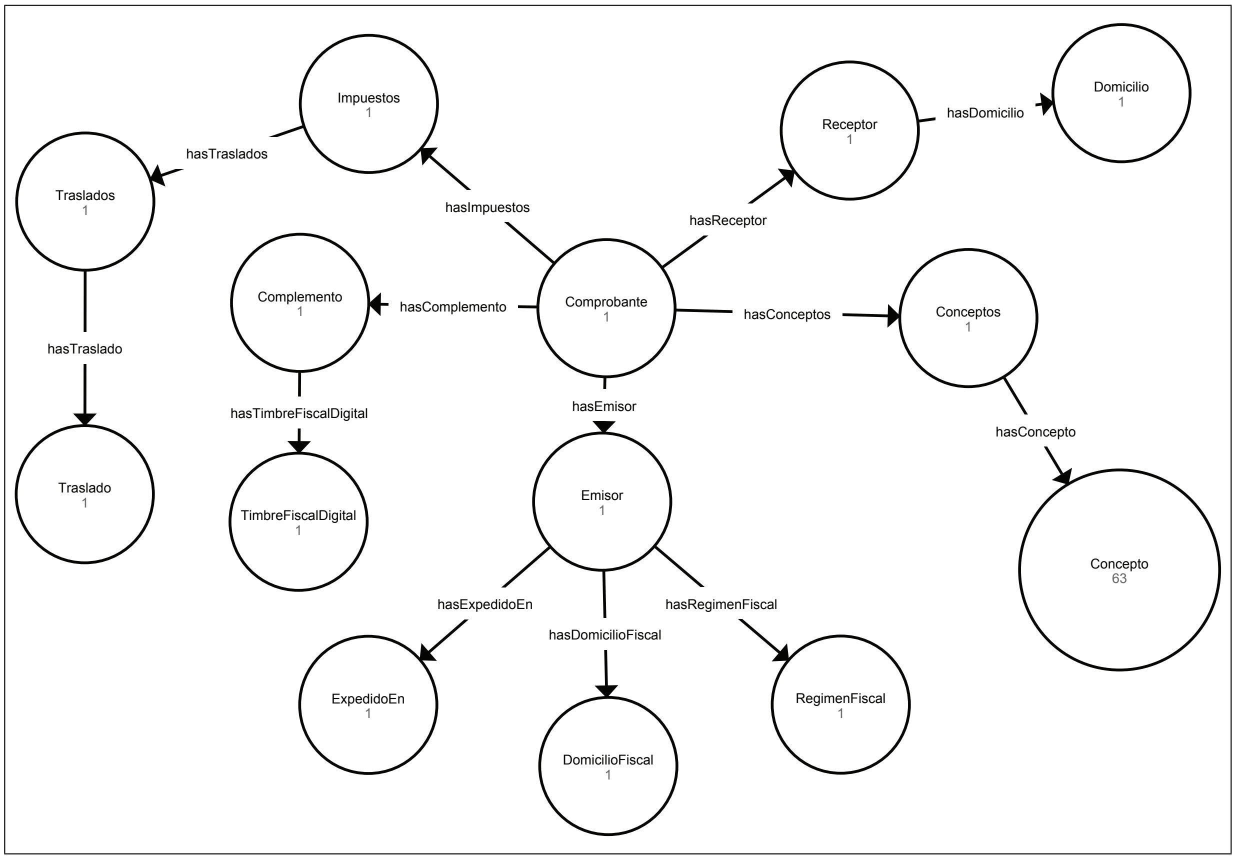


Fig. 8. OWL: TBox and ABox relation results.

more than one XML (in this case, the nodes went up to 1000). The transformation time, using a virtual instance with 1 CPU Intel Xeon 3.3 GHz and 1GB memory, the execution process was not so long to consider it an issue. In any case the test was reached but when the XSD schemes were not available, in this situation, it was necessary to implement a cache; by doing so, it was possible to reduce the transformation time to a few seconds. It was found that there is not much difference in the performance when using Jena or Owlapi; however, as explained above, the model structures are completely different. It was decided to just evaluate the outcome obtained with Owlapi, since the version of the OWL 1 that Jena uses does not deal with individuals explicitly.

The first necessary adjustment was for the Object Properties that do not have a double way, that is, the one way linked between A and B classes was changed to double way to have consistency, mainly in the queries of Sparql Figure 7.

Table 1
Ontoparser performamce results

Outcome	CFDIs	Nodes	Triplets
Simple	1	38	543
Complex	10	1,000	3,266

Table 1 shows the tests results. The results are divided in three columns. The first one shows the kind of test, basically there were just two kinds of tests, the so called Simple, that consisted of just one CFDI and the complex ones, that consisted of several CFDIs invoices from the same client; in this case, about 10 invoices were used. The second column shows the quantity of nodes that were detected during the mapping. The minimum quantity of nodes of a CFDI is 30, without considering the instances, the more the invoices the higher quantity of nodes. One invoice can have one or many concepts, that is why this number increase is not linear. At the end, the last column shows the quantity of triplets constructed during the mapping process. It is very interesting to observe in this table that it can be thought that the higher the amount of CFDIs the more duplicity, once the nodes were duplicated, eliminated or linked, a compact ontology was the result. This is how its use and exploitation is simplified.

The Figure 8 shows the ontology resulting from the transformation process. It can be seen that every concept has a number of individuals linked. It can be considered that the outcome is satisfactory since the ontologies obtained were tested with queries from Sparql and LD to prove the information remain logical, with a low consumption of resources.

When a transformation is made and this is added to an existent ontology, it is necessary to consider the repetitiveness of the concepts, e. g., the unique and not duplicate concepts like identification IDs. Besides, there is a doubt when classifying the properties in Data or in Objects, since there is not a clear rule to distinguish them.

5. Future works

By the time this report was finalized, there were some changes in the Mexican policies on the Ministry of Finances that affected the structure of the CFDI here described. The last version 3.3 of the CFDI [28] has some changes that improve the description of products or services that are included in the sale, that means that nowadays they have to be classified based on a universal catalogue of products or services, named UNSPCP (United Nations Standard Products and Services Code) [29], that holds about 50,000 classifications. This will allow to include, in the following projects, classifiers of products or services that are capable of detecting relations among invoices and, by doing so, improving the semantic quality of the document.

On the other hand, A NLP strategy is being developed in order to build a semantic QAS so that decision makers can exploit the knowledge generated and organized in the extracted ontology. This software is currently under development and will be reported later.

This work forms part of a wide-range research whose objective is to introduce semantic technologies to the small and medium-sized companies in Mexico. Not only the semantic extraction of CFDIs is of interest for this project, but also the use of Spanish language given the characteristics of Mexico's companies, and a simple form to present the information so that users with no or little experience in this technology can exploit the information that is extracted. The ultimate objective is to integrate the resulting works in an application that can be marked in business packages or manager tools. Take advantage of the legislation changes that enable people to have access to a great quantity of information stored in the cloud.

6. Conclusions

The transformation implemented in this report from CFDI to ontology was successful according with the proposed mapping strategy. It was possible to create a robust and compact ontology that allows the usage of semantic components, such as reasoners, in a very simple and efficient way. The developed prototype mapped successfully the CFDI information to a semantic structure (OWL), and the results were verified through SPARQL and DL queries. Thus, it can be confirmed the quality of the resulting model, which is high if it is considered that there was no loss of information. In addition, the prototype had an optimum performance for the hardware and software requirements. The processing of CFDI nodes with descriptions in a natural text (without structure) is still pending. This work left them out, but it is important to keep them in mind in subsequent work because they can offer interesting knowledge that currently is stored only in simple strings of text (without interpretation).

This first approach is important because it opens the way to use semantic technologies based on structured information in Mexico, and it will allow in the future to incorporate semantic applications of more complexity, e.g., a question and answer system could use semantic structures to improve the assertiveness in their answers.

References

[1] Mexican Government. (2015, May 29). *Estrategia Digital Nacional (EDN)* [Online]. Available:http://www.presidencia.gob.mx/edn.

[2] Data.gov. (2017, Jul. 29). *Developers: Semantic Web* [Online]. Available: https://www.data.gov/developers/semantic-web.

[3] Datos.gob.es. (2017, Jul. 29). *Ontology* [Online]. Available: http://datos.gob.es/es/talk-tags/ontology.

[4] W3C. (2017, Jul. 29). *Extensible Markup Language* [Online]. Available: http://www.w3.org/XML.

[5] T. Berners-Lee, J. Hendler, and O. Lassila, *"The semantic web,"* Sci. Am., vol. 284, no. 5, pp. 34-43, May 2001.

[6] D. Fensel, J. Hendler, H. Lieberman, and W. Wahlster, *Spinning the Semantic Web: Bringing the World Wide Web to Its Full Potential.* MA, USA: The MIT Press, 2005.

[7] F. Baader, D. Calvanese, and D. McGuiness, *The Description Logic Handbook: Theory, Implementation and Applications,* 2nd ed. Cambridge UK: Cambridge University Press, 2010.

[8] P. Hitzler, M. Krotzsch, and S. Rudolph, *Foundations of Semantic Web Technologies.* Broken Sound Parkway NW: CRC Press, 2011.

[9] XML Schema Part 0: Primer Second Edition (2017, Mar 1). [Online]. Available: https://www.w3.org/TR/2004/REC-xmlschema-0-20041028.

[10] T. Berners-Lee. (2017, Jul. 29). Why RDF Model is Different from the XML Model [Online]. Available: https://www.w3.org/DesignIssues/RDF-XML.html.

[11] W3C. (2017, Jul. 29). RDF Schema [Online]. Available: https://www.w3.org/TR/2014/REC-rdf-schema-20140225.

[12] W3C. (2017, Jul. 29). OWL Web Ontology Language Semantics and Abstract Syntax [Online]. Available: https://www.w3.org/TR/owl-semantics.

[13] R.González, *A Semantic Webapproach to Digital Rights Management,* no. 1, Nov. 2005.

[14] Ontmalizer. (2017, Jul. 29). A tool that performs comprehensive transformations of XML Schemas (XSD) and XML data to RDF/OWL automatically [Online]. Available: https://github.com/srdc/ontmalizer.

[15] OpenRefine. (2017, Jul. 29). A free, open source, power tool for working with messy data [Online]. Available: http://www.openrefine.org.

[16] TopBraid. (2017, Jul. 29). Visual modeling environment [Online]. Available: http://www.topquadrant.com/topbraid.

[17] M. Ferdinand, C. Zirpins, and D. Trastour, "Lifting XML schema to OWL," ICWE, vol. 3140, no. 4, pp. 354-358, Oct. 2004.

[18] W3C. (2017, Jul. 29). Transformation W3C [Online]. Available: https://www.w3.org/standards/xml/transformation.

[19] W3C. (2017, Jul. 29). OWL Web Ontology Language Overview [Online]. Available: http://www.w3.org/TR/owl-features.

[20] Apache Jena. (2017, Jul. 29). Apache Jena - Home [Online]. Available: http://jena.apache.org.

[21] HP. (2017, Jul. 29). Hewlett-Packard Company [Online]. Available: http://www8.hp.com.

[22] M. Horridge and S. Bechhofer, "The OWLAPI: a Java API for OWL ontologies," Semant. Web, vol. 2, no. 1, pp. 11-21, Jan. 2011.

[23] Apache Jena TDB. (2017, Aug. 28). Apache Jena TDB [Online]. Available: https://jena.apache.org/documentation/tdb.

[24] Virtuoso. (2017, Aug. 28). OpenLink Virtuoso Home Page [Online]. Available: https://virtuoso.openlinksw.com.

[25] AllegroGraph (2017, Aug. 28). AllegroGraph - Semantic Graph Database - Franz Inc. [Online]. Available: https://franz.com/agraph/allegrograph.

[26] Oracle 12c (2017, Aug. 28). Oracle 12c spatial and graph [Online]. Available: https://www.oracle.com/database/spatial.

[27] S. Lohmann, S. Negru, F. Haag, and T. Ertl, "Visualizing ontologies with VOWL," vol. 8982, no. 4, pp. 154-158, Jun. 2015.

[28] Factura Electronica (2017, Aug. 25). *SAT* [Online]. Available: http://www.sat.gob.mx/informacion_fiscal/factura_electronica.

[29] UNSPCP (2017, Aug. 25). *United Nations Standard Products and Services Code* [Online]. Available: https://www.unspsc.org.