

Mapping RDF and RDF-Schema to the Entity Relationship Model

¹Sikha Bagui, ²James Bouressa

¹Department of Computer Science, University of West Florida, Pensacola, FL 32514

²Special Operations Mission Planning Office, Ft Eustis, VA 23604

¹bagui@uwf.edu, ²jkbour78@yahoo.com

ABSTRACT

The management of Big Data and Big Semantic Data is becoming a challenge, especially since on the web, data comes from disparate data sources with different owners and different structures. In this paper, we discuss how to map RDF data on the web to a data visualization model, the Entity Relationship (ER) model.

Keywords: *RDF, RDF – Schema, ER model relational databases, mapping*

1. INTRODUCTION AND RELATED WORKS

Government, universities, digital libraries, researchers, innovation enterprises and other entities all create large amounts of data. Common activities such as adding friends on social networks, sharing photographs, electronic purchasing, etc., are continuously recorded in increasingly large data sets. Multiple sensors are constantly recording information [16]. The astronomical growth of the World Wide Web has resulted in data explosion, which in turn has given rise for the need for data representation methodologies and the management of Big Data and Big Semantic Data.

The use of graph-oriented representations and rich semantic vocabularies like RDF are gaining momentum for the management of Big Semantic Data [18]. Graphs provide a flexible way to integrate this data with different structures and enable heterogeneous data to be linked in a uniform way. The term, linked data, coined by Tim Berners Lee, refers to a style of publishing and interlinking structured web data [7, 12]. RDF, which can also be referred to as linked data, has become a standard model for data encoding and semantic technologies for publication, exchange, and consumption of this metadata and Big Semantic Data.

RDF's suitability to unstructured and semi-structured data, basically the flexibility of RDF, allowed for exponential growth of RDF data. This of course spun the need for tools and technologies for the management and visualization of this data.

[27] mention that one report suggests that internet accessible databases contain up to 500 times more data compared to static web and roughly 70% of the websites are backed by relational databases. This makes translation of RDF to relational databases central to the efficient management of the semantic web. The tools available for RDF data are fewer and less mature than the selection for relational databases [24]. Efficient storage and querying of RDF data is still an open problem [9, 24]. Rather than build RDF solutions from scratch, it is only logical to rely on relational representations of RDF [9, 24] to take advantage of the 35 plus years of research on efficient storage and querying, industrial strength transaction support, locking, security, etc. However,

given that RDF and relational models have differing architectures, there is a significant challenge in storing RDF in the relational model [9, 24]. In order to bridge this gap in architectures, in this paper we look at a mapping of RDF to a data modeling visualization paradigm, the Entity Relationship (ER) model. This data modeling visualization paradigm is the first step in visualizing the storage of data, which will eventually map to the relational model.

Several papers have discussed the mapping of relational databases to RDF [4, 8, 10, 11, 12, 15, 17, 18, 20, 23, 27, and 28]. These works make relational data accessible through RDF. [8] attempted to create RDF graphs from relational mappings. [15] used declarative meta schema consisting of quad map patterns to map SQL data to RDF ontologies. [17]'s work was in the spreadsheet domain and attempted to achieve richer spreadsheet to RDF translation. [4] extended SQL to publish linked data from RDF. [1, 2, 23] take a broader view. [23] used relational OWL to extract semantics of a relational database and map to RDF/OWL ontology. [1, 2] can work with a target ontology. [12] made an improvement over [24]'s D2R, demonstrating how to publish data in relational databases to RDF as linked data, making it browse-able. The W3C RDF2RDF Working Group is also developing a direct mapping standard that focuses on translating relational instances to RDF [3, 14]. [11] presented a relational database to RDF translation in a cultural heritage domain that will be accessible to semantic web tools. [27] mapped relational databases to an OWL ontology which then maps to RDF instances. [28] presented a series of RDB2RDF mapping patterns.

There have also been a few works that looked at mapping RDF to relational databases [22, 25, and 29]. [29] present a domain-specific mapping of the relational schema to RDF. [29]'s work requires the presence of ontological information to map RDFs to relational databases. However, [29]'s mappings involve the creation of a table for each property in RDF, regardless of cardinality; hence the resulting schema may not be fully normalized. [22] present a model that has similar drawbacks. [22] however, can map without ontological information. R2D does mapping through examination of triple patterns and relationships between resources. [25, 26] are an effort to make available relational tools to RDF

<http://www.cisjournal.org>

stores. This is done with the aid of a JDBC wrapper around RDF stores. This presents a relational view of the stores.

So, based on the fact that it is clear that there needs to be a clear way of translating relational data to RDF and vice versa, in our paper we focus on data visualization, specifically Chen's ER model (1976), as a first step to mapping RDF data to the relational model. There does not appear to be any papers that take this necessary approach.

In this paper, a case study format is used to present a logical mapping between the two (RDF/RDF-S and ER). We will highlight facets of both structures and provide a common language and symbology from which to present and describe the mappings.

A perfect one-to-one correspondence between the very general and extensible structure of RDF and the very detailed tenants of the ER structure is not always possible. We include a logical working solution, describe possible variations, discuss shortcomings and reflect on future work to include additional semantic constructs and the potential for a more enduring (evolutionary) standard promoting advanced web hosted data formats which will support a more seamless data integration and manipulation process.

The rest of this paper is organized as follows. Section 2 presents an overview of the ER model; section 3 presents an overview of RDF and RDF-Schema; in section 4 we discuss the RDF to ER mappings; in section 5 we present a complete example in RDF/XML and in section 6 we present the conclusion.

2. THE ER MODEL

In the Chen like ER model [13] information is represented in terms of entities, their attributes and relationships. Entities describe a complex structured concept like a person, place, thing or event of interest. Entities have keys. The primary key of an entity is used to uniquely identify a particular entity. Attributes are used to describe entities. Attributes can be either single value or multi-valued. And, relationships describe associations among entities. Relationships are explained in terms of their connectivity (or cardinality), and their connectivity can be indicated by one-to-one (1:1), one-to-many (1:M) and many-to-many (M:N) relationships. Cardinality is related to upper and lower bounds. Participation in this connectivity by member entities may be optional (partial) or mandatory (full).

We will use Chen-like ER diagrams, also used by [5] for our mappings. So, graphically we will use the following representations: entities will be shown in rectangles with the name of the entity inside the rectangle; single valued attributes will be shown in ovals with the attribute name inside the oval; attributes will be connected to the entity or relationship by a solid line; relationships will be shown in diamonds, connected by

lines to each of the participating entities; single lines between an entity and a relationship will indicate an optional relationship (this means that the entity may or may not participate in this relationship); double lines between an entity and a relationship will indicate a mandatory relationship between the entity and the relationship (this means that every instance of this entity participates in this relationship); and primary key attributes will be underlined.

3. THE RDF AND RDF-SCHEMA (RDF-S)

RDF is a language for representing information about resources in the World Wide Web. Taken as a whole, RDF is basically simple: a graph of nodes-and-arcs interpreted as statements about things identified by URI refs. This description is sourced principally from the W3C published standards RDF- CONCEPTS, RDF-PRIMER and RDF-SEMANTICS [6, 19, 21].

RDF uses Uniform Resource Identifiers (URIs) to describe resources in terms of simple properties and property values. This enables RDF to represent simple statements about resources as a graph of nodes and arcs representing the resources, and their properties and values. RDF's graph model is defined in [19]. In this notation, a statement is represented by:

- a node for the subject
- a node for the object
- an arc for the predicate, directed from the subject node to the object node.



Fig 1: Subject/Object

The simple RDF statement represented by the graph shown in Figure 1 is taken from the W3C Primer – Resource Description Framework:

`http://www.example.org/index.html` has a **creator** whose value is **John Smith**:

- subject `http://www.example.org/index.html`
- predicate `http://purl.org/dc/elements/1.1/creator`
- object `http://www.example.org/staffid/85740`

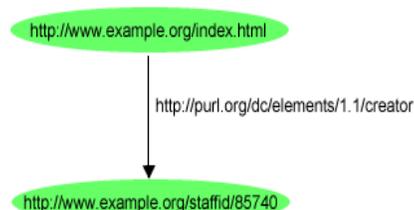


Fig 2: A Simple RDF Statement [21]

RDF, or linked data, is based on the idea that the things being described have properties, which have

<http://www.cisjournal.org>

values, and that resources can be described by making statements similar to those above that specify those properties and values. RDF uses a particular terminology for talking about the various parts of statements. Specifically, the part that identifies the thing the statement is about (the Web page, in this example) is called the subject. The part that identifies the property or characteristic of the subject that the statement specifies (creator, creation-date, or language, in these examples) is called the predicate, and the part that identifies the value of that property is called the object. So, taking the English statement:

<http://www.example.org/index.html> has a creator whose value is John Smith

The RDF terms for the various parts of the statement are:

- the subject is the URL <http://www.example.org/index.html>
- the predicate is the word "creator"
- the object is the phrase "John Smith"

Since RDF uses URI refs instead of words to name things in statements, RDF refers to a set of URI refs (particularly a set intended for a specific purpose) as a vocabulary.

URI refs are designed to be unique across the web and therefore reference a specific object. Each organization defines its own naming standards and context for the objects (e.g. products, people, etc.) it publishes on the web in RDF. It's incumbent upon any RDF application using a given organization's RDF data to understand the context established to correctly interpret or to make correct inferences on that data.

This illustrates one of the basic architectural principles of the Semantic Web, which is that anyone should be able to freely add information about an existing resource, using any vocabulary they please. This further illustrates that the RDF describing a particular resource does not need to be located in one place; instead, it may be distributed throughout the Web.

However, in addition to the basic techniques for describing things using RDF statements (triples representing RDF diagrams) discussed so far, it should be clear that people or organizations also need a way to describe the vocabularies (terms) they intend to use in those statements, specifically, vocabularies for:

- Describing types of things (like: Person)
- Describing properties (like: age and creation-date), and
- Describing the types of things that can serve as the subjects or objects of statements involving those properties (such as specifying that the value of an age property should always be specified by a Type: literal such as xsd integer).

- RDF itself does not provide facilities for defining application-specific classes of things, such as Person, or their properties, such as weight or age. Instead, such classes are described in the RDF schema.
- One important difference is that instead of describing a class as having a collection of specific properties, an RDF schema describes properties as applying to specific classes of resources, using domain and range properties. For example, a typical object-oriented programming language might define a class Book with an attribute called author having values of type Person. A corresponding RDF schema would describe a class Book, and, in a separate description, a property author having a domain of Book and a range of Person.
- In other words, statements in an RDF schema are always descriptions. They may also be prescriptive (introduce constraints), but only if the application interpreting those statements want to treat them that way.

A basic step in any kind of description process is identifying the various kinds of things to be described. RDF Schema refers to these "kinds of things" as classes. A class in RDF Schema corresponds to the generic concept of a Type or Category, somewhat like the notion of a class in object-oriented programming languages such as Java. RDF classes can be used to represent almost any category of thing, such as Web pages, people, document types, databases or abstract concepts. Classes are described using the RDF Schema resources `rdfs:Class` and `rdfs:Resource`, and the properties `rdf:type` and `rdfs:subClassOf` [21].

4. RDF TO THE ER MODEL

We will discuss the mapping of RDF to the ER model through a case study.

A Case Study: UWF Online University

The following simple case study of UWF Online University is used to present the conceptual mapping between RDF and the ER relational database structures. Figure 3 presents an ER diagram of UWF Online University. It depicts relationships and properties of three entities (objects). Our UWF Online University consists of Students, Instructors and Courses. We will limit the number of attributes (properties) for simplicity, but have included the various attributes types defined by the ER structure.

As per figure 3: We are storing information about Students, Instructors and Courses. For each student we are recording the student name (which is divided into first and last), student number (primary key), and major (or majors). For each course we are recording the course number (primary key), course name and credit hours. For each instructor we are recording the instructor number (primary key), instructor name (which is divided into first and last), and department.

<http://www.cisjournal.org>

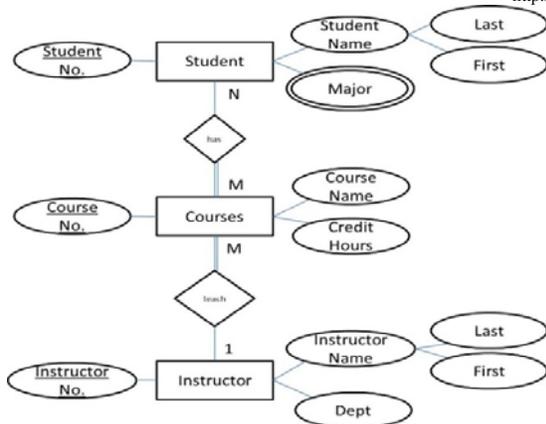


Fig 3: ER Diagram of UWF Online University

This UWF Online University ER Diagram (figure 3) can be represented as an RDF node diagram as depicted in Figure 4. The diagram is a generalized representation of RDF nodes and not a specific graph of RDF triples. It is intended as a bridge between the conceptual representation of data in an ER diagram and instance specific format of RDF triples. The Subject nodes and the Object nodes represent the ‘Class’ of things and their ‘attributes’ respectively, both described in RDF-S as classes (or subclasses). The Predicate shows the relationship between the two nodes. These classes and terms are defined by RDF vocabulary, in this case represented by the URI: ‘<http://UWFOnline.uwf.edu/COT6931/>’. This is a dereference-able URL, meaning there is a web page that can be looked up and read to determine the meaning of each term. Specific instances of these classes could then be defined in RDF triples with URIs referencing unique subjects and predicates with objects being defined by URIs or typed literals.

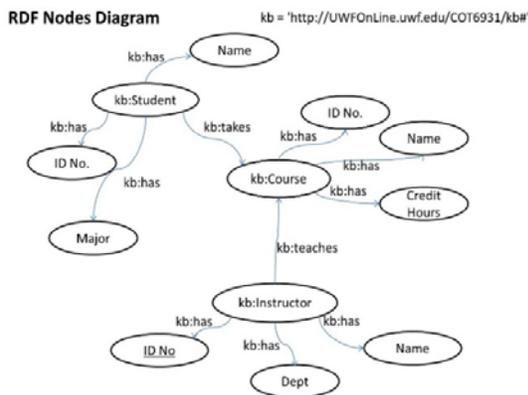


Fig 4: RDF Nodes Diagram of UWF Online University

4.1 Entities

In the ER model, entities are used to describe a complex structured concept like person, place, thing or event of interest. Entities group items or objects with similar characteristics (Bagui and Earp, 2012). In the ER model we are using, we will represent an entity in a rectangle.

In RDF Schema, a class or subclass describes a complex structured concept. A class is any resource having an rdf:type property whose value is the resource rdfs:Class. So the instructor class could be described by assigning the class a URI ref, for example, kb:Instructor (using kb: to stand for the URI ref 'http://UWFOnline.uwf.edu/COT6931/kb#', which is used as the prefix for URI refs from UWF Online vocabulary) and describing that resource with an rdf:type property whose value is the resource rdfs:Class. That is, UWF.edu would translate to the RDF statement:

```
<rdf:description rdf:about="kb:Instructor"
    rdfs:label="Instructor">
<rdfs:subClassOf rdf:resource="kb:Person"/>
</rdfs:Class>
```

Similarly, student would be another class or entity, as depicted in figure 5.

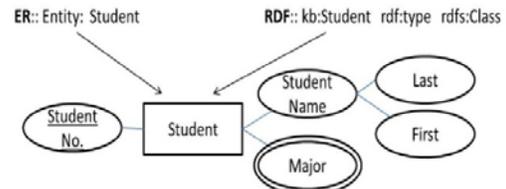


Fig 5: The Student entity or class

Figure 6 presents the corresponding RDF/XML for the entity, Student.

```
RDF/XML
<?xml version="1.0"?>
<!DOCTYPE rdf:RDF [<ENTITY xsd "http://www.w3.org/2001/XMLSchema#">]
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:kb="http://UWFOnline.uwf.edu/COT6931/kb#"
  >
  <rdf:Description rdf:ID="Students">
    <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#Class"/>
  </rdf:Description>
</rdf:RDF>
```

Fig 6: RDF/XML translation of the Student entity or class

Mapping Entities

A class or subclass from RDF-S translates to an entity in the ER model.

4.2 Attributes

In the ER model, attributes are used to describe entities. That is, they provide specific facts about individual entities. In the ER model, attributes can be simple, composite or multi-valued [5]. In the ER model, we will represent the attribute in an oval.

4.2.1 Simple/Atomic Attributes

Simple or atomic attributes cannot be subdivided further. In figure 7, Student_no would be an example of a simple attribute. These are of simple data types.

http://www.cisjournal.org

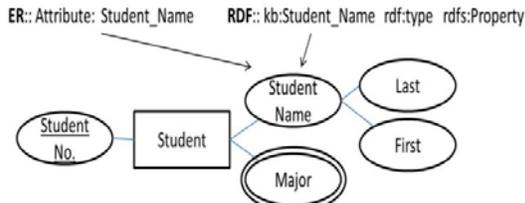


Fig 7: Student entity with attributes

In RDF Schema, properties are used to describe a class. Properties are described using the RDF class `rdf:Property`, and the RDF Schema properties `rdfs:domain`, `rdfs:range`, and `rdfs:subPropertyOf` [21].

Figure 8 presents the RDF/XML translation of the simple attribute, `Student_no`.

RDF/XML

```
<?xml version="1.0"?>
<!DOCTYPE rdf:RDF [
<!ENTITY xsd http://www.w3.org/2001/XMLSchema#>]
<rdf:RDF
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:rdfs="http://www.w3.org/2000/02/rdf-schema#"
xmlns:kb="http://UWFOOnline.uwf.edu/COP6931/kb#"
<rdf:Property rdf:ID="student_no">
<rdfs:domain rdfs:resource="#Student"/>
</rdf:Property>
</rdf:RDF>
```

Figure 8 RDF/XML translation of the simple attribute, `Student_no`

Mapping Simple Attributes

A property or sub Property of RDF/RDFS translates to a simple attribute in the ER model.

4.2.2 Composite Attributes

In the ER model, composite attributes are attributes that are broken up into parts. In figure 9, the composite attribute is `Student Name`, which has composite parts, `Last` and `First`.

One way of showing composite attributes in RDF would be through aggregates. Aggregates can be represented by several different methods (constructs) in RDF. RDF allows a Blank Node to be used locally to aggregate attributes. The Blank Node (Bnode) logically represents aggregate attributes. Multiple binary statements can assign each attribute to the blank node as individual properties.

For each composite relationship, one of the participants is chosen as the subject of the relationship (Jones, as shown in figure 9), and a blank node is created to represent the rest of the relationship (Sue, as shown in figure 9). Additional participants in the relationship (for example, if we had middle name) would then be represented as separate properties of the new resource represented by the blank node.

Composite attributes can also be mapped using constructs like containers. Containers define objects with associated members.

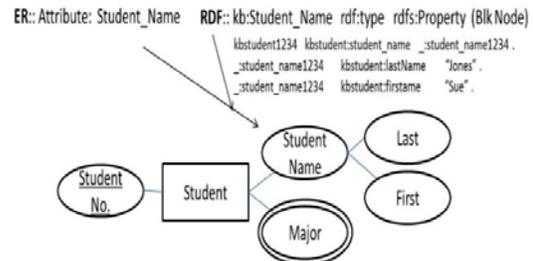


Fig 9: Composite attribute, `student_name`

Figure 10 presents the RDF/XML translation of a composite attribute using a Blank Node.

RDF/XML

```
<?xml version="1.0"?>
<!DOCTYPE rdf:RDF [
<!ENTITY xsd http://www.w3.org/2001/XMLSchema#>]
<rdf:RDF
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
xmlns:kb="http://UWFOOnline.uwf.edu/COT6931/kb#"
xmlns:exname="http://www.example.org/names/"
<rdf:Description rdf:about="http://UWFOOnline.uwf.edu/COT6931/student">
<kb:student_name rdf:parseType="Resource">
<rdf:value rdf:datatype="xsd:string">Jones</rdf:value>
<rdf:value rdf:datatype="xsd:string">Sue</rdf:value>
</kb:student_name>
</rdf:Description>
</rdf:RDF>
```

Fig 10: RDF/XML translation of a composite attribute

Mapping Composite Attributes

Composite attributes from the ER model can be mapped using Blank Nodes or Containers in RDF.

4.2.3 Multi-valued Attributes

In the ER model, attributes may be multi-valued, i.e. an entity may have more than one value for an attribute. For example, from figure 11, a student may have more than one major. Hence, `major` is multi-valued, showed with a double oval in the ER diagram.

Multi-valued attributes can be mapped as composite attributes are mapped, as discussed in the previous section, but the only issue would be that it would be difficult to distinguish between a composite attribute and a multi-valued attribute. Multi-valued attributes can also be mapped as binary relationships or using constructs like collections. Collections allow for a list of items.

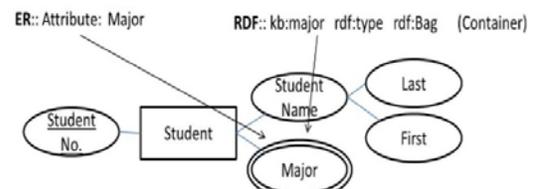


Fig 11: Multi-valued attribute, `major`

<http://www.cisjournal.org>

Figure 12 presents the RDF/XML of a multi-valued attribute using a bag(collection).

```

RDF/XML
<?xml version="1.0"?>
<!DOCTYPE rdf:RDF [<ENTITY xsd "http://www.w3.org/2001/XMLSchema#">]>
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:kb="http://UWFOnline.uwf.edu/COT6931/kb#"
  <rdf:Description rdf:about="http://UWFOnline.uwf.edu/COT6931/Majors">
    <kb:majors>
      <rdf:Bag>
        <rdf:li rdf:resource="http://UWFOnline.uwf.edu/COT6931/Majors/Computer Science"/>
        <rdf:li rdf:resource="http://UWFOnline.uwf.edu/COT6931/Majors/Math"/>
      </rdf:Bag>
    </kb:majors>
  </rdf:Description>
</rdf:RDF>

```

Fig 12: RDF/XML translation of multi-valued attribute, major

Mapping Multi-valued attributes

Multi-valued attributes from the ER model can be mapped using Collections or by availing of mapping binary relationships.

4.2.4 Keys

The concept of keys, a unique identifier for an entity is fundamental to the ER model. A primary key is an attribute that is used to uniquely identify an entity. In the ER model every instance has to have a primary key, and a primary key has the following characteristics: is a simple attribute; can have only one value; cannot be null; and has to be distinct in value.

RDF/RDFS does not directly support the concept of keys. RDF and RDF-S do not provide for constraints on data objects in the way the programming languages or relational database systems provide them. The benefit of RDF being able to recognize RDF constructs from across the web and combine them allows for interoperability but requires the RDF application to be aware of the context defined by each data source for useful implementations and inferences derived from the data. RDF can recognize its own constructs but cannot derive any meaning or direct association of the objects.

RDF requires objects (things) to be uniquely referenced by a URI. Each distinct URI is treated as a different object or thing; however, different sites may be describing the same object using different URIs. RDF-S does define an `rdfs:seeAlso` property which can point to another URI which may provide additional information about an object (subject) or that two or more URI references describe the same thing.

The `rdf:ID` attribute is a more constrained version of the `rdf:about` attribute. The `rdf:ID` attribute is somewhat similar to the `ID` attribute in XML, in that it defines a name which must be unique relative to the current base URI. Using `rdf:ID` enforces the assigning a set of distinct names, since a given value of the `rdf:ID` attribute can only appear once relative to the same base URI [21].

Assigning a key would start with identifying the RDF 'Subject' in each statement, verifying the URI is complete and therefore unique relative to each base URI. Combining multiple data sources (different base URIs) would require a translation process looking for common references. The `rdfs:seeAlso` property could be used to identify common objects.

Mapping Keys

`rdf:ID` would translate into a primary key in the ER model.

4.2.5 Mapping relationships

In the ER model, relationships are used to describe relationships or connections between entities. There can be one-to-one relationships, one-to-many relationships and many-to-many relationships. A one-to-one relationship is where one entity A is related to one and only one of the other entity B. A one-to-many relationship is where one entity A, is related to more than one entity B. A many-to-many relationship is where more than one entity A may be related to more than one entity B.

RDF directly represents only binary relationships and as such the RDF Schema provides basic capabilities for describing RDF vocabularies [21]. Additional capabilities are also possible, and can be useful. These capabilities may be provided through further development of RDF Schema, or in other languages based on RDF. Other useful schema capabilities have been identified and are only available in other languages.

Mapping relationships

A relationships in the ER model will have to be mapped using a combination of binary relationships in RDF.

5. A COMPLETE EXAMPLE IN RDF/XML (PRODUCES BY PROTÉGÉ V3.5)

Below is the RDF/XML equivalent of the ER diagram of UWFOnline University (figure 3).

RDF Vocabulary for UWF Online University:

```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE rdf:RDF [
  <ENTITY rdf "http://www.w3.org/1999/02/22-
rdf-syntax-ns#">
  <ENTITY kb "http://UWFOnline.uwf.edu/COT6931/kb#">
  <ENTITY rdfs "http://www.w3.org/2000/01/rdf-
schema#">
]>
<rdf:RDF xmlns:rdf="&rdf;"
  xmlns:kb="&kb;"
  xmlns:rdfs="&rdfs;">
<rdfs:Class rdf:about="&kb;Courses"
  rdfs:label="Courses">

```

http://www.cisjournal.org

```

    <rdfs:subClassOf rdfs:resource="&rdfs;Resource"
  />
</rdfs:Class>
<rdfs:Class rdfs:about="&kb;Instructor"
  rdfs:label="Instructor">
  <rdfs:subClassOf rdfs:resource="&kb;Person"/>
</rdfs:Class>
<rdfs:Class rdfs:about="&kb;Person"
  rdfs:label="Person">
  <rdfs:subClassOf rdfs:resource="&rdfs;Resource"
  />
</rdfs:Class>
<rdfs:Class rdfs:about="&kb;Student"
  rdfs:label="Student">
  <rdfs:subClassOf rdfs:resource="&kb;Person"/>
</rdfs:Class>
<rdf:Property rdfs:about="&kb;course_id"
  rdfs:label="course_id">
  <rdfs:domain rdfs:resource="&kb;Courses"/>
  <rdfs:rangerdfs:resource="&rdfs;Literal"/>
</rdf:Property>
<rdf:Property rdfs:about="&kb;credit_hrs"
  rdfs:label="credit_hrs">
  <rdfs:domain rdfs:resource="&kb;Courses"/>
  <rdfs:rangerdfs:resource="&rdfs;Literal"/>
</rdf:Property>
<rdf:Property rdfs:about="&kb;dept"
  rdfs:label="dept">
  <rdfs:domain rdfs:resource="&kb;Instructor"/>
  <rdfs:rangerdfs:resource="&rdfs;Literal"/>
</rdf:Property>
<rdf:Property rdfs:about="&kb;first_name"
  rdfs:label="first_name">
  <rdfs:domain rdfs:resource="&kb;Person"/>
  <rdfs:rangerdfs:resource="&rdfs;Literal"/>
</rdf:Property>
<rdf:Property rdfs:about="&kb;instructor_id"
  rdfs:label="instructor_id">
  <rdfs:domain rdfs:resource="&kb;Instructor"/>
  <rdfs:rangerdfs:resource="&rdfs;Literal"/>
</rdf:Property>
<rdf:Property rdfs:about="&kb;last_name"
  rdfs:label="last_name">
  <rdfs:domain rdfs:resource="&kb;Person"/>
  <rdfs:rangerdfs:resource="&rdfs;Literal"/>
</rdf:Property>
<rdf:Property rdfs:about="&kb;major"
  rdfs:label="major">
  <rdfs:domain rdfs:resource="&kb;Student"/>
  <rdfs:rangerdfs:resource="&rdfs;Literal"/>
</rdf:Property>
<rdf:Property rdfs:about="&kb;name"
  rdfs:label="name">
  <rdfs:domain rdfs:resource="&kb;Courses"/>
  <rdfs:rangerdfs:resource="&rdfs;Literal"/>
</rdf:Property>
<rdf:Property rdfs:about="&kb;student_id"
  rdfs:label="student_id">
  <rdfs:domain rdfs:resource="&kb;Student"/>
  <rdfs:rangerdfs:resource="&rdfs;Literal"/>
</rdf:Property>

```

```

  <rdf:Property rdfs:about="&kb;teaches_course"
  rdfs:label="teaches_course">
  <rdfs:rangerdfs:resource="&kb;Courses"/>
  <rdfs:domain rdfs:resource="&kb;Instructor"/>
</rdf:Property>
</rdf:RDF>

```

RDF/XML Data for UWFOne University:

```

<?xml version='1.0' encoding='UTF-8'?>
<!DOCTYPE rdf:RDF [
  <ENTITY rdf "http://www.w3.org/1999/02/22-
  rdf-syntax-ns#"
  <ENTITY kb "http://UWFOneLine.uwf.edu/COT6931/kb#"
  <ENTITY rdfs "http://www.w3.org/2000/01/rdf-
  schema#"
]>
<rdf:RDF xmlns:rdf="&rdfs;"
  xmlns:kb="&kb;"
  xmlns:rdfs="&rdfs;">
  <kb:Courses rdfs:about="&kb;UWF_OnLine_Program"
  rdfs:label="UWF OnLine Program"/>
  <kb:Student rdfs:about="&kb;KB_Students_1010"
  kb:first_name="Rex"
  kb:last_name="Blackford"
  kb:student_id="1010"
  rdfs:label="Blackford">
  <kb:major>English</kb:major>
  <kb:major>Music</kb:major>
</kb:Student>
  <kb:Instructor rdfs:about="&kb;KB_Instructor_9001"
  kb:dept="English"
  kb:first_name="Jackie"
  kb:instructor_id="9001"
  kb:last_name="Smith"
  rdfs:label="Smith">
  <kb:teaches_course rdfs:resource="&kb;KB_Cour-
  se_ENC_1101"/>
</kb:Instructor>
  <kb:Courses rdfs:about="&kb;KB_Course_ENC_1101"
  kb:course_id="ENC 1101"
  kb:credit_hrs="3"
  kb:name="English Composition I"
  rdfs:label="English Composition I"/>
</rdf:RDF>

```

6. CONCLUSION

We have presented a mapping of RDF to the Chen like ER model. We have shown how to map entities, attributes, keys and relationships. But, we found that the RDF Schema has some limitations with respect to the relational model, discussed below.

The following capabilities are not provided by RDF Schema directly:

- Cardinality constraints on properties, e.g., that a person has exactly one biological father.
- Specifying that a given property (such as ex:hasancestor) is transitive, e.g., that if

<http://www.cisjournal.org>

aex:hasancestor b, and b ex:hasancestor c, then aex:hasancestor c.

- Specifying that a given property is a unique identifier (or key) for instances of a particular class.
- Specifying that two different classes (having different urirefs) actually represent the same class.
- Specifying that two different instances (having different urirefs) actually represent the same individual.
- Specifying constraints on the range or cardinality of a property that depend on the class of resource to which a property is applied, e.g., being able to say that for a soccer team the ex:has players property has 11 values, while for a basketball team the same property should have only 5 values.
- The ability to describe new classes in terms of combinations (e.g., unions and intersections) of other classes, or to say that two classes are disjoint (i.e., that no resource is an instance of both classes) [21].

The additional capabilities mentioned above, are the targets of ontology languages such as DAML+OIL and OWL. Both these languages are based on RDF and RDF Schema (and both currently provide all the additional capabilities mentioned above). Our discussions on the mapping of the relationships is rather limited. This is mainly because relationships will be more completely mapped using the additional capabilities of DAML+OIL and OWL. Hence we have mapped as much as can be reasonably mapped from RDF to the ER model.

REFERENCES

- [1] Y. An, A. Borgida, and J. Mylopoulos, "Refining Semantic Mappings from Relational Tables to Ontologies", in Second International Workshop on Semantic Web and Databases, 2004, pp. 84-90.
- [2] Y. An, A. Borgida, and J. Mylopoulos, "discovering the semantics of relational tables through mappings", in Journal on Data Semantics, VII, 2006, pp. 1-32.
- [3] M. Arenas, A. Bertails, E. Preud'hommeaux, and J. Sequeda, "Direct mapping of relational data to RDF", in W3C Working Draft 29, May 2012, <http://www.w3.org/TR/2012/WD-rdb-direct-mapping-20120529/>.
- [4] S. Auer, S. Dietzold, J. Lehmann, S. Hellmann, and D. Aumeuller, "Triplify – Light-Weight Linked Data Publication from Relational Databases", in 18th International World Wide Web Conference, 2009, pp. 621-630.
- [5] S. Bagui, and R. Earp, Database Design Using Entity-Relationship Diagrams, 2nd edition, Francis and Taylor, Boca Raton, FL, 2012.
- [6] D. Beckett, "RDF/XML Syntax Specification (Revised)", in W3C Recommendation 10 February 2004 <http://www.w3.org/TR/rdf-syntax-grammar/>
- [7] T. Berners-Lee, "Linked Data", in <http://www.w3.org/DesignIssues/LinkedData.html>, 2009.
- [8] C. Bizer, R. Cyganiak, J. Garbers, and O. Maresch, "The D2RQ Platform", in <http://www4.wiwi.fu-berlin.de/bizer/d2rq>, 2012.
- [9] M. A. Bornea, J. Dolby, A. Kementsietsidis, K. Srinivas, P. Dantressangle, O. Udrea, and B. Bhattacharjee, "Building an Efficient RDF Store Over a Relational Database", in SIGMOD, June 22-27, 2013, New York, New York.
- [10] A. Borgida and J. Mylopoulos, "Data Semantics Revisited", in Proceedings of the 2nd International Conference on Semantic Web and Databases, 2004, pp. 9-26.
- [11] K. Byrne, "Relational Database to RDF Translation in the Cultural Heritage Domain", in Internet, 2008.
- [12] Y. Chen, X. Zhao, and S. Zhang, "Publishing RDF from Relational Database Based on D2R Improvement", in WSEAS Transactions on Information Science and Applications, Vol. 8, No. 10, 2013.
- [13] P. Chen, "The Entity Relationship Model – Towards a Unified View of Data", in ACM Trans. Database Systems, Vol. 1, No. 1, 1976, p. 9-36.
- [14] S. Das, S. Sundara, and R. Cyganiak, "R2xml:Rdb to rdf mapping language", in W3C Working Draft 29, May 2012, <http://www.w3.org/TR/2012/WD-r2rml-20120529/>.
- [15] O. Erling, and I. Mikhailov, "RDF Support in the Virtuoso DBMS", in 1st Conference on Social Semantic Web, 2007, pp. 1617-5468.
- [16] J. D. Fernandez, M. Arias, M. A. Martinez-Prieto, and C. Guierrez, "Management of Big Semantic Data", in Akerkar Rajendra, Big Data Computing, CRC Press, Boca Raton, Florida, 2014.
- [17] L. Han, T. Finin, C. Parr, J. Sachs, and A. Joshi, "RDF123: From Spreadsheets to RDF", in International Semantic Web Conference, LNCS 5318, 2008, pp. 451-466.
- [18] T. Hey, S. Tansley, and K.M. Tolle, "Jim Gray on eScience: A Transformed Scientific Method", in The Fourth Paradigm, Microsoft Research, 2009.

<http://www.cisjournal.org>

- [19] G. Klyne, and J. Carroll, "Resource Description Framework (RDF): Concepts and Abstract Syntax", in W3C Recommendation 10 February 2004, <http://www.w3.org/TR/rdf-concepts/>
- [20] P. C. Laborda, and S. Conrad, "Bringing Relational Data into the Semantic Web using SPARQL and Relational.Owl", 22nd International Conference on Data Engineering Workshop, 2006.
- [21] F. Manola, and E. Miller (eds.), "RDF Primer", in W3C Recommendation 10 February 2004, <http://www.w3.org/TR/rdf-primer/>
- [22] Z. Pan, and J. Heflin, "DLDB: Extending Relational Databases to Support Semantic Web Queries", in Practical and Scalable Semantic Systems, 2003.
- [23] C. Perez de Laborda and S. Conrad, "Bringing Relational Data into the Semantic Web Using SPARQL and Relational Owl", in 22nd International Conference on Data Engineering Workshops, 2006, pp. 55.
- [24] S. Ramanujan, A. Gupta, L. Khan, S. Seida and B. Thuraisingham, "R2D: A Framework for the Relational Transformation of RDF Data", in International Journal of Semantic Computing, World Scientific Press, 2009.
- [25] S. Ramanujan, A. Gupta, L. Khan, S. Seida, and B. Thuraisingham, "A Relational Wrapper for RDF Reification", in Proceedings of the Third IFIP WG 11.11, International Conference on Trust Management (IFIPTM), West Lafayette, USA, June 15-19, 2009.
- [26] S. Ramanujan, A. Gupta, L. Khan, S. Seida, and B. Thuraisingham, "Relationalizing RDF stores for tools reusability", in Proceedings of the 18th International Conference on World Wide Web, WWW, Madrid, Spain, April 20-24, 2009, pp. 1059-1060.
- [27] J. F. Sequeda, M. Arenas and D. P. Miranker, "On Directly Mapping Relational Databases to RDF and OWL", in Proceedings of the 21st International Conference on World Wide Web, 2012, pp. 649-658. ACM.
- [28] J. Sequeda, F. Priyatna and B. Villazon-Terrazas, "Relational Database to RDF Mapping Patterns", in Proceedings of the 3rd Workshop on Ontology Patterns, 2012.
- [29] W. Teswanichand S. Chittayasothorn, "A Transformation of RDF Documents and Schemas to Relational Databases", in IEEE PacificRim Conferences on Communications, Computers, and Signal Processing, 2007, pp. 38-41.

AUTHOR PROFILE

Dr. Sikha Bagui is Chair and Professor of the Department of Computer Science and Director for the Center for Cyber security at The University of West Florida. Dr. Bagui is active in publishing peer reviewed journal articles in the areas of database design, data mining, Big Data, pattern recognition, and statistical computing. Dr. Bagui has worked on funded as well unfunded research projects and has over 30 peer reviewed publications. She has also co-authored several books on database and SQL. Dr. Bagui also serves as Associate Editor and on the editorial board for several journals.

Mr. James Bouressa is the Configuration and Test Manager for the Special Operations Mission Planning Office. Mr. Bouressa's work includes a variety of software development projects using the latest in cut edge technology supporting and improving the preparation and battlefield awareness of US forces. Mr. Bouressa is a retired USAF officer with 24 years of active service.