Abstract—This paper intends to introduce the principles of Linked Data, show some possible uses in Social Networks, advantages and disadvantages of using it, as well as presenting an example application of Linked Data to the field of social networks.

I. INTRODUCTION

The web is a very good source of human-readable information that has improved our daily life in several areas. By using rich text and hyperlinks, it made our access to information something more active rather than passive. But the web was made for humans and not for machines, even though today a lot of the accesses to the web are made by machines. This has lead to the search for adding structure, so that data is more machine-readable. Even when the data is structured, the access is usually through Web APIs, requiring manual implementation for each API and making very hard to connect data from different APIs. Amid this chaos the field of Linked Data \cite{1} has appeared to define some standards and make connected data more easily available.

Its goal is to solve exactly those problems described, by defining a set of good practices in publishing and interlinking data, creating an environment where data can be accessed everywhere using the same interfaces and within the same framework. In summary, Linked Data uses the hyperlink concept (from the human web) to connect different datasets (and help create a machine web). In both cases, those hyperlinks enable a global web of information.

II. AN OVERVIEW OF LINKED DATA

To achieve the bold goals it hopes, whoever works must follow a set of principles and practices in order to create this so-called Web of Data.

A. Principles

The principles that guide Linked Data, listed in \cite{1}, can be paraphrased as as:

1) Use URIs as identifiers for objects and documents.
2) Use HTTP URIs so that objects can be refered and looked up both by humans as by machines.
3) When an URI is requested, provide useful information, using standards like RDF.
4) Relate this URI with other URIs, creating structure in the web.

The first one basically extends the scope of Universal Resource Identifiers (URIs) to instead of just referencing Webpages, also possibly reference any Object or Concept (or even relationships of concepts/objects).

The second one combines the unique identification of the first with a well known retrieval mechanism (HTTP), enabling this URIs to be looked up via the HTTP protocol.

The third principle tackles the problem of using the same document standards, in order to have scalability and interoperability. This is done by having a standard, the Resource Description Framework (RDF), which will be explained in section III-A. This standards are very useful to provide resources that are machine-readable and are identified by a URI.

The fourth principle basically advocates that hyperlinks should be used not only to link webpages, but also to link other data. Those hyperlinks, not like the web ones, would be typed. (e.g. A friend-type hyperlink between two people) This principle incites the adding of value to data resources by connecting them to other data that can provide more useful information.

B. The Web of Data

The use of these Linked Data principles by a large community led to the creation of a vast number of datasets, which interlinked form the Linking Open Data Cloud.

As with any big idea, it has to start somewhere. And that somewhere is probably the W3C Linking Open Data (LOD) Project, which sought to bootstrap this web of data (along with the semantic web community). Looking at what it was in 2007 (Figure 1) and what it has become (Figure 2), it gives some indications of being a success, at least in some areas.

This concept of a Web of Data is a very interesting one. It consists on the idea of having a giant global graph of information that covers all sorts of topics, from music to census data. It’s basically the idea of connecting the Data we have in a way very similar to the way the web connects documents.

As the Figure 2 can show, the Web of the data is very diverse, including Geographic, Gornmental, Media, Life Sciences and User Content Datasets, there are also Datasets from Libraries. That can also be seen which shows the distribution in the Domains of the Datasets in Table I.

This Web of Data also can have negative aspects. Being hugely dependent on other systems that provide added useful information, if one service goes discontinued or has a failure, it can affect all others. Other fact to be noted is that this graphs showed in Figures 1 and 2 can be somewhat misleading, not all
some were discon-
tinued, are out-
dated by years or have very poor docu-
mentation and/or quality, while others represen-
t some of the best the web can offer in matters of
Scientific Data. This problems will be discussed again on
the Section V.

III. PUBLISHING LINKED DATA

Before understanding how to publish linked data, some
things must be explained, especially how does the
RDF work and how are URIs Dereferenced. With that, the publishing
patterns become clearer.

A. Resource Description Framework

The Resource Description Framework (or RDF) is a
framework for representing information in the Web. [3] The follow-
ing subsections describe its key concepts.

1) Graph data model: The structure of the RDF expressions
are triples (subject, object, predicate). This structure is usually
denoted as two nodes (subject and object) and a directed arc
(predicate). This arc denotes that a property or relationship
holds between subject and object.

The nodes in the model can be either:
• an URI (with optional fragment identifier)
• a literal
• a blank node

Should also be noted that the Subject can never be a literal,
and the arc (predicate) is always an URI.

2) Datatypes: Datatypes are important in RDF to represent
values (such as booleans, integer, floating point numbers,
dates, ...)

They consist of: Value Space, Lexical Space and Lexical-
to-Value Mapping

One example (adapted from [3]) would be boolean datatype:
The Value Space would be {True, False}.
A Lexical Space could be{"0", "1", "true", "false"}
The Lexical-to-Value Mapping would then be {<"true", True>,<"1", True>,<"false", False>,<"0", False>}

3) Literals: Literals in RDF can be either plain or typed.
Plain literals are simple strings in natural language and are
recommended to be self-denoting. Typed literals are simple
strings, combined with a datatype URI that provides the
Lexical-to-Value mapping as shown in Section III-A2 and
represent the value given by the mapping.

4) Serialization Syntax: RDF is NOT a data format. It’s a
data model and in order to be published it must be serialized.
There are two standard W3C serialization formats: RDF/XML
[4] and RDFa [5] (and many other non-standard that are used
in specific situations).

There are some RDF Formats that are usually used: RDF-
XML, Turtle, N-Triples and RDFa. They are exemplified
below:

These examples also make use of FOAF (Section IV-1)
and are an excerpt from the Datasets used on the Example
Application (Section V).

```
RDF/XML
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-
rdf-syntax-ns#"
  xmlns:j.0="http://somewhere/ont#"
  xmlns:j.1="http://xmlns.com/foaf/0.1/" >
  <rdf:Description rdf:about="http://somewhere/socialnet#me">
    <j.1:knows rdf:resource="http://somewhere/socialnet#rosa"/>
    <j.0:ListenedTo>Bad Romance</j.0:ListenedTo>
    <j.1:name>Gustavo Valdez</j.1:name>
  </rdf:Description>
</rdf:RDF>
```
5) Expression of simple facts: RDF expresses only a conjunction of simple facts, that is, its logic model only comports AND (a list of facts means all of them hold). There’s no means to state other complex facts such as negation (NOT statement) and disjunction (OR statement) [3].

That’s not to say that it could not be added using the predicate structure given, but it would be cumbersome and too complicated, since there is no native support.

6) RDF, RDF-Schema and OWL: RDF’s describes resources in a (subject, predicate, object) triple, but it does not provide support for domain-specific terminologies, classes of things and relations of classes directly. These kinds of functions are better served by Ontologies, Vocabularies and Taxonomies. For expressing that we can use RDF-Schema [6] and OWL (Web Ontology Language) [7].

The difference probably is made more clear by an example. RDF is only capable of expressing relationships between two objects. RDFS adds more structure, making it possible to define classes for the objects and relationships. For example, with it it’s possible to express that brotherhood is a relationship between two People. OWL gives even more power, by adding semantics, that is: It’s able to infer that if some Person is married to another, the second is married to the first. Or that if an object is smaller than other in a partial order relationship and the second is smaller than a third, then the first will be smaller than the third.

B. URI Dereferencing

As stated in Principle 1 (Section II-A), Linked Data uses the widespread internet concept of URIs to reference real-world objects as well as documents about them. That means that a URI can represent (in the linked data context) either a Resource, a RDF document about a resource or a HTML document about a resource. It’s very important to not mistake the object for the document that describes it.

The two strategies to dereference URIs that are showed below allow for this separation between objects and documents to happen. They are 303 URIs and Hash URIs [8].

1) 303 URIs: The 303 URIs Strategy consists in using the HTTP response “303 See Other”, because real-world objects cannot be sent over the wire. this 303 response would then redirect to a document that describes the object.

That means that when a browser or linked data application makes a GET request to a URI that represents a real-world object, a 303 redirect will happen and the application will then have to do another GET request to the document it now knows describes this object.

The “303 See Other” can even point to different documents depending on the viewer, using content negotiation. For example, a browser could be redirected to an HTML document, while a linked data application would be to a RDF/XML document. Note that it can also point directly to a generic document that then will do this content negotiation.

2) Hash URIs: The Hash URIs Strategy uses the HTTP fragment identifier, instead of the “303 See Other”. The idea being that a vocabulary file is fetched and the concept is then looked for using the fragment identifier in this file.

For example, with the URI http://youruri.com/people with vocabulary about people in your organization. Then http://youruri.com/people#Steve would describe Steve and http://youruri.com/people#Bob would describe Bob.
That has the advantage of not requiring two GET requests to look for a real-world object, in various circumstances, one can even look at more than one object with one GET request, since the fragment identifier is local.

3) 303 or Hash URIs: Both strategies have advantages and disadvantages. 303 URLs can have a bigger latency, since more HTTP round trips are needed. On the other hand, Hash URIs may have to transmit a lot of useless data, since the whole vocabulary is transmitted (and may not be needed).

One has to decide which is best based on which kind of access will be normally made.

C. Publishing Patterns

To publish linked data, one has to be able to serve RDF files to the web. The method used to store and serve these files depends on volume, structure and dynamicity of the data. A general idea of the publishing methods is given in figure 3.

To publish linked data, one has to be able to serve RDF files to the web. The method used to store and serve these files depends on volume, structure and dynamicity of the data. A general idea of the publishing methods is given in figure 3.

![Fig. 3. Linked Data Publishing Options and Workflows. (Figure 5.1 of [1])](image)

The patterns are basically defined by what one already have that should be converted to Linked Data. Other factors may also be relevant to the choice, especially the way the data is accessed.

1) Queryable Structured Data: If the data is at a relational database, it can be easily published through one of the RDF Wrappers available (Such as D2R Server). In case it’s behind an API, a custom wrapper will probably be needed.

2) Static Structured Data: If the data is in form of structured files (Such as CSV, Spreadsheets or XML), it should be converted to RDF. Some “RDFizers” are available for lots of formats [9] (and it’s usually not that hard to write code to convert structured datasets)

3) Text Documents: In the case of having natural language textual documents to publish as linked data, things get a bit more complicated. In that cases, one should try using one of the Linked Data entity extractors available (like DBpedia Spotlight [10]). They annotate the textual data to increase the discoverability beyond full-text search.

IV. Usage of Linked Data in Social Networks

Linked data can also be used in social networks and its use is still growing. Even the big names in social networks (Facebook, Google Plus) are now working to introduce semantic information (See section IV-A), in order to ease third party development and allow more new applications.

1) Friend Of A Friend (FOAF): The Friend of a Friend project [11] intends to build a web of machine-readable documents about people, linking them to anything that is related to them. It tries to achieve that goal by defining certain standards for the fields in a profile. It’s basically an Ontology about the area that allows developers to much build RDF-capable social profiles much easier. Although its premises are very interesting, there are still open questions about privacy as it’s not really treated by the project. Other factor that could be seen as a problem is the fact that changes in the standard are very slow and done by consensus, which makes it less dynamic. For a use case scenario and example, section V can be seen.

2) Facebook’s Open Graph: Facebook launched recently a protocol for accessing it’s social network. It’s called Open-Graph [12] and it has now compatibility with RDF (specifically RDF/Turtle) and is an interesting step for making facebook part of the semantic web. With the importance of Facebook today, this move has to be taken into account when talking about social networks and semantic web, since their decisions affect most of the commercial developers of social applications. Besides Facebook, Google Plus has also implemented support for the OpenGraph format. For now, it’s use is becoming widespread throughout Facebook and most applications have adapted. We have still to wait whether a Data Mining “Killer App” will come with this new flow of structured data.

3) Semantically-Interlinked Online Communities (SIOC): The Semantically-Interlinked Online Communities [13] is a project that provides a Semantic Web ontology for rich data representation in RDF. One of its main aims is to provide interoperability between different social networks. It plans to do that by creating an Interchange format that could allow any social network to interact with any other. Besides that, it provides some semantic information on social networks that can be very useful if well applied.

A. A Discussion of Social Networks Applications

Nowadays, social networks are not anymore sistems written by a group of programmers that help people connect. They have evolved into platforms where any developer with a good idea can write code and mine the data of people’s Social Graph. It has, then, to become a concern of the developers of the network how easy it’s for developers to use their platform, much alike Operating Systems striving to have more developers so that the user has a better experience. Semantic web offers in this scenario a standardized framework for structured data, that social networks can apply. For the application developer it is also a good thing, since he can...
learn something not just to work in one social network, but in all that would support the standards.

There’s also the risk that Facebook, as the main player in the market, will not want real interoperability, much like Microsoft’s Windows some years ago. But even if they don’t want that interoperability, they have shown they do want the advantages structured data can bring as can be seen by the launch of Facebook Open Graph (Section IV-2). With those factor accounted, it seems us clear that even if the standards were not to be a complete success, learning the main ideas behind Linked Data is useful, because it’s concepts will certainly be needed by developers in the future.

V. Example Application

To learn more about the standards, some tools for programming using them, see what kind of problems can be faced by a developer and to have hands-on experience, a very simple application was developed. Its code and datasets can be found at [14].

The problem the application wants to tackle is the following: “Suppose someone wants to throw a party for some friends, but he’s not really sure of their musical tastes or that this someone want to invite a friend for a dinner and also need to know her/his musical tastes.”

The usual solution could be asking them, or if one wants to surprise them, other solution would be to use social networks to find what are they listening to and what bands they like. But this can be automated, and an application can be written that does this work. A simplified version of this problem is tackled by the example.

A. Requisites

Before going into the application per se, the tools and datasets used and tried must be clear. It was developed in Java with the help from the Apache Jena [15], which is a very good framework for working with Linked Data in the Java platform. For the social networks ontology used, there was not that much doubt, FOAF (Section IV-1) was selected. The problems really started to appear when came time to select the music ontology to be used.

The following music ontologies were tested: DBtune [16], Music Ontology [17], MusicBrainz [18] and BBC Music [19]. Each one had its own problems, even from not being updated (or not having the specification updated) in 3 to 5 years, not working that well anymore, having lots of links to somewhere that weren’t working anymore. The best one was MusicBrainz, which couldn’t be used with Jena, since it currently is only supporting RDFa and Jena has some problems with it. Besides, they are not that interested in the RDF/SPARQL facet anymore. Second best was BBC Music, which is very well organized, but has the disadvantage of being greatly incomplete, since it has different goals. DBpedia [10] was also tested which, as expected from a computer generated ontology, is not that much consistent.

SPARQL [20], which is a SQL-equivalent to query RDF databases, was also used, with some success, in some of the music ontologies. In the end, for restrictions of time and effort, the use of those music ontologies was abandoned.

B. Datasets

The datasets that were created for this application included: a FOAF network with some friends and what kinds of music they like (a small ontology with some Object Properties was created to make it possible to do that) and a small music dataset that include those musics and bands used by those friends.

C. Implementation

With those datasets, the code implemented was very simple. Not really a recommender system, but it lists the intersections and unions between music likes to create playlists. It’s also able to determine which friends have a specific music like. That is done using the datasets created. As stated before, the full code can be found at [14].

VI. Conclusion

Linked Data is a very powerful concept that has yet much to grow and evolve. It’s concepts are very important and useful to be known, but if it’ll be the standard for social networks is still an open question. It would be good if it was, but markets don’t take the best decision, but each player searches its interest and the market is just a result of that. To really see whether social networks will or not use those standards, maybe all we can do is wait. But, would be recommended to study those anyway, because the inherent principles and concepts semm to be here to stay.

Acknowledgment

The author would like to thank Abdulbaki Uzun for all the help supervising this work.

References