2 Background

In order to make the vast amounts of data stored in Relational databases (RDBs) more easily available to the Semantic Web, a connection must be established between RDBs and a format suitable for the Web. Data is being published as CSV data dumps, Excel spreadsheets, and in a multitude of domain-specific data formats. Structured data is embedded into HTML pages using Microformats[2] . Various data providers have started to allow direct access to their databases via Web APIs.

The Semantic Web community has been recently exploring the uses of RDF as a means to facilitate integration between separate Relational databases and to enhance the semantics of published relational data. RDF provides [Heath & Bizer 2011]:

– **A unifying data model.** By providing for the globally unique identification of entities and by allowing different schemata to be used in parallel to represent data, the RDF data model has been especially designed for the use case of global data sharing. In contrast, other methods for publishing data on the Web rely on a large variety of different data models, and the resulting heterogeneity needs to be bridged in the integration process (e.g Web APIs).

– **A standardized data access mechanism.** RDF commits itself to a specific pattern of using the HTTP protocol. This agreement allows data sources to be accessed by generic data browsers and enables the complete data space to be crawled by search engines. In contrast, Web APIs are accessed using different proprietary interfaces.

– **Hyperlink-based data discovery.** By using URIs as global identifiers for entities, Linked Data allows hyperlinks to be set between entities in different data sources. These data links connect all Linked Data, creating a single global data space, and enable Linked Data applications to discover new data sources at run-time. In contrast, Web APIs as well as data dumps in proprietary formats remain isolated data silos.

[2]http://microformats.org/

– **Self-descriptive data.** Linked Data eases the integration of data from different sources by relying on shared vocabularies, making the definitions of these vocabularies retrievable, and by allowing terms from different vocabularies to be connected to each other by vocabulary links.

2.1 Ontology

Ontologies are the building blocks of the Semantic Web. Guarino [Guarino et al. 1998] organized a list of definitions for the term ontology, summarized as follows:

1. Ontology is a philosophy discipline
2. Ontology is an informal conceptualization of a system
3. Ontology is a formal semantic description
4. Ontology is a specification of a conceptualization.
5. Ontology is the representation of a conceptual system using some logic theory.
6. Ontology is a vocabulary used by some logic theory.
7. Ontology represents the specification metalevel of some logic theory.

One of the earliest definitions for the term ontology comes from the field of Philosophy, where the term is defined as “the study of being or existence”. This definition forms the basic subject matter of metaphysics. As a philosophical discipline, ontology building is concerned with describing or providing category systems that account for a certain vision of the world.

Gruber [Gruber et al. 1993] formulated the definition of ontology most frequently quoted in the Semantic Web literature

> *“An ontology is a formal, explicit specification of a shared conceptualization”.*

According to [Breitman et al. 2006] conceptualization stands for an abstract model; explicit means that the elements must be clearly defined; and formal indicates that the specification should be machine processable. Going further, the authors stated that in Gruber’s view, an ontology is the representation of the knowledge of a domain, where a set of objects and their relationships is described by a vocabulary.

According to [Heflin 2004].

"An ontology defines the terms used to describe and represent an area of knowledge. Ontologies are used by people, databases, and applications that need to share domain information (a domain is just a specific subject area or area of knowledge, like medicine, tool manufacturing, real estate, automobile repair, financial management, etc.). Ontologies include computer-usable definitions of basic concepts in the domain and the relationships among them (note that here and throughout this document, definition is not used in the technical sense understood by logicians). They encode knowledge in a domain and also knowledge that spans domains. In this way, they make that knowledge reusable.

The word ontology has been used to describe artifacts with different degrees of structure. These range from simple taxonomies (such as the Yahoo hierarchy), to metadata schemes (such as the Dublin Core), to logical theories. The Semantic Web needs ontologies with a significant degree of structure, capable of specifying descriptions for the following kinds of concepts:

- *Classes (general things) in the many domains of interest*
- *The relationships that can exist among things*
- *The properties (or attributes) those things may have"*

2.2 RDF

The Resource Description Framework (RDF) is the basic framework upon which ontologies in Semantic Web are based. RDF provides a mechanism that allows anyone to make basic statements about anything and layering these statements into a single model.

According to [Manola & Miller 2004] the Resource Description Framework (RDF) is a language for representing information about resources in the World Wide Web. It was particularly intended for representing metadata about Web resources, such as the title, author, and modification date of a Web page. If we generalize the concept of a "Web resource", RDF can also be used to represent information about things that can be identified on the Web, even when they cannot be directly retrieved on the Web. Examples include information about items available from on-line shopping facilities (e.g., information about specifications, prices, and availability), or the description of a Web user's preferences for information delivery.

RDF is intended for situations where this information needs to be processed by applications, rather than being displayed for people only. RDF provides a common framework for expressing information on the Web so it can be exchanged between software applications [Manola & Miller 2004]. Since it is a common framework, application designers can leverage from the availability of common RDF parsers and processing tools. The ability to exchange information between different applications means that the information may be made available to applications other than those for which it they originally created for.

RDF is based on the idea of identifying things using Web identifiers (called Uniform Resource Identifiers, or URIs[3]), and describing resources in terms of simple properties and property values.

We refer to the things in the world as resources; a resource can be anything that someone might want to describe. In that way, the RDF language is based upon the idea of making statements about resources in the form of | *subject* | *predicate* | *object* | expressions. These expressions are known as RDF triples, where subject is the part that identifies the thing (following the grammatical notion that where the subject is the thing that a statement is about). The predicate is the part that identifies the property or characteristic of the subject of the statement. The value of the property is called the object. For instance, for the following english statement:

http://www.inf.puc-rio.br/psalas has a creator whose value is Percy Salas

Could be written as a RDF triple with the following values:

– Subject: URL http://www.inf.puc-rio.br/ psalas
– Predicate: “creator”
– Object: “Percy Salas”

When more than one triple refers to the same thing, sometimes it is convenient to view the triples as a directed graph. In such a graph each triple is an edge from the subject to the object, with the predicate as the label, as shown in Figure 1.

The following example, borrowed from [Manola & Miller 2004], provides and RDF representation for the following statements “there is a Person identified by http://www.w3.org/People/EM/contact#me, whose name is Eric Miller, whose email address is em@w3.org, and whose title is Dr.” The example is illustrated by the RDF graph in Figure 1.
The example illustrates that RDF uses URIs to identify:

[3]http://www.ietf.org/rfc/rfc2396.txt

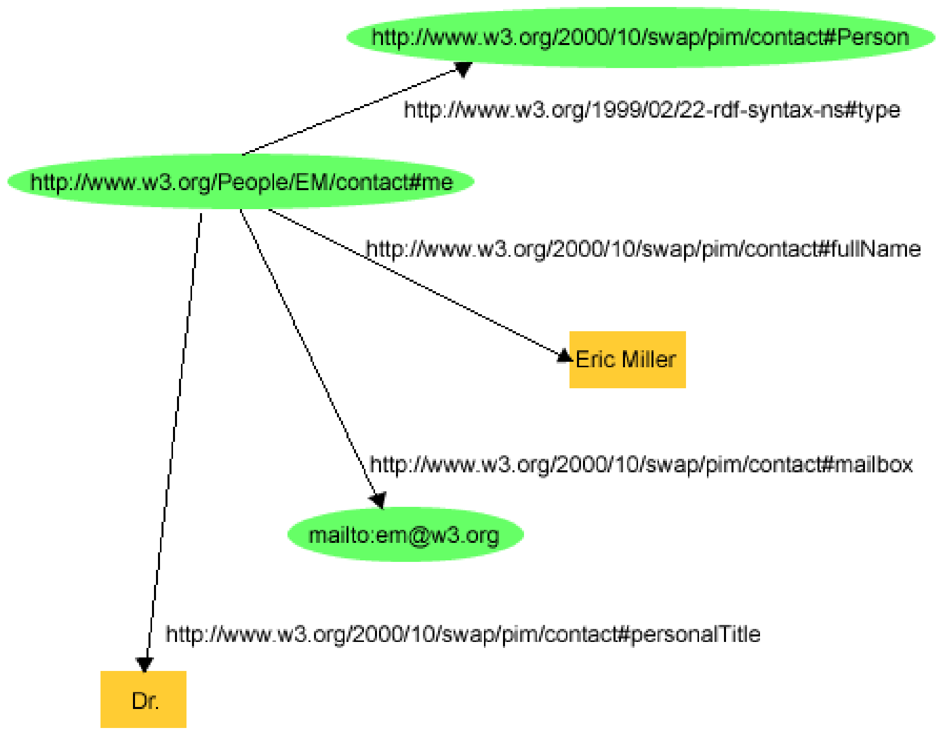
http://www.w3.org/2000/10/swap/pim/contact#Person
http://www.w3.org/1999/02/22-rdf-syntax-ns#type
http://www.w3.org/People/EM/contact#me
http://www.w3.org/2000/10/swap/pim/contact#fullName
Eric Miller
http://www.w3.org/2000/10/swap/pim/contact#mailbox
mailto:em@w3.org
http://www.w3.org/2000/10/swap/pim/contact#personalTitle
Dr.

Figure 1: An RDF Graph Describing Eric Miller [Manola & Miller 2004].

- Individuals, e.g., Eric Miller, identified by http://www.w3.org/People/EM/contact#me
- Kinds of things, e.g., Person, identified by http://www.w3.org/2000/10/swap/pim/contact#Person
- Properties of those things, e.g., mailbox, identified by http://www.w3.org/2000/10/swap/pim/contact#mailbox
- Values of those properties, e.g. mailto:em@w3.org as the value of the mailbox property (RDF also uses character strings such as “Eric Miller”, and values from other datatypes such as integers and dates, as the values of properties)

RDF provides a flexible way to describe both information and non-information resources, and how they relate to other things. The statements of relationships between things are, in essence, links connecting things in the world. RDF enables us to publish information on the Web in a form that others can discover and reuse. The key features and benefits of RDF, according to [Heath & Bizer 2011] are the following:

- RDF links things, not just documents. The data model enables you to set RDF links between data from different sources.

- RDF links are typed: HTML links typically indicate that two documents are related in some way, but mostly leave the user to infer the nature of the relationship. In contrast, RDF enables the data publisher to state explicitly the nature of the connection.
- By using HTTP URIs as globally unique identifiers for data items. The RDF data model is inherently designed for being used at global scale and enables anybody to refer to anything.
- RDF allows you to represent information that is expressed using different schemata in a single graph, meaning that you can mix terms for different vocabularies to represent data.
- Clients can look up every URI in an RDF graph over the Web to retrieve additional information.
- RDF allows you to use as much or as little structure as you need, meaning that you can represent tightly structured data as well as semi-structured data.

RDF enables Web publishers to make the hyperlinks explicit, and in such a way that RDF-aware applications can follow them to discover more data [Heath & Bizer 2011]. We assume, that if data is both published and linked using RDF it will be significantly more discoverable, and therefore more usable.

2.3 RDF Standard Vocabularies

Vocabularies provide domain-specific terms for describing classes of things in the world, and how they relate to each other. Depending on their power of expression vocabularies can be classified from taxonomies to ontologies [McGuinness 2002].

RDFS and OWL provide vocabularies for describing conceptual models in terms of classes and their properties. For instance, someone may define an RDFS vocabulary about pets that includes a class *Dog*, of which all individual dogs are members. They may also define a property *hasColour*, thereby allowing people to publish RDF descriptions of their own dogs using these terms.

Depending on the domain, communities have specific vocabularies preferences in which to publish their data. The Web of Data therefore, comprises several arbitrary vocabularies that are used in parallel.

According to [Heath & Bizer 2011], it is considered a good practice to reuse terms from well-known RDF vocabularies whenever possible. If the

adequate terms are found in existing vocabularies, these should be reused to describe data. Reuse of existing terms is highly desirable, as it maximizes the probability of the data being consumed by applications tuned to well-known vocabularies, without further processing or modifying the application.
The vocabularies in the following list cover a widespread set of domains and are used by a very large community. To ensure interoperability, it is recommended that these vocabularies are reused wherever possible [Bizer et al. 2007].

– The **Dublin Core Metadata Initiative (DCMI)**[4] Metadata Terms vocabulary defines general metadata attributes such as title, creator, date and subject.
– The **Friend-of-a-Friend (FOAF)**[5] vocabulary defines terms for describing people, their activities and their relations to other people and objects.
– The **Semantically-Interlinked Online Communities (SIOC)**[6] vocabulary (pronounced "shock") is designed for describing aspects of online community sites, such as users, posts and forums.
– The **Description of a Project (DOAP)**[7] vocabulary (pronounced "dope") defines terms for describing software projects, particularly those that are Open Source.
– The **Music Ontology**[8] defines terms for describing various aspects related to music, such as artists, albums, tracks, performances and arrangements.
– The **Programmes Ontology**[9] defines terms for describing programmes such as TV and radio broadcasts.
– The **Good Relations Ontology**[10] defines terms for describing products, services and other aspects relevant to e-commerce applications.
– The **Creative Commons (CC)**[11] schema defines terms for describing copyright licenses in RDF.
– The **Bibliographic Ontology (BIBO)**[12] provides concepts and properties for describing citations and bibliographic references (i.e., quotes, books, articles, etc.).

[4]http://dublincore.org/documents/dcmi-terms/
[5]http://xmlns.com/foaf/spec/
[6]http://rdfs.org/sioc/spec/
[7]http://trac.usefulinc.com/doap
[8]http://musicontology.com/
[9]http://purl.org/ontology/po/
[10]http://purl.org/goodrelations/
[11]http://creativecommons.org/ns#
[12]http://bibliontology.com/

- The **OAI Object Reuse and Exchange**[13] vocabulary is used by various library and publication data sources to represent resource aggregations such as different editions of a document or its internal structure.
- The **Review Vocabulary**[14] provides a vocabulary for representing reviews and ratings, as are often applied to products and services.
- The **Basic Geo (WGS84)**[15] vocabulary defines terms such as lat and long for describing geographically-located things.

Bizer and others argues that, only if none of these vocabularies provide the required terms, the data publishers should define new — data source-specific — terminology [Bizer et al. 2007]. W3C provides a set of guidelines to help users in publishing new vocabularies, such as:

> *"if new terminology is defined, it should be made self-describing by making the URIs that identify terms Web dereferencable. This allows clients to retrieve RDF Schema or OWL definitions of the terms as well as term mappings to other vocabularies"* [Berrueta & Phipps 2008].

2.4 Vocabulary Reuse

Generally, searching for an adequate term to describe a relationship, item or domain is a difficult task. Mostly because there is no central authority, ontology repository or Internet-based directory that can be consulted on such purpose. Existing databases and services, e.g., Schemapedia[16] , Watson[17] , SchemaWeb[18] , SchemaCache[19] , and Swoogle[20] provide a list of suggestion terms to help guide users in this process. Further insights into patterns and vocabulary usage pragmatics can be obtained from the vocabulary usage statistics provided by [Kinsella et al. 2008]. The State of the LOD Cloud document[21] also provides useful statistical data on RDF vocabulary usage.

Probably part of the difficulty of finding adequate RDF vocabularies comes from the fact that there is no widely accepted methodology for evaluating ontologies. This evaluation could be useful in tagging a vocabulary as recommended for further integration into a ontology repository.

[13]http://www.openarchives.org/ore/
[14]http://purl.org/stuff/rev#
[15]http://www.w3.org/2003/01/geo/
[16]http://schemapedia.com/
[17]http://kmi-web05.open.ac.uk/WatsonWUI/
[18]http://www.schemaweb.info/
[19]http://schemacache.com/
[20]http://swoogle.umbc.edu/
[21]http://lod-cloud.net/state#terms

Zimmermann in [Zimmermann 2010] proposes a set of features that a Web terminology should meet in order to be considered a quality vocabulary. The proposed criteria takes into consideration:

– **Justifying the existence of the vocabulary:** A vocabulary should be accompanied by a statement about its utility (description, scope, use cases); supported by at least some data publishers

– **Ease of reuse and publication:** Should be understandable by non-ontology experts, presence of clear labels and textual description

– **Interoperability:** Published in Semantic Web Standard format, vocabularies should follow best practices of Linked Data [Bizer et al. 2007], desirable to enable interoperability of the vocabulary with both OWL and RDFS.

According to [Heath & Bizer 2011] the following criteria should be applied in order to select vocabularies for reuse:

– **Usage and uptake:** is the vocabulary in widespread usage? Will the use of this vocabulary make a data set more or less accessible to existing Linked Data applications?

– **Maintenance and governance:** is the vocabulary actively maintained according to a clear governance process? When, and on what basis, are updates made?

– **Coverage:** does the vocabulary cover enough of the data set to justify adopting its terms and ontological commitments ?

– **Expressivity:** is the degree of expressivity in the vocabulary appropriate to the data set and application scenario? Is it too expressive, or not expressive enough?

2.5 Publishing New Vocabularies

When the existing vocabularies do not cover all the classes and properties one needs to describe, it is necessary to define new terms in a separate new vocabulary.

RDF classes and properties are resources themselves, identified by URIs, and published on the Web. In this sense, the following guidelines, adapted from [Heath & Bizer 2011], should be taken into consideration when defining new vocabularies:

1. Complement existing vocabularies rather than reinventing or introducing new terms.

2. Only define new terms in a namespace that you control.
3. Use terms from RDFS and OWL to relate new terms to those in existing vocabularies.
4. URIs of terms should be dereferenceable so that other applications can look up their definition [Berrueta & Phipps 2008].
5. Document each new term with human-friendly labels and comments – rdfs:label and rdfs:comment are designed for this purpose.
6. State all important information explicitly. For example, state all ranges and domains explicitly. Remember: humans can often do guesswork, but machines can't. Don't leave important information out!
7. Only define things that matter: for example, defining domains and ranges helps to clarify how properties should be used, but over-specifying a vocabulary can also produce unexpected inferences when the data is consumed. Thus, you should not overload vocabularies with ontological axioms, but better define terms rather loosely (for instance, by using only the RDFS and OWL terms introduced above).

2.6 Linked Data

Bizer and others [Bizer et al. 2009] define Linked Data as simply using the Web to create typed links, using Semantic Web technology (e.g. RDF) to connect data from different sources.

> *"The Semantic Web isn't just about putting data on the web. It is about making links, so that a person or machine can explore the web of data. With linked data, when you have some of it, you can find other, related, data. Like the web of hypertext, the web of data is constructed with documents on the web. However, in the web of hypertext, links are relationships anchors in hypertext documents written in HTML, while in the web of data the links are between arbitrary things described by RDF"* [Berners-Lee 2007].

The quote above was extracted from the "Linked Data" Web architecture key note in which Tim Berners-Lee outlines a set of best practices for publishing and interlinking structured data on the Web. Those include the following heuristics:

1. Use Uniform Resource Identifiers (URIs) as names for things.

2. Use HTTP URIs, so that people can look up those names in the Web.

3. When someone looks up a URI, provide useful information, using the Semantic Web standards (RDF, RDFS).

4. Include links to other URIs, so that they can discover more things.

The above items have become known as the “Linked Data principles”, and provide a basic recipe for publishing and connecting data using the infrastructure of the Web, while adhering to its architecture and standards.

This takes into consideration that the goal of Linked Data is to enable people to share structured data on the Web, as easily as they share documents today in the document Web [Heath & Bizer 2011]. The Linked Data approach takes advantage of the general architecture of the World Wide Web [Jacobs & Walsh 2004] and enables sharing structured data on a global scale.

According to [Jacobs & Walsh 2004] the document Web is built on a small set of simple standards: Uniform Resource Identifiers (URIs) — as the globally unique identification mechanism —, the Hypertext Transfer Protocol (HTTP) — as the universal access mechanism —, and the Hypertext Markup Language (HTML) — as widely used content format. In addition, the Web is built on the idea of setting hyperlinks between Web documents that may reside on different Web servers.

The development and use of standards enables the Web to transcend different technical architectures. Hyperlinks enable users to navigate between different servers and connect content from them into a single global information space. In that way we can visualize the Web as one huge database.

2.7 Mapping Relational Databases to RDF (RDB-to-RDF)

The Web of Data is constantly growing, mostly due to its compelling potential to facilitating data integration and retrieval. Nevertheless, there is still a significant gap, if one compares the growth rates of the Web to those of the Semantic Web. The main reason is that most of existing websites (over 70%) derive their data from relational databases (RDB) [He et al. 2007]. Given the astounding amount of data stored in relational databases, a critical requirement for the evolution of the Semantic Web is the publication of such data. In order to make this huge amount of relational data available to the Web of Data, a connection between RDBs and a format suitable for the Web of Data must be established. The relational database to RDF (RDB-to-RDF) approach emerges as a solution that makes it possible to map relational data to the Resource Description Framework (RDF) format.

In “Relational Databases on the Semantic Web”, Tim Berners Lee discusses common and distinct characteristics of RDF and the Relation Database Model. He identified an interesting correlation between Semantic Web and Relation Databases transcripted as follows.

> *“A relational database consists of tables, which consists of rows, or records. Each record consists of a set of fields. The record is nothing but the content of its fields, just as an RDF node is nothing but the connections: the property values. The mapping is very direct:*
>
> – *A record is an RDF node;*
> – *The field (column) name is RDF propertyType; and*
> – *The record field (table cell) is a value.*
>
> *Indeed, one of the main driving forces for the Semantic web, has always been the expression, on the Web, of the vast amount of relational database information in a way that can be processed by machines.”* [Berners-Lee 1998b]

The publication of Relational Databases as RDF is known as the RDB-to-RDF approach. In the RDB-to-RDF approach, a mapping takes as input a Relational Database (schema and data) and produces one or more RDF graphs as the output [Prud’hommeaux & Hausenblas 2010], as depicted in Figure 2.

The RDB-to-RDF approaches may be classified into two main categories. The first category is “Direct mapping” and its goal is to create a new ontology from the input relational database schema. This approach is easy to automate because it involves direct transformations rules, such as tables-to-classes and columns-to-properties.

The second category is know as “Domain ontology mapping” and its goal is to map the database schema to an existing domain ontology. The idea is to find a series of correspondences between this ontology and the relational database input. The mapping process here is not direct, as in the previous case. In this approach human intervention is necessary to aid the mapping process. Human intervention is needed because the modeling criteria used for designing databases are different from those used for designing ontology models [Barrasa et al. 2004].

According to [Ghawi & Cullot 2007] both mapping approaches above include the following two processes:

1. **Mapping definition:** The transformation of database schema to ontology structure.

2. **Data migration:** The migration of database contents into ontology instances. This migration can be done in two ways [Rodriguez & Gómez-Pérez 2006]:

   – **By materializing:** As a batch process by dumping all database instances in the ontology repository.
   – **By virtual access:** As an interface that queries the underlying database and transforms the database response to a given query, i.e. only the data needed to answer the user's query is retrieved from the sources.

The mapping definition is a major step in the RDB-to-RDF process, and consist in deciding how to represent database schema concepts in terms of RDF classes and properties. This is represented by the RDB-to-RDF mapping file, in a specific language and format, which is used as the base for the RDF triples generation.

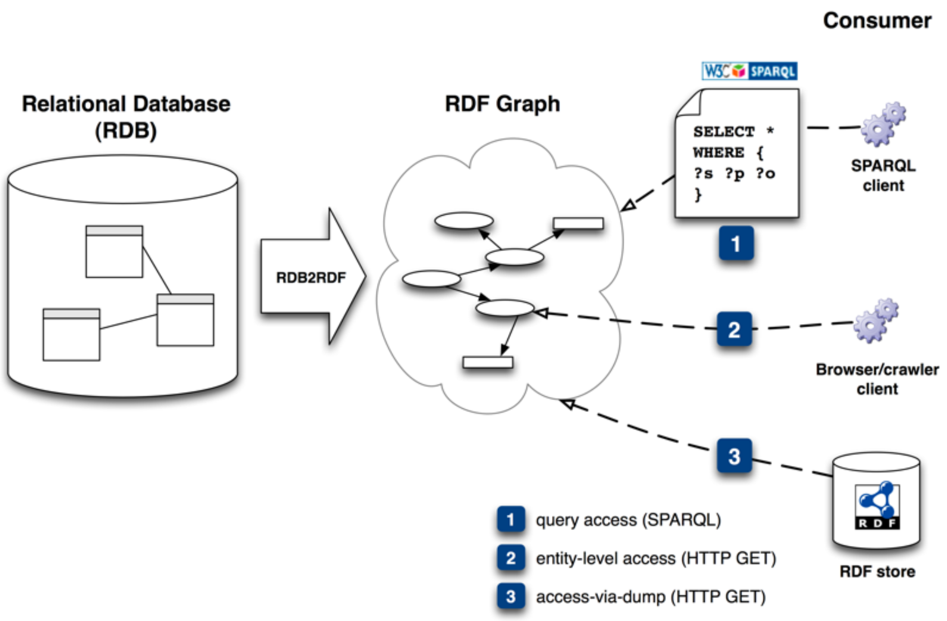

Figure 2: RDB-to-RDF Mapping Process

The consumer of the RDF Graph (virtual or materialized) can access the RDF data in three different ways [Prud'hommeaux & Hausenblas 2010]:

– Query access: the agent issues a SPARQL query against an endpoint exposed by the system, receives and processes the results (typically the result is a SPARQL result set in XML or JSON);

– Entity-level access: the agent performs an HTTP GET on a URI exposed by the system, and processes the result (typically the result is an RDF graph);

– Dump access: the agent performs an HTTP GET on dump of the entire RDF graph, for example in Extract, Transform, and Load (ETL) processes.

2.8 Ontology Matching

Ontology adoption, by itself, is not sufficient to secure interoperability. More often than not, the process of conceptual model alignment is done using the ontological representation, in OWL, as opposed to directly in RDF, to capitalize from previous work in ontology alignment [Euzenat & Shvaiko 2007].

In a distributed and open system, such as the semantic web, heterogeneity cannot be avoided. Different parties would, in general, adopt different ontologies to describe an specific domain of interest. Thus, merely using ontologies does not reduce heterogeneity: it raises heterogeneity problems to a higher level.

Ontology alignment is the problem of finding the semantic mappings between two given ontologies. This problem is closely related to the schema matching problem, as both schemata and ontologies provide a vocabulary of terms to describes a domain of interest.

Schema matching is considered a fundamental operation in the manipulation of database schemata. It consists of taking two schemata as input and producing a mapping between pairs of elements that are semantically equivalent [Rahm & Bernstein 2001].

According to [Casanova et al. 2007] the matching approaches may be classified as syntactic vs. semantic and, orthogonally, as *a priori* vs. a posteriori. In order to clarify the differences between these approaches, we borrow an example from [Casanova et al. 2007].

2.8.1 Syntactic Approach

This approach involves matching two schemata based on syntactical hints, only considering schema information, such as attribute names, descriptions, data types, relationship types, constraints, and schema structure. It assumes that syntactical proximity implies semantic similarity.

For example, consider two schemas S and T that describe databases whose application domains aren't entirely clear. Assume that S has a set of objects named Games, with attributes Name and ESRB (Entertainment Software Rating Board), and T has a set of objects named Gaming, with attributes Name, Price, and Rating, as shown in Figure 3. Using only syntactical similarity, Games would probably match with Gaming, and the Name attributes would definitely match with each other, but ESRB would not match with Rating.

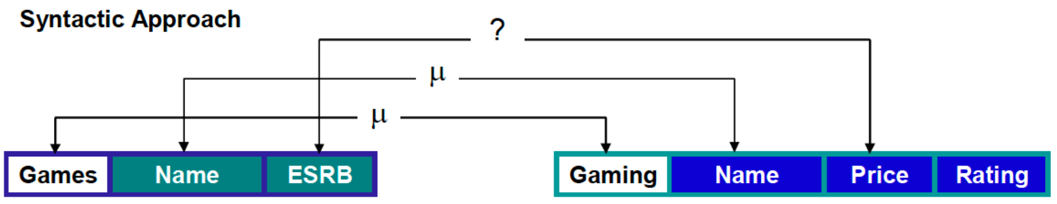

Figure 3: Syntatic Approach [Casanova et al. 2007].

If S and T describe computer game databases, this matching is reasonable, though it still misses the match between ESRB and Rating, which likely refers to ratings assigned by the ESRB. However, if S describes the database of a travel agency specializing in safaris, matching Games (meaning big game hunting) with Gaming (meaning computer games) is obviously inaccurate.

2.8.2 Semantic Approach

This approach uses semantic clues to generate hypotheses about schema matching such as data stored inside of tables. It is based on the assumption that data can provide important insight into the contents and meaning of schema elements [Rahm & Bernstein 2001]. This approach generally tries to detect how the same real-world objects are represented in two different datasets, and leverages on the information obtained to match different URIs.

The semantic approach is more robust than the syntactic one, but it seems to apply only when the schemas to be matched are simple. It also can help disambiguate between equally plausible candidate syntactic matches by choosing to match the elements whose instances are more similar.

Returning to the example of the schemas S and T with objects named Games and Gaming, respectively, the mediator might implement the procedure shown in Figure 4. It could select a few typical objects stored in Gaming, such as Flight Simulator and Super Mario Bros, and probe S to check whether they indeed occur in Games. The mediator could then use this information to match Games with Gaming, the Name attributes, and ESRB with Rating, as expected.

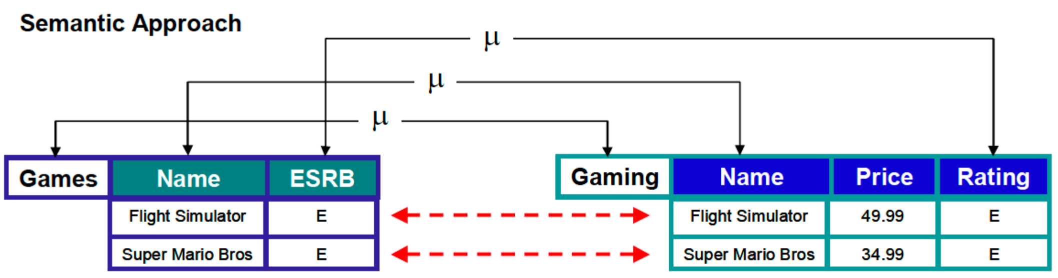

Figure 4: Semantic Approach [Casanova et al. 2007].

Note that this procedure would succeed when S and T both describe computer game databases, but not when S describes a travel agency database and T describes a computer game database. By overgeneralizing, the mediator might in fact completely ignore that Games and Gaming are syntactically similar and try to match S and T only using sets of typical objects.

2.8.3 *A priori* approach

Both syntactic and semantic approaches work a posteriori, in the sense that they start with existing datasets, and try to identify links between the two. [Casanova et al. 2007] propose a third alternative called the *a priori* approach, which specified that:

> *"When specifying databases that will interact with each other, the designer should first select an appropriate standard, if one exists, to guide design of the exported schemas. If none exists, the designer should publish a proposal for a common schema covering the application domain"*.

As defined by W3C ontologies can serve as the global schema or standard for the *a priori* approach. The authors in [Casanova et al. 2007] list the following steps to define a common schema for an application domain

- Select fragments of known, popular ontologies such as WordNet [22] that cover the concepts pertaining to the application domain;
- Align concepts from distinct fragments into unified concepts; and
- Publish the unified concepts as ontology, indicating which are mandatory and which are optional.

Matching two schemata that were designed according to the *a priori* approach, is an easier process as there is a consensus on the semantics of terminology used. Thus avoiding possible ambiguities.

[22]http://wordnet.princeton.edu/