Chapter 4: Introspection (Introduction)

Introspection is a term taken from the field of object-oriented programming, and it refers to the ability of data objects to describe themselves, when called upon.

- Studies emanating from Big Data resources have no scientific value, and the Big Data resources are all a waste of time and money if data analysts cannot find, or fail to comprehend, the basic information that describes the data held in the resources.
Chapter 4: Introspection (Introduction)

Examples: (Knowledge of self) – Ruby like

• First, let’s create a new object, “x”; we will assign “hello world” to the object:
  – x = "hello world"  yields  "hello world"

• We can check any object to determine its class by sending the “class” method to the object:
  – x.class  yields  String

• We can learn the name of the superclass of Class String by sending the superclass method, as shown:
  – x.class.superclass  yields  Object

• Ruby assigns a unique identifier to every created object. We can find the object identifier by
  sending “x” the object_id method: ( The unique object identifier for “x” is 22502910 )
  – x.object_id  yields  22502910

• If we ever need to learn the contents of “x”, we can send the inspect method to the object:
  – x.inspect  yields  "hello world"
Chapter 4: Introspection

• How does introspection, a feature of object-oriented programming languages, apply to Big Data?
  – In principle, Big Data resources must have the same features of introspection that are automatically provided by object-oriented programming languages.
  – Specifically, all data pertaining to the object must be encapsulated within the object to include:
    • the raw data, a description for the raw data (the so-called metadata),
    • the name of the class to which the data object belongs, and
    • a unique identifier that distinguishes the data object from all other data objects.

• But … the author must admit that most Big Data resources lack introspection.
Chapter 4: Introspection

There is always a price to pay when Big Data resources lack introspection. Symptoms of an introspection-free Big Data resource include the following:

1. The resource is used for a narrow range of activities, somewhat less than was originally proposed when the resource was initiated.

2. The resource is used by a small number of domain experts; to all others, the resource is inscrutable.

3. The resource cannot merge its data with data contained in other Big Data resources.

etc. (book page 52 A)
Chapter 4: Introspection

• Introspection is **not** a feature that you can attach to a Big Data resource as an afterthought.

• It is a foundational component of well-designed resources.

• Most Big Data resources will never attain the level of introspection available in object-oriented programming languages, but some introspective features would seem essential.

In the next Slides we will discuss the first step in introspection: Data description
Chapter 4: Introspection + XML

When you think about it, numbers are meaningless.

• The number “8” has no connection to anything in the physical realm until we attach some information to the number (e.g., 8 candles, 8 minutes).

• Some numbers, such as “0” or “-5,” have no physical meaning under any set of circumstances. There really is no such thing as “0 dollars”; it is an abstraction indicating the absence of a positive number of dollars. Likewise, there is no such thing as “-5 walnuts”; it is an abstraction that we use to make sense of subtractions (5 - 10 = -5).
Chapter 4: Introspection + XML

XML (eXtensible Markup Language) is a syntax for attaching descriptors (so-called metadata) to data values

- A file that contains XML markup is considered a proper XML document only if it is well formed. Here are the properties of a well-formed XML document.
  1. The document must have a proper XML header. The header can vary somewhat, but it usually looks something like `<?xml version = "1.0"?>`.
  2. XML files are ASCII files, consisting of characters available to a standard keyboard.
  3. Tags in XML files must conform to composition rules (e.g., spaces are not permitted within a tag, and tags are case sensitive).
  4. Tags must be properly nested (i.e., no overlapping).
- Web browsers will not display XML files that are not well formed.
Chapter 4: Introspection + XML

• The actual structure of an XML file is determined by another XML file known as an XML Schema. The XML Schema file lists the tags and determines the structure for those XML files that are intended to comply with a specific Schema document. A valid XML file conforms to the rules of structure and content defined in its assigned XML Schema.

• Every XML file valid for the same XML Schema will contain data described with the same tags, permitting data integration among those files. This is one of the great strengths of XML.
Chapter 4: Introspection + XML

• The greatest drawback of XML is that data/metadata pairs are not assigned to a unique object. XML describes its data, but it does not tell us the object of the data.

• This gaping hole in XML was filled by Resource Description Framework (RDF), a modified XML syntax designed to associate every data/metadata pair with a unique data object.

• Before we can begin to understand RDF – as cited in the last chapter - we need to understand the concept of “meaning” in the context of information science (to link data/description to an object).
Chapter 4: Meaning

- Metadata gives structure to data values, but it does not tell us anything about how the data value relates to anything else. For example:
  - `<height_in_feet_inches>5’11”</height_in_feet_inches>`
  - What does it mean to know that 5’11” is height value, expressed in feet and inches? Nothing really. The metadata/data pair seems to be meaningless.

- In informatics, meaning is achieved when described data (i.e., a metadata/data pair) is bound to the unique identifier of a data object. For example, consider this sentence: “Jules J. Berman’s height is five feet eleven inches.”
  - This sentence has meaning because there is data (five feet eleven inches), and it is described (person’s height), and it is bound to a unique individual (Jules J. Berman)
Chapter 4: Meaning

- Semantics is the study of meaning. In the context of Big Data, semantics is the technique of creating **meaningful assertions about data objects**. All meaningful assertions can be structured as a three-item list consisting of an **identified data object**, a **data value**, and a **descriptor** for the data value.

- For example: <Jules J. Berman> <height_in_feet_inches> <5’11”>

- These three-item assertions are referred to as “triples.”
Chapter 4: Meaning

In practical terms, semantics involves making assertions about data objects (i.e., making triples), combining assertions about data objects (i.e., aggregating triples), and assigning data objects to classes—hence relating triples to other triples.
Chapter 4: Namespace

A namespace is the metadata realm in which a metadata tag applies.

- Its purpose is to distinguish metadata tags that have the same name, but a different meaning.
- For example, within a single XML file, the metadata term “date” may be used to signify a calendar date, a fruit, or a social engagement. To avoid confusion, the metadata term is given a prefix that is associated with a Web document that defines the term within an assigned Web location.
- For example, an XML page might contain three date-related values, and their metadata descriptors:

  <calendar:date>June 16, 1904</calendar:date>
  <agriculture:date>Thoory</agriculture:date>
  <social:date>Pyramus and Thisbe</social:date>
Chapter 4: RDF - RESOURCE DESCRIPTION FRAMEWORK TRIPLES

RDF is a dialect of XML designed to convey triples.

Every Big Data manager must be aware of RDF features that enhance the value of Big Data resources. These would include the following:

1. The ability to express any triple in RDF (i.e., the ability to make RDF statements).
2. The ability to assign the subject of an RDF statement to a unique, identified, and defined class of objects (i.e., the ability to assign the object of a triple to a class).
3. The ability for all data developers to choose the same publicly available RDF Schemas and namespace documents with which to describe their data, thus supporting data integration over multiple Big Data resources. This last feature allows us to turn the Web into a worldwide Big Data resource composed of RDF documents.
Chapter 4: RDF - RESOURCE DESCRIPTION FRAMEWORK TRIPLES

We will briefly examine each of these three features in RDF. First, consider the following triple:

```
pubmed:8718907, creator, Bill Moore
```

- Every triple consists of
  - an identifier (the subject of the triple),
  - followed by metadata,
  - followed by a value.

- In RDF syntax, the triple is flanked by metadata indicating the beginning and end of the triple. This is the `<rdf:description>` tag and its end tag `</rdf:description>`. The identifier is listed as an attribute within the `<rdf:description>` tag and is described with the `rdf:about` tag, indicating the subject of the triple. There follows a metadata descriptor, in this case `<author>` (the text use author and creator), enclosing the value, “Bill Moore”.

```
<rdf:description rdf:about = "urn:pubmed:8718907">
<creator>Bill Moore</creator>
</rdf:description>
```

The RDF triple tells us: Bill Moore wrote the manuscript identified with the PubMed number 8718907. The PubMed number is the National Library of Medicine’s unique identifier assigned to a specific journal article.
We could express the title of the article in another triple:

```
pubmed:8718907, title, “A prototype Internet autopsy database. 1625 consecutive fetal and neonatal autopsy facesheets spanning 20 years.”
```

In RDF, the same triple is expressed as:

```
<rdf:description rdf:about = "urn:pubmed:8718907">  
 <title>A prototype Internet autopsy database. 1625 consecutive fetal and neonatal autopsy facesheets spanning 20 years</title>  
</rdf:description>
```
Chapter 4:
RDF - RESOURCE DESCRIPTION FRAMEWORK TRIPLES

Resource Description Framework permits us to nest triples if they apply to the same unique object:

```xml
<rdf:description rdf:about = "urn:pubmed:8718907">
  <author>Bill Moore</author>
  <title>A prototype Internet autopsy database. 1625 consecutive fetal and neonatal autopsy facesheets spanning 20 years</title>
</rdf:description>
```

Here we see that the PubMed manuscript identified as 8718907 was written by Bill Moore (the first triple) and is titled “A prototype Internet autopsy database. 1625 consecutive fetal and neonatal autopsy facesheets spanning 20 years” (a second triple).
Chapter 4: RDF - RESOURCE DESCRIPTION FRAMEWORK TRIPLES

What do we mean by the metadata tag “title”? How can we be sure that the metadata term “title” refers to the name of a document and does not refer to an honorific (e.g., The Count of Monte Cristo or the Duke of Earle).

We append a namespace to the metadata.

Namespaces were described in this chapter.

<rdf:description rdf:about = "urn:pubmed:8718907">
  <dc:creator>Bill Moore</dc:creator>
  <dc:title>A prototype Internet autopsy database. 1625 consecutive fetal and neonatal autopsy facesheets spanning 20 years</dc:title>
</rdf:description>

In this case, we appended “dc:” to our metadata.

By convention, “dc:” refers to the Dublin Core metadata set at http://dublincore.org/documents/2012/06/14/dces/.
Chapter 4:
RDF - RESOURCE DESCRIPTION FRAMEWORK TRIPLES

The Dublin Core is a set of metadata elements developed by a group of librarians at a workshop in Dublin, Ohio, in 1995.

It would be very useful if every electronic document were annotated with the Dublin Core elements. The most popular Dublin Core metadata properties are:

**contributor** - the entity that contributes to the document
**coverage** - the general area of information covered in the document
**creator** - entity primarily responsible for creating the document
**date** - a time associated with an event relevant to the document
**description** - description of the document
**format** - file format
Chapter 4:
RDF - RESOURCE DESCRIPTION FRAMEWORK TRIPLES

**identifier** - a character string that uniquely and unambiguously identifies the document

**language** - the language of the document

**publisher** - the entity that makes the resource available

**relation** - a pointer to another related document, typically the identifier of the related document

**rights** - the property rights that apply to the document

**source** - an identifier linking to another document from which the current document was derived

**subject** - topic of the document

**title** - title of the document

**type** - genre of the document
Chapter 4: RDF - RESOURCE DESCRIPTION FRAMEWORK TRIPLES

Resource Description Framework was developed as a semantic framework for the Web.

- The object identifier system for RDF was created to describe Web addresses or unique resources that are available through the Internet.
- The identification of unique addresses is done through the use of a Uniform Resource Name (URN).
- In many cases, the object of a triple designed for the Web will be a Web address. In other cases, the URN will be an identifier, such as the PubMed reference number in the example above. In this case, we appended the “urn:” prefix to the PubMed reference in the “about” declaration for the object of the triple:

  <rdf:description rdf:about = "urn:pubmed:8718907">
Chapter 4:
RDF - RESOURCE DESCRIPTION FRAMEWORK TRIPLES

Let us create an RDF triple whose subject is an actual Web address:

<rdf:Description rdf:about = "http://www.usa.gov/">
</rdf:Description>

Here we make a triple wherein the object is uniquely identified by the Web address http://www.usa.gov/, and the title of the Web page is “USA.gov: The U.S. Government’s Official Web Portal.”
Chapter 4:
RDF - RESOURCE DESCRIPTION FRAMEWORK TRIPLES

How can we assign a data object to a class of objects?
We do this with the “rdf:type” declaration:

```xml
<rdf:description rdf:about = http://www.usa.gov/>
    <rdf:type resource = "http://dublincore.org/documents/2012/06/14/dcim-terms/?v=terms#FileFormat">
</rdf:description>
```

This RDF statement tells us that the Web address “http://www.usa.gov” is a type of FileFormat (i.e., is a member of Class FileFormat).

Information about Class FileFormat is available in a Schema document at the following address: http://dublincore.org/documents/2012/06/14/dcim-terms/?v=terms#FileFormat.
If everyone were to assign classes to their data objects, and if everyone used the same set of Schemas to define their classes, then the Web would become a worldwide Big Data resource.

This is the principle underlying the concept of the Semantic Web popularized in recent years by the World Wide Web Consortium (W3C).
Chapter 4: Reflection

Reflection is a programming technique wherein a computer program will modify itself, at run time, based on information it acquires through introspection.

- Because introspection is a property of well-constructed Big Data resources, reflection is an available technique to programmers who deal with Big Data.

Be careful ...
Chapter 4:

Use case – trusted time stamp

A time stamp is a data value that contains the time that an event occurred and can be expressed as the date plus Greenwich Mean Time.

The time stamp is irreversibly and immutably linked to an event or any data object. Because the time stamp is a data value, it should be attached to metadata. A few triples can clarify how time stamps can be used.

- 882773 is_a “patient” 882773 has_a 73002548367
  73002548367 is_a “lab test” 73002548367
  GMT_accession_time “2012-6-28 01:01:36” 73002548367
  GMT_validation_time “2012-6-28 01:10:42"
Chapter 4: Summary

- Introspection refers to the ability of a data object to describe itself. In practical terms, a data analyst should be able to determine the data contents of every data object, the descriptors that apply to each data element, the identifier of the data object, the class assignment for the data object, and the properties that apply to the assigned class, as well as the names and properties of the classes that are related to the class assigned to the data object.

- Resource Description Framework is a formal syntax for triples. The subjects of triples can be assigned to classes of objects defined in RDF Schemas and linked from documents composed of RDF triples. When data objects are assigned to classes, the data analysts can discover new relationships among the objects that fall into a class and can also determine relationships among different related classes (i.e., ancestor classes and descendant classes, also known as superclasses and subclasses).

RDF triples plus RDF Schemas provide a semantic structure that supports introspection
Chapter 4: Exercícios - exercício 6

Entrega: 21/outubro.

1. Considere o caso de estudo às fls. 59/60 e slide 26. Comente a questão de “time-stamp”, confiável e não-confiável, dentro de um exemplo que você construa.

2. Construa um exemplo de descrição de uma aplicação via RDF. Se necessário busque uma bibliografia.