OPEN ACCESS

International
Journal
Of Modern Engineering Research (IJMER)

Dublin Core RDF Metadata Automatic Generation from Stored Information in A Relational Database

Angela Ríos Luna[1], Irene Aguilar Juarez[2], Joel Ayala De La Vega[3],
Leticia Dávila Nicanor [4]

[1, 2,3] *(Centro Universitario UAEM Texcoco, Universidad Autónoma Del Estado De México, México)*
[4]*(Centro Universitario UAEM Valle De México, Universidad Autónoma Del Estado De México, México)*

ABSTRACT: *When a website is created, it is designed the way texts, images and sounds will be presented to the people. But it is equally important to establish the channels through other sites and software agents consume such information. Part of the information of Internet 3.0 sites is available in open formats or is described in metadata files. The structure of the semantic web sites enables filtering and automatic promotion of relevant content to users, and allows internet sites to interact. This paper describes the development of a software component for Java applications that generates Dublin Core standard RDF metadata files from classes with annotations. This would allow its integration into existing information systems with minimal or no changes to the database, with the additional advantage of synchronization between the information in the database and its semantic equivalent published online.*

Keywords *: Dublin core, rdf, Semantics, object in java, annotations*

I. INTRODUCTION

The work area known as semiotics is the study of signs. Signs are considered the elements that compose communication; evidently communication is a social phenomenon as it involves human interactions and intentions, effective communication occurs when the intension of the issuer corresponds largely with the message interpretation made by the receiver. Signs and sign systems can be grouped into four interdependent levels: pragmatics, semantics, syntax and empirical. Pragmatics deals with the aspect of intentions, semantics with the meaning of a message, syntax with the formalism used to represent messages, and empirical takes on the used signals to encode and transmit a message. [1] Semantics is the study of signs reference. Communication involves using and interpreting signs. When we communicate, the issuer must state its intentions using some signs. In a face to face conversation, this will involve using linguistic signs. The receiver of the message must interpret the signs, that is to say, it must assign some meaning to the signs in the message. Semantics deals with this process [1].

Semantic Web is an extended web, provided with greater significance in which Internet users can find answers to their questions faster and easier thanks to a better-defined information [2]

Companies that offer Internet search services, generate keys whereby a site will appear as a search result. The sites are indexed starting with the home page and recursively following the hyperlinks that connect files together. Based on the large volume of content that exists on the Internet, and that increases day by day, Tim Berners-Lee, father of the internet and director of the World Wide Web Consortium, initiated a solution for dealing with this content explosion: let software find and manage Internet information for us. It means, changing the current document retrieval prototype for and from humans to delegate tasks to software.

W3C (World Wide Web Consortium) standards were created to lead the Web to its full potential by developing common protocols that promote its evolution and ensure its interoperability. The model of using the Web changes from being a document repository to be a great base of knowledge for advanced systems capable of performing complex tasks. Semantic Web, an approach by the creators of the WWW that attempts to create a new network for web applications, details a way to publish semantic content in RDF (Resource Description Framework) format as it is one of the standards defined by the W3C used for data exchange.

In 2007 the project *Linked Open Data* arose in the W3C, with the aim of spreading the semantic web. Each year a graph showing the links created between the semantic web network is published, it is possible to see that the growth has been exponential [3]. A current project is the DBpedia instance, this extracts information from Wikipedia to propose a semantic version.

Dublin Core is an organization dedicated to promoting the adoption of interoperable metadata standards, in order to be implemented XML (Extensible Markup Language) is generally used and it is based on the RDF Core was created in 1995 through initiatives by the associations of American librarians, and sponsored by OCLC (On Line Computer Library Center). It has its origin in an intellectual circle of Dublin, in the state of Ohio in the United States. Dublin core is associated with 52 expert researchers in the fields of library science, computer science, text encoders and related areas, with the aim of promoting the development of descriptive records of online information resources. Its progress has occurred in parallel with the development of the XML and the RDF; in October 2001, it was published the set of elements of Dublin vocabulary (DCMES, Dublin Core Metadata Element Set) which is defined by ISO in the ISO 15836 norm in 2009 and the Z39.85-2012NISOstandard [4].

This paper describes a strategy for automatic generation of semantic content (resources + metadata) from information stored in a relational database using the Java programming language. The RDF standard and international metadata format Dublin Core are used as they are the most cited and accepted, this is a product format an international and interdisciplinary effort with a very intense life, and the most influential in relation to the development of the theory of the use of metadata for information retrieval on the net.

II. PROBLEM

The web helps us to easily communicate with everyone at any time and at low cost, yet we generated an overload of information and diversity of information; organizations currently store in its databases confidential operational information that is not shown on their websites, and public information that needs to be spread. It is necessary to publish authorized information of an organization so that search services providers can index content and spread to the general public.

The need for sharing information between institutions and companies, without duplicated information. In recent years the number and size of RDF files with semantic data availability applied to any discipline has increased [5]; the data represented in RDF can be interpreted, processed and reasoned by software agents. Hence the change from the relational model to the RDF model for handling information is one of the challenges of current research [6]. So, according to the guidelines of the Web 3.0, our content requires a visual side (for people) and metadata (to be processed automatically by machines).

The information represented semantically uses different definitions owned by each company and is virtually impossible to relate it correctly as there are many distributed graphs all along the network. Therefore, the Linked Data initiative has emerged in order to link all available information into the graphs of the Web and build a unique graph that represents all the knowledge stored in the net [7].Given the volume of information available on the Web, it is inevitable to automate some of the processes involved in the development of large structures of knowledge, applying the procedures that have been experimented since the retrieval of information. Currently there are companies that develop tools for automated classification [8].The massive publication of data involves having to preserve the privacy of people and institutions, ensuring that it is not possible to deduce certain confidential information indirectly. Moreover, the fact that anyone can publish and link data on the web means that some aspects about the origin of the data, its quality and reliability of sources must be taken into account [9].

The web-based systems face challenges such as interoperability, the use of domain ontologies, contextualization and consistency of metadata. Such problems are related to the attempt to represent the information on the web in a way that computers can understand and manipulate. The main goal of semantics is to reuse the resources available in Web-based technologies through norms [10].

III. RDF

The Semantic Web architecture is focused on the RDF model, which is a universal format for data on the web. Currently, it is used for conceptual description or modeling the information that is implemented in web resources using syntax annotations.

The RDF provides a semantics for metadata; a better precision in resource exploration than the achieved by the search engines that track the full text, and better applications. All this while the corresponding schemes are developed. In general, the RDF provides the basis for generic tools that create, manage or search data on the web in an understandable way for computers, promoting the transformation of the web into a repository of information manageable by computers. The objectives of linked data are using data on the Web in the same way as the documents, linking data together, and using data as a collection of a self-related datasets, technologies must be available in a common RDF, in order to access existing databases either through a conversion or during runtime[12].

Advantages of the RDF data management as a database object:

- It is easier to model RDF applications.
- RDF data can be easily integrated with other company data.
- Reuse of RDF objects makes it possible to develop applications more efficiently.
- Objects abstraction, and encapsulation of RDF-specific behaviors make applications easier to be understood and maintained [1]
- No mapping between the RDF client-side objects, database columns and tables containing triplets is required.
- No additional configurations for storing RDF data are required [11].

1. Data model based on graphs

The structure of the syntax is set of three elements called triplets and is composed of a "subject, predicate and an object" [12], this set is called RDF graph, where an RDF graph can be displayed as a node in the graph and an arc of the graph, where each triplet is represented as a link node.

It is noteworthy that these three elements are represented by a URI (Uniform Resource Identifier).

- In the subject: it is described whom is being talking about,
- In the predicate, is also known as property: this represents a connection between a subject and an object, usually a subject's relationship with something or an attribute of the subject.
- In the object: the value of the declared.

Next, an example of a triplet is shown in Table 1:

Table 1 Basic Triplet

SUBJECT	PREDICATE OR PROPERTY	OBJECT
Httt://ejemplo.com/pepe	Httt://ejemplo.com/nacimineto	“14/08/1988”

IV. DUBLIN CORE

Dublin Core is an organization focused on "core metadata" for simple descriptions and generic resources. It comprises fifteen elements, it is noteworthy that "Dublin Core" has been ratified as IETF RFC 5013, ANSI / NISO standard Z39.85-2007, and ISO 15836: 2009.

Since 2000, the Dublin Core community focused on "application profiles" - the idea is that the Dublin Core metadata records would be used along with other specialized vocabularies to meet specific application requirements. It has worked with the World Wide Web Consortium in a metadata data generic model, the RDF. As part of an extended set of metadata terms DCMI, Dublin Core became one of the most popular vocabularies to be used with RDF [13], supporting the Linked Data movement.

Dublin Core aims to define a basic set of attributes that serve to describe all existing resources on the network, this format will help search engines in global retrieval of online information, and this goal turns it into a general purpose format.

Dublin Core defines fifteen elements which can be modified and scaled due to its flexibility, this allows authors of Web pages to encode their documents at the time of generating them.

Elements mainly related to the resource content:

1. Title
2. Subject
3. Description
4. Source
5. Languaje
6. Relation
7. Coverage

Elements mainly related to the resource when viewed as an intellectual property:

8. Creator (author)
9. Publisher (editor
10. and other collaborations
11. Contributor (other authors / collaborators)
12. Rights.

Elements mainly related with the instantiation of the resource:

13. Date
14. Type (resource type)
15. Format
16. Identifier

V. RELATED WORKS

Recent studies show that it is possible to transport the relational model to the RDF graph-based model and the implementation of the Dublin Core metadata model. Through different methods and tools it is possible to do upgrades that reduce the cost of transferring information from databases to the RDF data model. The Research [6] focuses on the incremental update of RDF graphs from relational databases. When a change happens RDF triplets are generated representing updates, this information is updated on the RDF graph. This approach reduces the cost in time and computational resources compared to traditional techniques. This work shows the importance of following an incremental approach, which avoids processes from scratch and takes advantage of existing results before the changes.

Another research project is "TRIOO" in this work were studied and implemented RDF data models in an object-oriented language [14]"; the author proposes a technology that allows a simple way to use RDF data directly from object-oriented languages, allowing the origin and form of the data does not modify the object-oriented design, while the data semantics is reflected as accurate as possible. It is important to mention that this research has a defined architecture but did not culminate.

In the Article Integration of Heterogeneous Relational Data bases: RDF Mapping Approach [15] it is reported that the mapping process is important in data integration and knowledge of multiple heterogeneous relational data bases. The mapping from a relational database to a description that uses the RDF shows promising results without compromising the semantics of the data. Integration is also important when carrying out multiple simultaneous databases queries. Mapping is an important process to provide homogeneous points of view for the user. Mapping a relational database (or diagram entity relationship) to an RDF schema for data representation is a solution to overcome the problems of semantic and syntactic heterogeneities. The RDF is a common data format for the Semantic Web. It is a language that describes different kinds of objects (resources), their properties and the relationships between them by using statements.

VI. PROPOSED SOLUTION

The developed software component is responsible for generating RDF files from objects. This process can be part of an information system, website, RSS (Really Simple Syndication), web service, etc. It depends on the context programmer and the method by which ontologies are published.

An object is anything with an independent existence from the universe of discourse being modeled.

The object model allows to integrate within a single modeling procedure both dynamics data and the various aspects of the data structure. An object model is a series of objects. Objects can be considered as extended entities, entities with a certain behavior.

A class is the abstraction of the common characteristics of an objects group. Objects can have common attributes, relationships or methods, object models usually do not indicate objects but their classes. An attribute is a property of a class. A relationship is a connection between classes. A method establishes a behavior of a class.

In object modeling a relationship can be of three kinds:

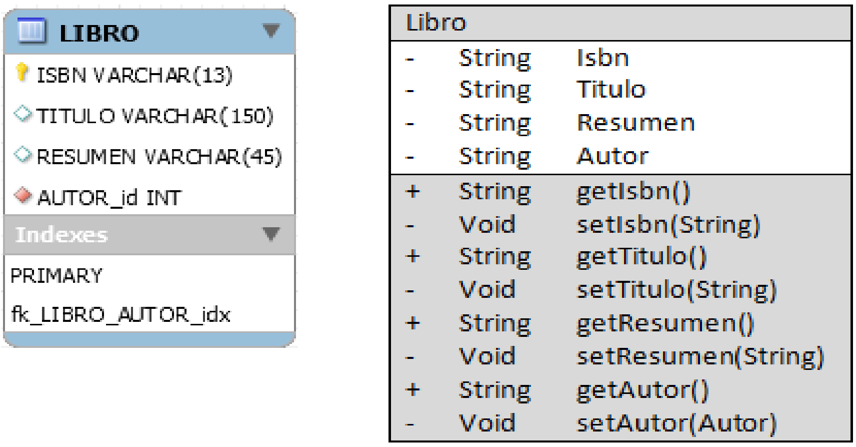
Association. An associative relationship establishes a connection between instances of classes, and are defined by their cardinality and optionality. Therefore, the association relationships are common for those of structural models.

Generalization. This relationship establishes levels of abstraction between object classes. Generalization is the extraction process of the corresponding common features of a group of object classes and the suppression of the differences between them.

Aggregation. This relationship serves to gather a collection of different classes into a unit or aggregation. An aggregation relationship occurs between whole and its parts. An aggregation is an abstraction in which a relationship between objects is considered as another top-level object.

When a database that represents a particular environment is designed, entities are represented as tables. Similarly, in OOP, each entity in the problem domain is coded as a class[1]

A correspondence between tables and classes of an application is called object-relational mapping. An example of Object-Relational Mapping is shown in Figure 1.The persistence API encapsulates the access to the database and allows the programmer to work on an object-oriented way. Persistence is the ability of objects to survive beyond the execution of the created routine or program.



LIBRO
ISBN VARCHAR(13)
TITULO VARCHAR(150)
RESUMEN VARCHAR(45)
AUTOR_id INT
Indexes
PRIMARY
fk_LIBRO_AUTOR_idx

```
Libro
-  String   Isbn
-  String   Titulo
-  String   Resumen
-  String   Autor
+  String   getIsbn()
-  Void     setIsbn(String)
+  String   getTitulo()
-  Void     setTitulo(String)
+  String   getResumen()
-  Void     setResumen(String)
+  String   getAutor()
-  Void     setAutor(Autor)
```

Figure 1 Object-relational mapping, table and class for the Book entity.

One class is created in the application per each table in the database, by each column in each table and a variable is added to the corresponding class. What class corresponds to which table and which variable to which column are specified using annotations. It is precisely these classes the ones meant to be published as RDF files as each one represents a particular concept.

In order to publish these objects as RDF files, the programmer places annotations in the classes whose objects are meant to be published. These annotations are a form of metadata that provide information about a program without being part of the program itself. The annotations have no direct effect on the operation of the code [16].

In order to mark the classes and publishing fields, there were designed: A set of four RDF annotations (Table 1) and a set of 15 Dublin Core Annotations (Table 2).

Table 2 Developed annotations

Annotation	Definition
@RDFSujeto(url, extension):	This annotation is placed in the classes that will be converted to RDF files. In this part the repository URL where the files will be placed in and its extension are specified. E.g. "www.library.com/books"
@RDFIdentifier	It is placed on the method that returns a string whose value is used as the metadata file name. This identifier must not produce duplicated values. It is equivalent to the primary key in the database. E.g."book32165", "the little Prince";
@RDFPropiedad (url, prefix, nombre)	It is placed to alphanumeric variables whose values will be inserted as nodes in the RDF file. It is necessary to specify the prefix, tag, and the name space of the node to generate. It is used to define custom fields in the RDF file.
@RDFResource(url, prefix, nombre)	It is placed on variables that refer to other classes. In the RDF file a hyperlink to another RDF file is added. The hyperlink is built depending on the referenced class annotations.

The RDF annotation *Property* is used to specify custom fields in the RDF file. Fields Dublin Core standard are contained in the set of additional annotations shown below in Table 3.

Table 3 Dublin Core annotations set.

Annotation	Definition
@RDFDublinCoreTitle	Title
@RDFDublinCoreSubject	Subject
@RDFDublinCoreDescription	Description
@RDFDublinCoreSource	Source
@RDFDublinCoreLanguaje	Language
@RDFDublinCoreRelation	Relation
@RDFDublinCoreCoverage	Coverage
@RDFDublinCoreCreator	Creator

@RDFDublinCorePublisher	Publisher
@RDFDublinCoreContributor	Contributor
@RDFDublinCoreRights	Rights
@RDFDublinCoreDate	Date
@RDFDublinCoreType	Resource Type
@RDFDublinCoreFormat	Format
@RDFDublinCoreIdentifier	Identifier

An class example (diagram) with annotations is shown below in Table 4:

Table 4 Annotations diagram.

@RDFSujeto(url=”http://www.uaemex.com/cursos”, extensión=”rdf”) Curso
Private id:Integer
Private nombre:String
Private descripción:String
Private fechaInicio:Date
Private profesor:Profesor
@Transient @RDFIdentifier Public getIdRdf:String
@RDFDublinCoreTitle Public getNombre:String
@RDFDublinCoreDescription Public getDescripcion:String
@Transient @RDFPropiedad(url=”http://www.uaemex.com/cursos” ,prefix=”uaem”, nombre=”fechaCurso”) Public getFechaCurso:String
@Transient @RDFDublinCoreCreator Public getCreador:String
@RDFResource(url=”http://www.uaemex.com/profesores”, prefix=”uaem”, nombre=”profesor”) Public getProfesor:Profesor
Otros getters & setters

We can observe that cover all standard dublin core labels depends on the information available in the database.Otherwise the missing fields must be added, although some derived attributes can be treated as object-oriented using additional methods that concatenate variables. In the previous example @Transient marked getters are methods whose value is not exactly equal to that offered by the attributes and are not in the interest of persistence api, as can be numeric fields and dates that must be converted into text strings.

Annotated classes are processed using the Java API REFLECTION. This suite contained in the JDK allows to inspect objects and classes, and obtain information from their variables and methods. The reflection API also allows invoking the execution of methods by name. Annotations and the values obtained from the inspection of objects generates a temporary database structure which is the basis for generating the RDF content as text string eventually.

The result may, as an instance, be published in a public section of the application server and the rest of the website will refer to them using hyperlinks. Or an RSS news archive can be generated to announce the current courses, etc.

VII. IMPLEMENTATION OF THE PROPOSAL

To implement this proposal a system developed with the architecture MVC (Model, View, Controller) was developed, where the presentation layer user interface is shown, the business logic layer receives user requests and responses are sent after the process, and data resides in the data layer. Implementing the MVC pattern allows to easily maintenance web applications due to a clear separation among the presentation layer, business logic and data access. For the development of this research a module is created in the business logic layer for RDF; so that later on the information is statically published on a public website and in parallel the RDF is stored in a repository; in Figure 2, the architecture of this proposal is shown.

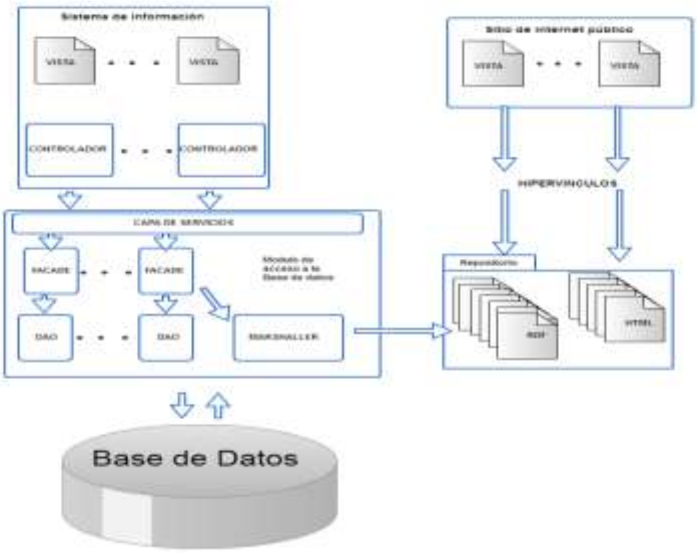
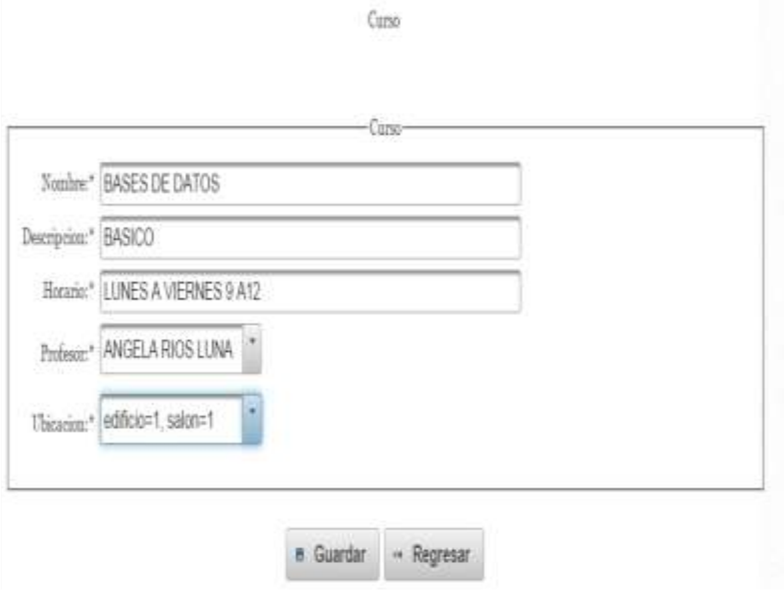


Illustration 2 Proposed Architecture.

This system was developed to manage information on courses offered by teachers at a university. This application requires to be a dynamic web application in which some of your information needs to be released as open content, in a static and semantic format for external users, referenceable by a unique URL. The web application was developed with 2.1 JSF technology (Java Server Faces) implemented with the Primefaces 5.2 component library; web application development framework Spring 4 and Hibernate 4 persistence API were used. The information was stored in a MySQL relational database. All this, into a Tomcat 7 server on JVM 1.7 configured to enable writing on the server. The descriptive classes of objects were developed through the Reflection API, to do so, the necessary notations for the application where defined. JAXB was used to obtain the HTML and XML pages that publish the desired data. The web application is modified so that when a CRUD (Create -Read -Update -Delete) operation is run, it can be performed on any item in the catalogues, [17]. The modifications also allow to generate the respective static HTML files that are saved in a public folder in the Tomcat server.

The web application allows to following up the courses administration. Figure 3 shows the new curse registration interface.



Curso

Curso

Nombre:* BASES DE DATOS

Descripcion:* BASICO

Horario:* LUNES A VIERNES 9 A12

Profesor:* ANGELA RIOS LUNA

Ubicacion:* edificio=1, salon=1

Figure 3 View of the registration of a new course

With software developed in this work it is accomplished to automatically generate the RDF for each course, and this is stored in a public folder in the Tomcat server; below, the generated RDF is shown in Figure 4.

```
<?xml version="1.0"?>
<rdf:RDF
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:dc="http://purl.org/dc/terms"
xmlns:uaem="Http://www.uaemex.mx/cursos">
<rdf:Description rdf:about="Http://www.uaemex.mx/cursos/Curso1.html">
<dc:title>matemáticas 1</dc:title>
<dc:description>algebra</dc:description>
<dc:creator>Juan_Perez_Hernandez</dc:creator>
<uaem:fecha_de_inicio>Mon Oct 24 00:03:59 CDT 2016</uaem:fecha_de_inicio>
<uaem:profesor
rdf:resource="Http://www.uaemex.mx/profesores/Profesor_Juan_Perez_Hernandez.html"></uaem:profes
or>
<uaem:horario>L-V 7:00 a 9:00</uaem:horario>
</rdf:Description>
</rdf:RDF>
</xml>
```

Figure 4. The generated RDF

The Generated information by the RDF can be displayed as a graph (see Figure 5).

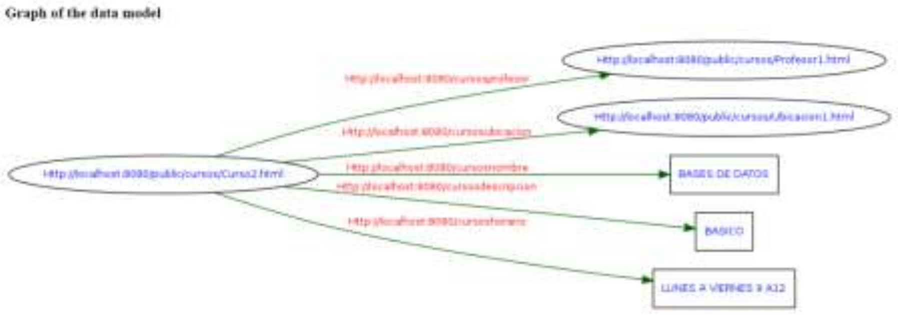

Graph of the data model
Http://localhost:8080/public/cursos/Curso2.html
Http://localhost:8080/cursos/profesor
Http://localhost:8080/public/cursos/Profesor1.html
Http://localhost:8080/cursos/ubicacion
Http://localhost:8080/public/cursos/Ubicacion1.html
Http://localhost:8080/cursos/nombre
BASES DE DATOS
Http://localhost:8080/cursos/descripcion
BASICO
Http://localhost:8080/cursos/horario
LUNES A VIERNES 9 A12

Figure 5 Representation of a course as a graph

IX. CONCLUSION AND FUTURE WORK

Unlike other options, the generated software component can be incorporated into existing databases and applications, and is the programmer who defines the architecture that allows publishing the metadata files.

The programmer can establish a strategy that will allow the information into the invoked metadata files to be updated whenever data is updated in the database through the application, keeping them synchronized.

It will also be possible to develop general purpose applications for end users who are not programmers, and work with specific requirements and generate semantic content. One advantage of RDF annotations is that the reuse of persistence classes without conflicting with the already developed systems, thus reducing the required effort to be implemented and the risk of changing the system operation.

The incorporation of metadata allows the interaction between websites and the creation of new communication and commerce channels for businesses and institutions. Semantic sites position institutions as sources of reliable and referential content, increasing its presence on the Internet.

The dublin core metadata definition standard is internationally recognized. It is flexible, interoperable, semantic and in constant growing and evolving through a formal institution consortium by DCMI, it is noteworthy that Dublin core has bonds with Linked Data in order to achieve standardized files header metadata, this would ease automatic management and improve the effectiveness of search engines.

Adding metadata to an information system increases its presence in search engines. The implementation of structured vocabularies to control the content of these metadata and to organize information allows achieving a more effective organization of the collections that will lead to a recovery of more relevant information.

The development of meta information structures should be combined with the power of computer processing.

Both tendencies would converge in the approach of the Semantic Web, in which an intelligent search that takes advantage of the structured knowledge and the added human value embedded into knowledge structures is pursued.

REFERENCES

[1]. **Davies, Paul Beynon.** Sistemas de informacion. Barcelona,España : Reverte, 2014, pág. 636.

[2]. **w3c.** Estandares de la web semantica. [En línea] http://www.w3.org/standards/semanticweb/.

[3]. **Heath, Tom.** Linked Data. [En línea] [Citado el: 27 de 03 de 2016.] http://linkeddata.org/.

[4]. Descripción de los recursos de información en Internet: formato Dublín Core. **Founier, Isabel Daudinot.** 2006, SciElo.

[5]. RQ-RDF-3X: Going Beyond Triplestores. **Jyoti, Leeka.** Chicago IL : IEEE, 2014. 5th International Workshop on Data Engineering Meets the Semantic Web. pág. 6.

[6]. Actualización incremental de grafos RDF a partir de bases de datos relacionales. **Álvarez, Liudmila Reyes.** Cádiz : Javier Tuya, Mercedes Ruiz y Nuria Hurtado, 2014. Actas de las XIX Jornadas de Ingeniería del Software y Bases de Datos. pág. 6.

[7]. Desarrollo Orientado a la Semántica. **Pacheco, Ebenezer Hasdaí.** 2015, Develop Network No. 6.

[8]. Vocabularios estructurados , Web Semántica y Linked Data : oportunidades y retos para los profesionales de la documentación. **Castro, Carmen Caro.** 2012.

[9]. Diez años construyendo una web semántica. **Schorlemmer, Marco.** 2011, Fundación General CSIC.

[10]. **Priya.L.** Improving E-learning System using Ontology Web Language. International Journal of Modern ngineering Research (IJMER). Jan-Feb de 2012, pág. 5.

[11]. RDF Object Type and Reification in the Database. **Ravada, Nicole Alexander and Siva.** 2006 , Proceedings of the 22nd International Conference on Data Engineering (ICDE'06) 8-7695-2570-9/06 $20.00 © IEEE.

[12]. **w3c.** RDF conceps. [En línea] 25 de febrero de 2014. http://www.w3.org/TR/rdf11-concepts/.

[13]. **Initiative, Dublin Core Metadata.** Dublin Core Metadata Initiative. dublincore. [En línea] 23 de septiembre de 2016. [Citado el: 20 de ocubre de 2016.] http://dublincore.org/metadata-basics/.

[14]. **López, Sergio Fernández.** TRIOO, estudio e implementación de modelos de datos RDF en lenguajes orientados a objetos. 2010.

[15]. Integration of Heterogeneous Relational Databases: RDF Mapping Approach. **Ismail, Maizatul Akmar.** 2008, IEEE, pág. 7.

[16]. **Oracle java.** Annotations Basics. [En línea] [Citado el: 2016 de 10 de 23.] https://docs.oracle.com/javase/tutorial/java/annotations/basics.html.

[17]. **Dittawit, Kornschnok.** A Linked Data Model for E-books. 2012 IIAI International Conference on Advanced Applied Informatics . s.l. : IEEE, 2012, pág. 5.