Modeling the Web of Data (introductory overview)

Claudio Gutierrez
Department of Computer Science
Universidad de Chile

Contents

Scope and Disclaimer These notes are meant as a companion to a lecture on the topic at the Reasoning Web Summer School 2011. The goal of this work is to present diverse and known material on modeling the Web from a data perspective, to help students to get a first overview of the subject.

Methodologically, the objective is to give pointers to the relevant topics and literature, and to present the main trends and development of a new area. The idea is to organize the existing material without claiming completeness. In many parts the notes have a speculative character, oriented more towards suggesting links and generating discussion on different points of view, rather than establishing a consolidated view of the subject.

The historical accounts and references are given with the sole objective of aiding in the contextualization of some milestones, and should not be considered as signaling intellectual priorities.

[0] *Acknowledgments.* Materials of this lecture have been taught to students at Universidad de Chile, Chile; Universidad de la República, Uruguay; Biblioteca del Congreso, Chile; to whom I thank for feedback and suggestions. Also thanks to R. Angles, J. Fernández, D. Hernández, J. E. Muñoz plus anonymous referees that helped with detailed comments to improve previous versions. Of course, the responsibility for what is finally said here is mine.

Introduction

From the point of view of information, the most naive –and probably also the most understandable– model of the Web is that of an infinite library. The idea is not new: in 1939 Jorge Luis Borges published the story *The Total Library*[1], where he writes:

> *"Everything would be in its blind volumes. Everything: the detailed history of the future, Aeschylus' The Egyptians, the exact number of times that the waters of the Ganges have reflected the flight of a falcon, the secret and true nature of Rome, the encyclopedia Novalis would have constructed, my dreams and half-dreams at dawn on August 14, 1934, the proof of Pierre Fermat's theorem, the unwritten chapters of Edwin Drood, those same chapters translated into the language spoken by the Garamantes, the paradoxes Berkeley invented concerning Time but didn't publish, Urizen's books of iron, the premature epiphanies of Stephen Daedalus, which would be meaningless before a cycle of a thousand years, the Gnostic Gospel of Basilides, the song the sirens sang, the complete catalog of the Library, the proof of the inaccuracy of that catalog. Everything: but for every sensible line or accurate fact there would be millions of meaningless cacophonies, verbal farragoes, and babblings. Everything: but all the generations of mankind could pass before the dizzying shelves-shelves that obliterate the day and on which chaos lies-ever reward them with a tolerable page."*

The view of a universal space of information as the (infinite) generalization of a library is an extremely useful one. It includes almost all facets we would like to incorporate when abstracting and modeling such an artifact. There is one crucial slant, though: the library is composed of books, let us say in Web terms, of documents. Documents (books) are artifacts produced by humans to be consumed by humans. If one replaces data in the place of books, we essentially have an abstract model of the "Web of Data". But this is not a minor change, bringing with it complex challenges.

Modeling the Web of data is a relevant goal. The big excitement about current levels of production, availability and use of data indicates that we are witnessing a fundamental change in information practices. The tide of data was observed a few years ago by cutting-edge technology analysts. In his widely read 2005 article that sparked the notion of Web 2.0 [66], O'Reilly wrote that "data is the next Intel Inside." On a more academic level, the Claremont Report on Database Research [6] centered its analysis on the challenges that this phenomena is posing, stating that ubiquity of "Big Data" will shake up the field [of databases]. Szalay and Gray pointing to the fact in 2006, that the amount of scientific data is doubling every year, spoke of an "exponential world" [30] and

[1] J. L. Borges, La Biblioteca Total, Sur No. 59, August 1939. Trans. by Eliot Weinberger. In Selected Non-Fictions (Penguin: 1999).

Bell et al. [18] called it "Data Deluge". They state that, compared to volumes generated only a decade ago, some areas of science are facing hundred- to thousandfold increases in data volumes from satellites, telescopes, high-throughput instruments, sensor networks, accelerators, and supercomputers.

The phenomena is not exclusive of the scientific fields. A similar trend can be found in almost all areas. Social networks are generating not only high volumes of data, but complex networks of data which call for a new stage in data management. New technologies have also impacted government policies. Transparency laws and wide-range archiving and publishing initiatives are posing similar challenges to the public sector [25]. Managing, curating and archiving of digital data is becoming a discipline per se. Today some people are even talking about "data science" [46].

It is no surprise that this phenomena has put data at the center of computing discipline itself, both, at the level of systems, architecture and communications (see "petascale computational systems" [17]), new database architectures at web-scale [65, 60], and at the programming and modeling levels. In these new developments the Web, as the natural common platform for handling such data, plays a central role.

Data management at Web scale With the advance of computing power in the last decade, the perspective on the Web is gradually shifting from a document centric-view to a data centric-view. Originally conceived as a global hypertext model, today the granularity of the information on the Web has reached the level of atomic data. For example, the project *Linked Data* [44, 20] views the Web as a huge collection of data and links.

How to manage data at Web scale? Since the very origins of the Web, the database community has addressed this challenge. In the late nineties the efforts to integrate the new Web phenomena and database technology provoked a heated discussion. Is the Web a database? Could the classical database techniques be of any use in this new environment?

Two main lines of thought were developed. The first one conceived the Web as a collection of documents plus hyperlinks, and extended the ideas and technologies of hypertext and followed the lines of semi-structured data and information retrieval techniques [21, 2, 5]. This was consistent with the view that "sites" and Web pages were the central objects of interest. This, combined with the need to model documents and the exchange and integration of information, made this conception dominant. The research centered on semi-structured data and query languages, which with the advent of XML, dominated the scene for the decade of 2000 [5].

A different perspective called for modeling the Web as a database and developed the so called Web-query languages [55]. The systematic exploration of the idea of modeling the Web as a huge repository of structured data using database techniques did not succeed, likely because the amount of structured data on the Web did not yet reach a critical level. Such ideas were too futuristic for the time, though recent developments as the one mentioned at the beginning, show

that the need has reemerged.

In the meantime, several areas of research have addressed, with variable emphasis and focus, the problems of data on the Web. Among them, projects like Semantic Web, Linked data, Open data, put the the topic in the main discussion forums. From a database point of view, areas such as distributed, semi-structured and graph databases, and particular topics like incomplete information, cost models, etc., have addressed similar problems on a smaller scale. There are also other areas such as information system, multimedia, etc., that touch on problems of data on the Web, but their exhaustive enumeration would be too long to fit here.

Notes Outline These notes present an overview of the work done in modeling data on the Web and discusses requirements needed to convert the current Web of documents in a Web of Data. The organization is as follows: in section 1 we study the principles of the Web as devised by their founders and the evolution of the Web. In section 2, we present basic tools and projects that have helped build the Web of Data. In section 3, we review data representations on the Web and data models of the Web. In section 4 we bring forward a group of requirements and themes that should be addressed in a model of the Web of Data. In section 5, we briefly review the work in related areas which touch on the problems, concerns and techniques faced in our "field". Finally in section 6, we round up our trip through this new area.

1 The Web

Tim Berners-Lee (TBL from now on), the creator of the Web, states that its "major goal was to be a shared information space through which people and machines could communicate" [11]. Let us read between the lines. He meant a "global" information space, a kind of gigantic, infinite, blackboard to write and read: "The most important thing about the Web is that it is universal"[12]. But this is not enough: another key consideration is that it should be "shared". By whom? Not by a company, not by a government, not by a particular organization: shared by all people around the world.

The problem he was addressing what that of people working at CERN, located around the world, in different research labs and academic places. This was a heterogeneous group, managing and exchanging heterogeneous type of information (addresses and phone lists, research notes, official documentation, etc.), via a heterogeneous infrastructure of terminals, servers, supercomputers, etc., with diverse operating systems, software and file formats. As Roy Fielding [27] sated, the challenge was to build a system that would provide a universally consistent interface to this structured information, available on as many platforms as possible, and incrementally deployable as new people and organizations joined the project.

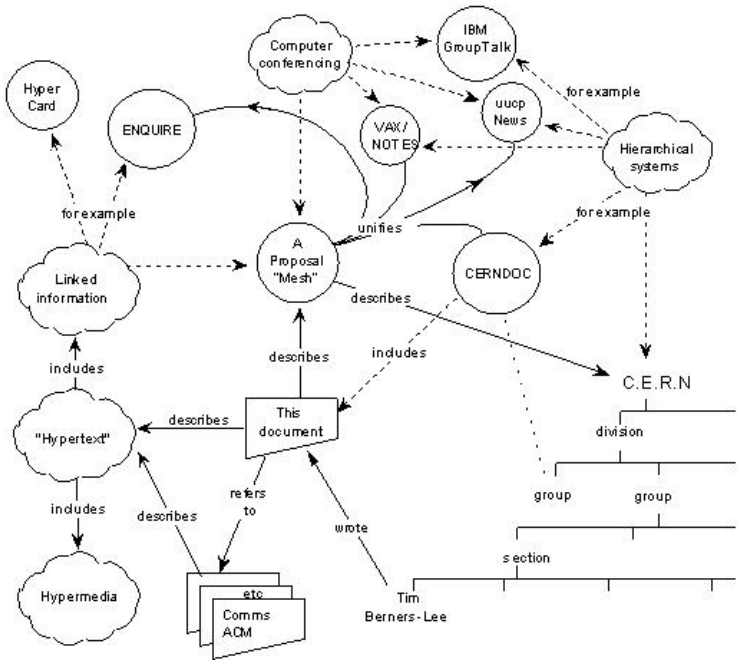

Figure 1: The first proposal of the Web by TBL. Note the underlying ideas: heterogenous data, heterogeneous users, lack of hierarchies, networking, mainly documents. (Picture taken from TBL, Information Management: A Proposal)

1.1 The classical Web

In 2001, in his Japan Lecture [12], TBL defined the Web as follows:

> *"The concept of the Web integrated many disparate information systems, by forming an abstract imaginary space in which the differences between them did not exist. The Web had to include all information of any sort on any system. The only common idea needed to tie it all together was the Universal Resource Identifier(URI) identifying a document. From that cascaded a series of designs of protocols (such as HTTP) and data formats (such as HTML) which allowed computers to exchange information, mapping their own local formats into standards which provided global interoperability."*

The architecture of the Web is based on three basic pillars:

1. URI (*Universal Resource Identifiers*), a set of global identifiers which can be created and managed in a distributed form.

2. HTTP (*Hyper Text Transfer Protocol*): a protocol for exchanging data on the Web whose basic functions are putting data in, and getting data from, this abstract space.

3. HTML (*Hyper Text Markup Language*): a language for representing information and displaying (visualizing) it to humans.

Of these three, the global identifiers are the keystone. TBL highlights this point saying that "the Web still was designed to only fundamentally rely on one specification: the Universal Resource Identifier." The particular form of the transfer protocol and of the language, are temporal solutions with the technology and knowledge available at the time.

If one would like to generalize, the Web can be thought of as supported by three basic specifications:

1. Global Identifiers.

2. A protocol to exchange data.

3. A language to represent data.

TBL's general requirements In the Japan lecture, TBL stated the following principles/requirements that should guide the development of this architecture (the text closely follows his wording):

1. *Device independence.* The same information should be accessible from many devices. The size of the screens, the means of input and output information should be independent of the hardware.

2. *Software Independence.* The Web should support diverse programs and software. The decentralization of software development was and always will be crucial to the unimpeded growth of the Web. It also prevents the Web itself from coming under the control of a given company or government through control of the software.

3. *Internationalization.* The Web should not depend on one country or culture. Internationalization should take into account not only the language, but also the direction in which text moves across the page, hyphenation conventions, and even cultural assumptions about the way people work and address each other, and the forms of organization they make.

4. *Multimedia.* Multimedia is at the heart of modern digital objects. Images, music, video have to be essential part of the design of the Web.

5. *Accessibility.* Just as people differ in the language, characters and cultures to which they belong, so they differ in terms of their capacities, regarding vision, hearing, motor or cognition. The universality includes making the Web a place which people can use irrespective of disabilities.

6. *Rhyme and Reason.* There is another axis along which information varies: its purpose and usage. At one end of the axis is the poem, at the other, the database table. Most information on the Web now contains both elements. The Web technology must allow information intended for a human to be effectively presented, and also allow machine processable data to be conveyed.

7. *Quality.* Quality notions are very subjective, and change with time, and all of them should be allowed in the Web. To support this, the technology must allow powerful filtering tools which, combining opinions and information about information from many sources, are completely under the control of the user.

8. *Independence of Scale.* Although the Web is a global phenomenon, personal, family and group information systems are part of it too. The Web must support all of those, allowing privacy of personal information to be negotiated, and groups to feel safe in controlling access to their spaces. Only in such a balanced environment can we develop a sufficiently complex and many-layered fractal structure which will respect the rights of every human being.

Requirements for the Protocols Roy Fielding, one of the authors of the HTTP protocol,

In his doctoral thesis, Roy Fielding, one of the authors of the HTTP protocol, explored in depth the architecture of the Web [27]. He identified the following requirements (the text follows his wording):

1. Low Entry-barrier. Since participation in the creation and structuring of information was voluntary, a low entry-barrier was necessary to enable sufficient adoption. This applied to all users of the Web architecture: readers, authors, and application developers.

2. Extensibility. While simplicity makes it possible to deploy an initial implementation of a distributed system, extensibility allows us to avoid getting stuck forever with the limitations of what was deployed. Even if it were possible to build a software system that perfectly matches the requirements of its users, those requirements will change over time just as society changes over time. A system intending to be as long-lived as the Web must be prepared for change.

3. Distributed Hypermedia. Hypermedia is defined by the presence of application control information embedded within, or as a layer above, the presentation of information. Distributed hypermedia allows the presentation and control information to be stored at remote locations.

The usability of hypermedia interaction is highly sensitive to user-perceived latency: the time between selecting a link and the rendering of a usable result. Since the Web's information sources are distributed across the global Internet, the architecture needs to minimize network interactions (round-trips within the data transfer protocols).

4. Internet-scale. The Web is intended to be an Internet-scale distributed hypermedia system, which means considerably more than just geographical dispersion. The Internet is about interconnecting information networks across multiple organizational boundaries. Suppliers of information services must be able to cope with the demands of anarchic scalability and the independent deployment of software components.

a) *Anarchic Scalability.* Most software systems are created with the implicit assumption that the entire system is under the control of one entity, or at least that all entities participating within a system are acting towards a common goal and not at cross-purposes. Such an assumption cannot be safely made when the system runs openly on the Internet. Anarchic scalability refers to the need for architectural elements to continue operating when they are subjected to an unanticipated load, or when given malformed or maliciously constructed data, since they may be communicating with elements outside their organizational control.

b) *Independent Deployment.* Existing architectural elements need to be designed with the expectation that later architectural features will be added. Likewise, older implementations need to be easily identified so that legacy behavior can be encapsulated without adversely impacting newer architectural elements. The architecture as a whole must be designed to ease the deployment of architectural elements in a partial, iterative fashion, since it is not possible to force deployment in an orderly manner.

Based on these principles, he proposed a style of software architecture for distributed hypermedia systems called *REST* (Representational State Transfer). These are the constraints defined for it:

1. *Client-server.* Separation of concerns is the principle behind the client-server constraints. Clients are separated from servers by a uniform interface. This separation of concerns means that, for example, clients are not concerned with data storage, which remains internal to each server, so that the portability of client code is improved. Servers are not concerned with the user interface or user state, so that servers can be simpler and more scalable. Servers and clients may also be replaced and developed independently, as long as the interface is not altered.

2. *Stateless.* Each request from client to server must contain all of the information necessary to understand the request, and cannot take advantage of any stored context on the server. Session state is therefore kept entirely on the client. This constraint induces the properties of visibility, reliability, and scalability.

3. *Cacheable.* Cache constraints require that the data within a response to a request be implicitly or explicitly labeled as cacheable or non-cacheable. If a response is cacheable, then a client cache is given the right to reuse that response data for later, equivalent requests.

4. *Uniform interface.* The central feature that distinguishes the REST architectural style from other network-based styles is its emphasis on a uniform interface between components.

5. *Layered system.* The layered system style allows an architecture to be composed of hierarchical layers by constraining component behavior such that each component cannot "see" beyond the immediate layer with which they are interacting.

What about the language requirements? Paradoxically, one of the reasons for the success of the Web was the weaknesses of its language HTML: loose structure (allowing the display badly-formed pages) and only oriented to visualization (by humans). The next generation language, XML, improved on both aspects: (1) strict enforcement of structure and constraints (allowing semi-structured querying); and (2) flexible to code different objects languages (for visualization, exchange, domain specific, etc.) Nevertheless, one fundamental bias remained: it was designed with a document-style organization in mind.

Today we know that, although documents are important part of our global data, there is plenty of data that has no document-style organization: table data, raw data, sensor data, streams, images, etc. What is a "good" language for a global exchange of data? We would like to advance some basic general requirements for it:

1. Include codification for data, metadata and knowledge.

2. Be flexible enough to describe most types of data.

3. Be minimalist and efficient (regarding user needs and evaluation complexity).

4. Scale in a non-centralized form.

We will come back to the language theme repeatedly, because it is one of the cornerstones of the Web of Data.

1.2 The Semantic Web

A review of the Web would be incomplete without covering the Semantic Web. Indeed, the original project included as an ideal target a Web where all contents shared global semantics.

There are two driving forces behind the development of the Semantic Web: first, the fact that if data and information scale to meta-human levels, the only possibility to access, organize and manage such data is via machines; Second, the problem of meaning of information: what is the meaning of each piece of information on the Web? This has to do fundamentally with the semantics and meaning of concepts (even in the same language).

The first problem is an old one and is at the root of the discipline of databases on one hand, and of information retrieval on the other. One deals with structured data and the task of the organization of data –via logic– to allow semi-automatic querying and management of it. The other deals with unstructured data and documents, and relies on statistical methods to approximate the user needs.

The basic assumptions of classic database models (closed world, known goals, well defined users, etc.) do not scale at planetary level. The statistical approach has shown to be more suited to scale, but at the cost of trading logical precision by approximate results.

The Semantic Web aims to partially solve this problem based on the simple idea of organizing information at planetary level. The Semantic Web is *the Web of machine-processable data*, writes TBL, and this amounts to standardize meanings. Is this program viable? Naive approaches in this direction, like the Esperanto language, have failed miserably. Optimistically one could think that there were basic design failures in that project: centralized approach, lack (or high cost) of extensibility, not machine processable, complex semantics, little participation of (prospective) users in their enrichment.

The Semantic Web program devised two humble goals in order to overcome these problems:

1. *Develop languages* for describing metadata, sufficiently flexible, distributively extensible, machine-processable. (Note how this fits smoothly with the requirements for a global language for the Web discussed in a previous section). Two families of languages have been developed:

   (a) *Resource Description Framework, RDF* [42]. A basic language, in the style of semantic networks and graph data specifications, based on universal identifiers. Basic tools for interconnecting (linking) data, plus a lightweight machinery for coding basic meanings.

   (b) *The Web Ontology Language, OWL* [50]. A version of logic languages adapted to cope with the Web requirements. Composed of basic logic operators plus a mechanism for defining meaning in a distributed fashion.

2. *Develop an infrastructure* for it. Among the most important building blocks for the Semantic Web are protocols, query languages, specifications and applications for accessing, consulting, publishing and exchanging data.

Goal (1) has been a successful program. As time went on, two more or less defined communities have been developing this area (see Figure 2; Information Retrieval will not be discussed here):

The logic and knowledge representation community. It is oriented towards developing high level and expressive languages for describing information on the Web. One could summarize its accomplishments saying that it has achieved the internationalization and global extensibility of logic languages, particularly

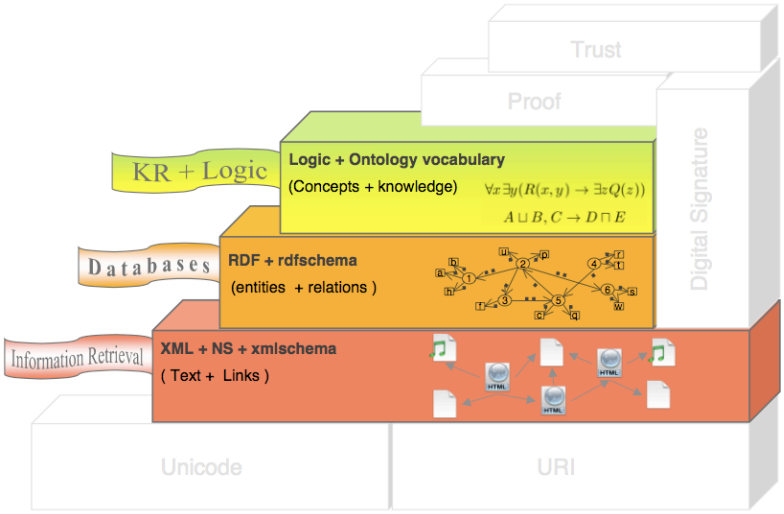

Figure 2: The technical fields involved in data aspects of the Semantic Web Tower: Information Retrieval, Databases and Knowledge Representation.

via OWL. At the same time, it is important to understand its limitations: this approach does succeed in describing data at a massive scale. In fact, the most basic tools have a computational complexity far exceeding the needs of big-scale data management.

The database community. It is centered on the development of RDF and its query language SPARQL. In the next section we will expand on these languages.

Goal (2) has been partially successful. On the positive aspects, there is a solid community behind the specifications and increasing interest by different stakeholders (e.g. governments, scientific communities). On the other hand, the area is still looking for applications that show the full potentialities of these approaches (an issue that deserves a careful analysis, beyond the scope of these notes).

2 Towards the Web of Data

Roughly speaking, the Web of data can be defined as follows:

> *The Web of data is the global collection of data produced by the systematic and decentralized exposure and publication of (raw) data using Web protocols.*

From this point of view, one of the main question arising is how to identify the changes produced to data management when incorporating (raw) data to the classical Web model reviewed .

In this section we will address the most visible initiatives aimed either, at enhancing and overcoming the Web of documents, or at addressing new challenges to information management posed by the new developments of data at Web scale. First, we will summarize the challenges posed by the "data deluge" on data management. Then, we will review the data level of the role and perspectives of RDF in this new setting. Finally we will discuss the contributions of two important projects, Linked Data and Open data to the goals of the Web of Data.

2.1 The Data Deluge structure

The data deluge described in the introduction consists of different types of data and data sources. Today there is a widespread feeling that this is beginning of a chaotic new era. I think it is important to realize that this tsunami of data is going to stabilize; that we should not act like the people shaken by the first big waves, but try to get a comprehensive picture of the process that is opening.

Any modeling of data on the Web should begin with a clear picture of the sources of such data. First of all, one has to consider the traditional publishing sources (editorials, writers, in general: sources that surely will remain, although in different formats) and scientific data that is gradually changing because of the increasing capacity and will to record and store. An important additional source of data are sensors, either capturing data directly from non-human natural processes (metereology, radioastronomy, animal behavior, etc.) or directed at humans (surveillance, logs of computer applications, medical, etc.)

Determining what types of data are more relevant is of paramount importance. The resources are finite, and hence naturally the most relevant data is the one that will constitute the main source of the sea of data.[48]

A characterization of the data itself (independent of its source) is another challenge. Classical methodologies and results about it (e.g. those of librarians) dealt essentially with human-produced data in natural language. Photography and video are still datasets that for most of us have hold little meaning outside of the human annotations (in natural language) attached to them. Clearly this is going to change.

2.2 RDF as infrastructure

It should not be a surprise that the notion of Web of data has a close relationship with the Semantic Web. The most influential semantic technologies, the RDF model and the SPARQL query language, have given new impetus to the development of the idea of Web of data. Here we will briefly explain the strengths of RDF and the coming challenges. (For RDF and SPARQL, consult [10].)

RDF was designed to facilitate automatic processing of information on the Web via metadata. The 1999 Recommendation stated it clearly: *"RDF is intended for situations in which this information needs to be processed by applications, rather than only displayed to people"*. Thus, it is at the core of its goal

the incorporation of machine readable information to the Web. But the design of RDF had another rather unexpected outcome: its graph nature (due to its triple structure) allows for representation of any type of data, and hence opens the door for converting the Web of documents into a Web of Data.

The power of RDF resides in the combination of two ideas: (1) a flexible model able to represent plain data as well as metadata in a uniform manner, pushing the idea of objects of information where data and metadata (schema) have the same status; (2) a graph structure that represents naturally, interconnections and relationships between data. In fact this latter feature is the one that led to the development of the Linked Data initiative.

These two ideas crystallize at the structural level of RDF in two main blocks of the RDF language: its (graph) data structure and its vocabulary.

Data structure RDF triples can be considered from a logical point of view as statements. But at the same time, they naturally represent a graph structure. Hence its expressive power: the structure really represents a linked network of statements.

This graph can be considered as relational data (a set of triples is a table with three columns). This viewpoint has the advantage of dealing with a well understood object; thus, allows the reuse of well studied and proven relational technology to manage such data.

It is important to understand the implications of the fact that RDF is a graph model: we face an object of study still not well understood, but with enormous potential to represent and model information [9].

Vocabulary RDF was designed to be flexible and extensive regarding vocabulary, allowing to give meanings to the relationships indicated by its graph structure. It has a few pre-defined (built-in) keywords with a light semantics (see [35]). The compromise here is the usual: the computational complexity of processing such data increases with the expressive power of its vocabulary semantics. Today one can roughly separate the vocabulary in three groups: (a) Having light (or no) semantics (essentially `type`, `subClassOf`, `subPropertyOf`, etc.); (b) RDF Schema plus some light extensions; (c) OWL, the Web ontology language. For linking and describing raw data, (a) seems to be enough.

A Remark concerning Blank Nodes. Blank nodes allow flexibility in structure data and representation of incomplete information. For a global model of information seems that these features are unavoidable. The problem, nevertheless, is that data with such features increases the computational complexity of processing and its semantics of querying is not simple [51].

2.3 Linked Data

Among the most successful world-wide projects addressing the problem of ubiquitous data on the Web, *Linked Data* stands out [44, 20]. This project originated in the practice of linking data and TBL's ideas on Web architecture [14], and

has become one of the main driving forces pushing the idea of exposing data on the Web. As the authors of the project state [44]:

> *Linked Data is about using the Web to connect related data that wasn't previously linked, or using the Web to lower the barriers to linking data currently linked using other methods. More specifically, Wikipedia defines Linked Data as "a term used to describe a recommended best practice for exposing, sharing, and connecting pieces of data, information, and knowledge on the Semantic Web using URIs and RDF".*

The idea is simple: thanks to the Web technologies, the possibility to produce, publish and consume data (not only documents in the form of Web pages) has become universally available. These processes are being done by different stakeholders, with different goals, in different forms and formats, in different places. Taking full advantage of this new scenario is a challenge. One of the main problems –the one addressed by the Linked Data project– is that this universe of data is not interlinked meaningfully.

The relevance of the Linked Data project has been eloquently expressed TBL [15] as follows:

Linked Data allows different things in different datasets of all kinds to be connected. The added value of putting data on the Web is given by the way it can be queried in combination with other data you might not even be aware of. People will be connecting scientific data, community data, social web data, enterprise data, and government data from other agencies and organizations, and other countries, to ask all kinds of interesting questions not asked before.

Linked data is decentralized. Each agency can source its own data without a big cumbersome centralized system. The data can be stitched together at the edges, more as one builds a quilt than the way one builds a nuclear power station.

The Linked Open Data movement uses open royalty-free standards from W3C. These do not bind governments nor agencies to any specific supplier.

A virtuous circle. There are many organizations and companies who will be motivated by the presence of the data to provide all kinds of human access to this data, for specific communities, to answer specific questions, often in connection with other data from different sites.

The TBL's "five-stars" test to measure the level of implementation of these ideas demonstrates the strategic goal of the Linked Data project:

1. Make your stuff available on the web (whatever format).
2. Make it available as structured data (e.g. excel instead of image scan of a table).
3. Use non-proprietary format (e.g. csv instead of excel).
4. Use URLs to identify things, so that people can point at your stuff.

5. Link your data to other people's data to provide context.

The Linked Data project has rapidly earned solid support among developers and governments (e.g. [25]), and is slightly gaining space in Academia [45]. The applications of database techniques to it, particularly the development of an infrastructure for querying and navigating such network, are just taking off [37, 38]. There is a recent book by Heath and Bizer [39] that covers the area systematically.

2.4 Open Data

Open data is a movement towards facilitating both, the *production* and *dissemination* of data and information at global scale.[2] In this regard, it is closely related with the original goals of the Web project. Because of its relationship with the issues arising in the "public versus private" sphere, it has become influential in management of information in Government and big organizations. On the other hand, regarding it as the data version of similar movements for software, we can define it as follows:

> *Open data is a movement whose goal is to develop and spread open standards for data.*

The big question here is what does openness mean for data. We will follow here the methodological approach of Jon Hoem in his study of openness in communication [41], and adapt the discussion to data. There are several possible dimensions from where one can consider openness. Three important ones are: the content, the logical and the physical levels. For data, this means respectively: semantics; datatypes and formats; and applications to access the data. For communication Hoem isolates two other crucial parameters: control of production and re-use; and control of distribution and consumption.

People working with public (government) data are among the ones that have elaborated more on this subject. Early in 2007, eight principles for openness in data were proposed [62]. Although they refer to "public data", the principles offer good insights into the requirements for open data: (1) *Be Complete:* All data is made available. (Restrictions: valid privacy, security or privilege limitations); (2) *Be Unprocessed:* Data is as collected at the source, with the highest possible level of granularity, not in aggregate or modified forms; (3) *Be Timely:* Data is made available as quickly as necessary to preserve the value of the data; (4) *Be Accessible:* Data is available to the widest range of users for the widest range of purposes; (5) *Be Machine processable:* Data is reasonably structured to allow automated processing; (6) *Be Non-discriminatory:* Data is available to anyone, with no requirement of registration; (7) *Be Non-proprietary:* Data is available in a format over which no entity has exclusive control; (8) *Be License-free:* Data is not subject to any copyright, patent, trademark or trade secret regulation. Reasonable privacy, security and privilege restrictions may be allowed.

[2] Usually Open data refers to open "information", understanding information as data apt for direct human consumption. For the discussion in this section, the distinction is not relevant.

	satellite	medical	surveillance	historical	leisure
Use	Closed [O]	Closed [C]	Closed [C]	Any	Any
Re-use	Closed [O]	Closed [C]	Closed [C]	Any	Any
Access	Closed [O]	Closed [C]	Closed [C]	Any	Any
Prodn.	Closed [C]	Closed [C]	Open [O]	Closed [C]	Any

Table 1: Examples of data openness for images. We show openness allowed by current socio-economic model and in brackets intrinsic openness of the application. (C=closed; O=open; "Any"= both models are possible.)

A more systematic set of parameters characterizing data can be obtained from an analysis of the cycle of (digital) data. For our purposes, the following four basic processes in that cycle give a good first approximation:

1. *Production:* producing data from the physical world; production of bits (writing, sensors, music, images, etc.)
2. *Access:* possibility of of getting (copying, locally storing) digital data.
3. *Use:* final (terminal) consumption of the data (can be thought of as "returning" the bits to the physical world).
4. *Re-use:* producing data using other data (already produced).

Each of these processes can have restrictions (be closed) or be available to everyone (be open). In Table 1 we show examples of the behavior of these parameters for some types of images. Note that they permit us to discriminate between some basic types of images. Note also how external factors like the socio-economic impact the openness criteria in some cases (e.g. although the satellite images could have open access, use and re-use in a future world, it is unrealistic to imagine that everybody could produce them).

Table 2, on the other hand, indicates current policies on openness for their data of some paradigmatic repositories and applications. Note how databases are closed in all four criteria. For the new "data enterprises", whose essential driving force –and business model– is to get and process data of other people like Google, Yahoo!, Facebook, Twitter, etc.; it is crucial to enforce and open model of production of data while keeping a closed model for access, re-use and use.

These perspectives on data production and consumption necessitate new requirements and pose new challenges to a Web model. The impulse to develop "open" data models has disclosed a number of activities that were, either considered as "given", or did not gain the prominence they have today. Among them: open digital windows to existing data; availability of digital data; linkage of data; building of an infrastructure for data.

	library	broadcasting	database	Web2 appl.	Web page
Use	Open	Closed	Closed	Closed	Open
Re-use	Closed	Closed	Closed	Closed	Any
Access	Closed	Closed	Closed	Closed	Open
Prodn.	Open	Closed	Closed	Open	Open

Table 2: Classical repositories and applications and their current policies on openness criteria for the data they hold. (C=closed; O=open; "Any"= both models are possible.) Web2 application signifies Web applications based on data-intensive processing: search engines, social networks, etc.

Several new requirements emerge at this point: preparing data; cleaning data; pre-processing (for publication) data; logical design of the internationalization (vocabularies, models, etc.); and at the physical dimension, availability, service, formats, etc.

3 Modeling data on the Web

The Web can be viewed from multiple points of view. In this section we examine ideas and abstract conceptualization of the Web. First we review the notion of data model. Then we briefly present the ideas and viewpoints that people have elaborated on regarding the "object" called Web. Then we study the most widespread concept of the Web, that is, a collection of documents. Next, we look at models and representations of data beyond documents. Finally, we describe the most comprehensive attempts to model the Web as a whole.

3.1 Data Models and their role

A data model is a set of concepts that can be used to describe the structure and operations of a database for a given domain [57], where database is defined as a collection of data with the following properties:

1. Represent some aspect of the real (or an imagined) world, the "domain" of application.
2. Be logically coherent, i.e., the data has to have some common domain and must have some purpose.
3. Directed at an intended group of users (known in advance), and usually refined to preconceived applications.

The two last conditions define a clear difference of data found in classical data management applications and data on the Web.

3.2 The Web as information artifact

One could generalize the notions of database given above, by defining a data model (in general) as a set of concepts (a conceptual framework) that describes at an abstract level an information system or artifact. Examples of information systems are libraries, databases, tables, photo albums, etc. The Web also can be viewed as an information artifact, and hence devising models of it is pertinent.

In fact, many researchers have described, characterized, and even modeled, the Web for different purposes. The following list shows the most typical characterizations of it:

1. The Web is an abstract (imaginary) space of information. (Berners-Lee, [13].)
2. The Web is not a database. (Mendelzon, [53].) The Web is a large, heterogeneous, distributed collection of documents connected by hypertext links. (Mendelzon, Mihaila, Milo, 1996 [56].)
3. The Web is one huge database. (Asilomar Report, 1998, [19].)
4. The Web is a vast collection of completely uncontrolled heterogeneous documents. (Brin and Page, 1998, [23].)
5. The Web is a huge heterogenous distributed database. (Konopnicki and Shmueli, 1999 [59].)
6. The Web provides a simple and universal standard for the exchange of information. (Abiteboul, Buneman, Suciu, 2000, [5].)
7. The pages and hyperlinks of the Web may be viewed as nodes and edges in a directed graph. (Kumar et al, 2000, [43].)

3.3 The Web of documents

Perhaps the most clear expression of the most consolidated conception of the Web is the one that the creators of Google, Sergey Brin and Lawrence Page, gave in their well-known paper on Search Engines [23]: "The web is a vast collection of completely uncontrolled heterogeneous documents". Here the Web is defined by contrasting it with the world of "well controlled documents". This contrast nicely parallels the one found in databases between the Web and the world of closed and structured information.

In that paper, Brin and Page identify a core set of challenges to be addressed when dealing with information at Web scale:

1. Documents have extreme internal variation, in language, vocabulary, format, form of generation (human, machine).
2. External meta information. External meta-information was defined as information that can be inferred about a document but is not contained within it.

3. Things that are measured vary by many orders of magnitude.

4. Virtually no control over what people can put on the web; flexibility to publish anything.

5. The contrasting interests between "the enormous influence of search engines" and companies "deliberately manipulating search engines", with the user interests.

6. Metadata efforts have largely failed with search engines.

Let us extract the underlying view and characteristic that this influential design had: *heterogeneity in format and usage (items 1,4); the key idea that relationships between documents (networked data) is of fundamental importance (item 2); the understanding that scalability is a breaking point with the previous world of information management (item 3); and finally, the implicit assumption that search engines are the basic data access tools at Web scale (items 5,6).*

It is worth noting that multimedia and raw data do not play a special role in this model. On the other hand, their solution –which will be the solution implemented by an ample set of successful companies– is a centralized one. The user plays the passive role of consulting information. Brin and Page are essentially addressing the challenges of heterogeneity in content and massive access produced by the new scale. But overall, they are anchored in a Web of documents and centralized services oriented directly to a human user.

3.4 Models of data on the Web

Documents are at the heart of the classical Web. Its original language for specifying data was HTML, that although has facilities to represent data (via tables), has as primarily goal the representation and visualization of documents.

As the Web became popular, the need for better formats to represent more structured data was raised. Such a language had two basic requirements: (1) to be able to represent documents, the most popular information object on the Web (and in daily practice), and (2) to have some level of structuring, so to be able to be queried much like the well known and successful relational technology (SQL).

At the abstract level the answer was the notion of *semistructured data*. The guiding motivations of semi-structured data were the paradigm shift in data management produced by the advent of the Web and the new type of data [5, 21]. The main characteristic of this new type of data was that it was neither raw nor strictly typed: its structure is irregular, implicit, partial, with large, evolving and sometimes ignored schemas, and the distinction between schema and data is blurred [2]. Probably the most representative, abstract and minimalist model is OEM [61]. Grahne and Lakshmanan [31] slightly extend the OEM model to better capture the notion of data independence in these models.

As "real world" version of semistructured data emerged XML, that rapidly became a standard for exchanging data (more precisely: documents) on the

Web. XML has another important feature: it unifies in one information object the data and the metadata (traditionally split in classical databases). Despite its success, XML is a verbose format not designed to codify raw data.

The data format, JSON, considered by its followers as being a "fat-free alternative to XML", is a lightweight data-interchange format, with the goal of being "easy for humans to read and write and easy for machines to parse and generate". It has become popular to code data at Web scale by its flexibility and minimality. It resembles the OEM model.

In parallel to XML, the Web Consortium developed a standard for representing metadata. It is the RDF universal model of triples for representing data, metadata and knowledge on the Web. Its structure and flexibility to represent any kind of data, and moreover, to *link* (to establish relationships between) different datasets, the feature that has made RDF a prime candidate for representing data on the Web, and a candidate for base data format for the Web of Data.

In summary, we have today a universal syntax, on the lines of a minimal semi-structured model (XML, JSON, etc.) plus a model for describing and linking data (RDF).

3.5 Data models of the Web

The philosophy of the formalizations we have seen so far is to develop good data models applicable at Web scale. Something more elaborated is to model the Web itself as a whole. Indeed, jointly with the explosion of the Web of documents, researchers have been trying to model the Web as a huge data system. We would like to call the reader's attention to two of the most interesting such attempts.

Abiteboul & Vianu's model Abiteboul and Vianu [3, 4] presented a model of the Web which is more sophisticated than a graph. They assume that the characteristics of the Web –departing from traditional notions of databases– are its global nature and the loosely structured information it holds.

They model the Web as an infinite set of semistructured objects over the relational schema {*Obj(oid), Ref(source, label, destination), Val(oid, value)* }, where *oid* is an identifier of objects (URIs), *Ref* specifies a finite set of labeled arcs, and *Val* specifies the value of an object. (The reader can see how the triple model emerges again here.) Intuitively objects are Web pages, the value is the content of a page, and references are links. The model –departing from the traditional database notions– enrich the notion of computable query. Considerer the following simple query: list all links that point to my page. This query is not computable because we do not have global information on links. The formalization given is based in a slight generalization of the classical notion of computability according to the new scenario. They introduce the notion of *Web machine*, that essentially is a Turing machine dealing with possibly infinite inputs and outputs. Based on this machine, the notion of query on the Web is formalized.

The model explores only basic aspects of querying and computing on the Web, leaving out, among others: communication costs; the notion of locality; the essentially distributed nature of the Web; the fact that queries on the Web are intrinsically concurrent processes; updates; and the fact that users often seem to be satisfied with incomplete and imprecise answers.

Mendelzon & Milos's model Almost concurrently with Abiteboul and Vianu's, Mendelzon and Milo [53, 54, 55] introduced another model, assuming that the Web is not a database (mainly due to the lack of concurrency control and limited data access capabilities). The central difference with Abiteboul and Vianu's model is the infinite character of the Web. The Web is huge, but finite at any given moment, state Mendelzon and Milo. Second, the infinity assumption blurs the distinction between intractable and impossible. For example, the query "List all pages reachable from my page", in an infinite model is not computable, but in Mendelzon and Milo's is in principle computable, although intractable. The formalization is done via a Turing machine with an oracle, which simulates the navigational access from a set of URIs to the Web graph it spans. They present results on computability of queries on the Web, and introduce a Web query language, which is a generalization of the seminal WebSQL query language that integrates data retrieval based in contents, structure and their topology [56].

Mendelzon and Milo's model does not address heterogeneity of data, degrees of autonomy among users, and lack of structure. Also it is restricted to the static case, that is, it ignores updates.

The importance of these works is that they introduce the notion of a data space with peculiarities that are just emerging: the practical impossibility of accessing all data; the intrinsically distributed nature of updating and querying data; heterogeneity of data; etc. This and other open issues reaffirm the need of a model including all or most of these features.

4 Requirements for the Web of Data

In this section we will explore several parameters that play relevant roles in the data system underlying the Web. The general philosophical principles of the Web as declared by TBL continue to be the basis of this new artifact. There is no claim of completeness, nor theory behind them. They are presented with the goal of sparking ideas and motivating the identification of relationships between different views.

4.1 Architectural views

The most comprehensive discussion of Web architectural principles is Fielding's thesis [27]. Departing from the classic Web, several ideas have been developed, in different directions making them, strict sensu, incomparable.

In Table 3 we put together three influential such models, just to let the reader grasp similarities and differences.

	Classic Web	RESTful Web	Web of Data
Access Tool	Navig/Search Eng.	Web Service	SPARQL, endpoints
Language	HTML	XML, JSON	RDF
Access Protocol	HTTP	HTTP 1.1	HTTP "++"
Data primitive	URI	URI	URI

Table 3: A rough comparison of architectural styles. The Web of Data uses the existing background of the Web, and should enhance it to support massive exchange querying of data. The language for logical specification of data and metadata is the RDF model (syntax is not relevant here). The access is semi-automatized via the SPARQL query language.

The definitive architecture of the Web of Data is yet to be designed, but should include several facets besides the ones shown in Table 3. In particular, enhancements of the current access/put protocol for data on the Web.

4.2 Static versus Dynamics

The classic models assume that the Web is an essentially static object. The models of Abiteboul & Vianu and Mendelzon & Milo speak of a non-dynamic Web. Also the models of the Web as a graph share the same implicit assumption. By dynamics we mean not only the addition of new data, or deletion of old data, but also modifications of it. It is useful to exemplify this difference: a library is static, in the sense that it incorporates books, disposes books, but does not create nor change them. The same happens today with image collections. On the contrary, a database management system is essentially a dataset that is constantly being modified by applications. Its essence is the volatile data that is created and modified constantly. Table 4 shows a classification of information artifacts when crossing the dynamics and the openness parameters.

	Static Data	Dynamic Data
open world	Data Govs	Classic Web Web of Data
closed world	Libraries Archives	Dataspaces Databases Desktops

Table 4: Dynamics versus openness. A rough classification of some information artifacts and projects based on these dimensions. Where should projects like Linked Data and Open Data be classified?

A closely related issue, *transactionality*, is a notion inseparable from classical data management and its dynamics. Gray defines a transaction as "a transfor-

mation of state which has the properties of atomicity (all or nothing), durability (effects survive failures) and consistency (a correct transformation). The transaction concept is key to the structuring of data management applications." [29] Does this notion make sense at Web scale? Is it consistent with Web principles?

4.3 Data Access methods

For the common user, access to data on the Web is accomplished either by navigation or by filling in forms. These methods do not scale. For Web volumes of data, semi-automatic and automatic methods are necessary. Figure 5 shows the menu of most common access methods available today.

	human	semi-automatic	automatic
non-structured	Navigation	Search engine	Statistical techniques
structured	Forms	Query language	API, Web serv., Endpoints

Table 5: The most popular current methods to access data on the Web. The Web of Data currently points to structured and automatic retrieval of data.

As for query and transformation languages (prime methods when working with massive data), Table 6 shows the most "popular" access languages dealing with data on the Web.

	Keyword	**SQL**	**XQuery**	**SPARQL**
application	text	spreadsheets	documents	statements
abstract data	strings	tables	Trees	Graphs
data format	nat. lang.	SQL table	XML	RDF
technique	statistics	algebra/logic	autom./logic	algebra/patterns

Table 6: Most popular semi-automatic data access approaches

4.4 Cost models

Any model for the Web of Data has to include a corresponding cost model for accessing and exchanging data, and even for data itself [48]. There is need for common *cost models* to evaluate information on the Web [26]. Many of the models proposed include cost models for accessing and exchanging data; nevertheless there is yet not a common approach to compare them. The need to explore and incorporate ideas from other areas (e.g. classical cost models, economic ones, response time models, communication complexity models, etc.) As large data companies have advanced in this area (see e.g. [52]), it is now important to devise models for the open world.

4.5 Incomplete and partial information

The ability to deal with incomplete or partial information is one of the basic requirements for a model of the Web of Data.

There are several database developments that partially address this issue. The most natural one is the theory of incomplete information. A whole area of research has spun off from this subject since the seminal work by Lipski [49]. This research has been partially subsumed in the area of *Probabilistic Databases* [32, 63]. Theories of incomplete information deal with unknown and uncertain information, whereas probabilistic databases can be considered as numerical quantification of the uncertainty. The models mainly follow the ideas of the possible-worlds approach. Even though the Web has other facets that escape these models, they are valuable starting points.

The theories above deal essentially with the problem of how to code partial information and query it. A different perspective presents itself when trying to model the user of the Web, who can only get partial information from the network of data that constitutes the Web. This approach overlaps with the problem of the behavior of agents having bounded capacity and bounded information. Theories like *bounded rationality* [58] are worth exploring here.

4.6 Organizing data

As we learned, the RDF graph model is a good candidate for a universal format for representing data and their relationships on the Web. Applications like social networks, or projects like Linked data, are increasingly showing success in this task. With these we are seeing the distributed construction of a huge network of data. Although it is possible to find partial classifications of such data, they resemble that of the imperial encyclopedia imagined by Borges where the animals were divided into categories as follows: (a) belonging to the emperor, (b) embalmed, (c) tame, (d) sucking pigs, (e) sirens, (f) fabulous, (g) stray dogs, (h) included in the present classification, (i) frenzied, (j) innumerable, (k) drawn with a very fine camelhair brush, (l) etcetera, (m) having just broken the water pitcher, (n) that from a long way off look like flies.[3]

A natural question arises: does the task of organizing the network of data on the Web make sense? Note that the task is not impossible in principle and that librarians succeeded in organizing data coded with human languages. The experience of tagging and folksonomies is valuable, but sheds little light on the problem of organization of structured data at Web scale. Most of the challenges in this regard are still open, even at small scale (cf. the experience of graph data models [9]).

There is more. The challenges posed by the growing amount of data known as *multimedia* (images, videos, scans, etc.) is something that any model of the Web of Data should address. Until today they have been treated as collections of black boxes whose descriptions are done by tagging, with little and poor

[3] J. L. Borges, *The Analytical Language of John Wilkins*, Translation of Lilia Graciela Vázquez.

	poor services	good services
no central control	Web	Peer to Peer
central control		Distributed DB

Table 7: A rough classification of some data systems according to their distributed nature and the intrinsic quality of their services (cf. Bernstein et al. [16]).

additional metadata, and no relationships among their "contents". Although this is not the place to discuss this topic in depth, it is important to call attention to the crucial role it will play in the Web of Data.

5 Other relevant related areas

A discussion of models for the Web of Data would not be complete without mentioning other areas of research which are closely related to this goal. In this section we briefly address the most relevant of them.

5.1 Distributed data management

A *distributed database* is one that has a central control, but whose storage devices are not all attached to a common central server, that is, they are stored in multiple computers, in the same physical location or over a network of computers.

The notion of distribution (of tasks, of people, of data, etc.) is intrinsic to the Web, hence there being several characteristics from this model that are common to Web phenomena. The commonalities among distributed databases, P2P systems and the Web can be established as shown in Table 7 [16]. Without doubt, the P2P approach is the most interesting and fruitful source of ideas in this regard.

Peer to Peer P2P systems became popular with Napster, Gnutella and Bit Torrent. The model of P2P has two characteristics which made it one of the closest in spirit with the Web principles: (1) The sharing of computer resources by direct exchange, rather than requiring the intermediation of a centralized server; and (2) The ability to treat instability and variable connectivity as the norm, automatically adapting to failures in both network connections of computers, as well as to a transient population of nodes [8].

Gribble et al. [33] enumerate the following principles as general characteristics of the P2P model:

1. No client/service necessary: each peer is a provider or a consumer. Everybody more or less has the same role, with the same duties and rights.

2. No central control. In particular each agent decides to enter/be part of or leave/abandon the network at his/her convenience.

3. Exchange of large, opaque and atomic objects, whose content is well described by their name. Large-granularity requests for objects by identifier.

Valdurriez and Pacitti [70], studying data management in large-scale P2P systems, indicate the main requirements for such systems:

1. *Autonomy.* Peers should be able to join or leave the system at any time, and control the data it stores.

2. *Query Expressiveness.* Allow users to describe data at the appropriate level of detail.

3. *Efficiency.* Efficient use of system resources: bandwidth, computer power, storage.

4. *Quality of Service.* Completeness of query results, data consistency, data availability, query response time, etc.

5. *Fault-tolerance.* Efficiency and quality of services should be provided despite the occurrence of peer's failures. Given the nature of peers, the only solution seems to rely on data replication.

6. *Security.* The main issue is access control (including enforcing intellectual property rights of data contents).

Wide Distributed Systems The project Mariposa [67] is an influential proposal for developing architectures for distributed systems at large, that is, working over wide networks. The main goal is to overcome the main underlying assumptions on the area, that in their opinion, do not apply to wide-area networks (and less to the Web): Static data allocation; Single administrative structure; and Uniformity. The guiding principles of the new design are the following: Scalability to a large number of cooperating sites; Data mobility; No global synchronization; Total local autonomy; and Easily configurable policies.

Dataspaces *Dataspaces* [36] is another abstraction for information management that attempts to address the "data everywhere" problem. It focuses on supporting basic functionalities of data management, such a keyword searching for loosely integrated data sources and relational-style querying for more integrated ones. Currently the project has not included the publishing-of-data agenda.

5.2 Logic Approaches

Under the logical framework there have been some works that model aspects of the Web. Let us show a few examples just to give a flavor of the possibilities and scope of this approach.

Himmeröder et al. [47] propose to use F-logic to model knowledge on the Web, particularly Web queries. The model, though, is just a graph of documents with arcs representing hyperlinks, where they concentrate on a language to explore the Web. From another perspective, Terzi et al. [68] present a constraint-based logic approach to modeling Web data, introducing order and path constraints, and proposing a declarative language over this model. The basic assumptions, though, are the same as those of the semi-structured model.

More recently, Datalog, the classic logic query language, has been the object of attention by the people working in distributed systems, and suggested as model for the Web. We would like to call the attention to two interesting ongoing projects in this direction. One is being developed by Joseph Hellerstein and his group [40], and centers on "data-centric computation" motivated by the urgency of parallelism at micro and macro scale. They develop an extension of Datalog that, relying on the time parameter, addresses the fundamental issues of distribution. The other project is headed by Serge Abiteboul [71], which also uses Datalog to specify the problems of distribution. As stated in their project, "the goal is to develop a universally accepted formal framework for describing complex and flexible interacting Web applications featuring notably data exchange, sharing, integration, querying and updating".

6 Concluding Remarks

The massive production and availability of data at world scale due to the technological advances of sensors, communication devices and processing capabilities, is a phenomena that is challenging the classic views on data management.

The Web has become the premium infrastructure to support such data deluge. Designed originally as a worldwide interrelated collection of documents, and oriented primarily to direct human visualization, today the Web is rapidly incorporating the data dimension and evolving towards automatic handling of such volumes of data.

The challenges for computer scientists are immense, as long as the new scenario involves data management, knowledge management, information systems, Web protocols, user interfaces, Web engineering, and several other disciplines and techniques. To have a unified and consistent view of the data dimension at Web scale one necessarily must have a model of such Web of Data. All indications point to the fact that such a model should follow the original Web principles of decentralization, distribution and collaborative development, and depart from small-scale and closed views of data and knowledge management that have been deployed until today.

In these notes we tried to present an introductory overview of the themes

and techniques arising in such program for the development of a Web of Data.

References

[1] S. Abiteboul, D. Quass, J. McHugh, J. Widom, J. L. Wiener, *The Lorel query language for semistructured data*, International Journal on Digital Libraries, 1 (1), 1997.

[2] S. Abiteboul, *Querying semi-structured data*, International Conf. on Database Theory-ICDT'97, 1997.

[3] S. Abiteboul, V. Vianu, *Queries and Computation on the Web*, ICDT 1997.

[4] S. Abiteboul, V. Vianu, *Queries and Computation on the Web*, Theor. Comput. Sci. 239(2), 2000.

[5] S. Abiteboul, P. Buneman, D. Suciu, *Data on the Web. From Relations to Semistructured Data and XML*, Morgan Kaufmann Publ. California, 2000.

[6] R. Agrawal et al., *The Claremont Report on Database Research*, 2008. `http://db.cs.berkeley.edu/claremont/`

[7] L. G. Alex Sung, N. Ahmed, R. Blanco, H. Li, M. Ali Soliman, D. Hadaller, *A Survey of Data Management in Peer-to-Peer Systems*, Web Data Management, 2005.

[8] St. Androutsellis-theotokis, D. Spinelis, *A Survey of Peer-to-Peer Content Distribution Technologies*, ACM Surveys, Vol. 36, No. 4, Dec. 2004.

[9] R. Angles, C. Gutierrez, *Survey of Graph Database Models*, ACM Computing Surveys, Vol. 40, No. 1, 2008.

[10] M. Arenas, C. Gutierrez, J. Perez, *Foundations of RDF Databases (Tutorial)*, Reasoning Web Summer School, 2009.

[11] T. Berners-Lee. *WWW: Past, present, and future.* IEEE Computer, 29(10), Oct. 1996.

[12] T. Berners-Lee, *Commemorative Lecture The World Wide Web - Past Present and Future. Exploring Universality.* Japan Prize Commemorative Lecture, 2002 `http://www.w3.org/2002/04/Japan/Lecture.html`

[13] T. Berners-Lee, *Frequently asked questions*, `http://www.w3.org/People/Berners-Lee/FAQ.html`

[14] T. Berners-Lee, *Design Issues/Linked Data*, `http://www.w3.org/DesignIssues/LinkedData.html`

[15] T. Berners-Lee, *Linked Open Data. What is the idea?*, `http://www.thenationaldialogue.org/ideas/linked-open-data`

[16] P. A. Bernstein, F. Giunchiglia, A. Kementsietsidis, J. Mylopoulos, L. Serafini, I. Zaihrayeu, *Data Management for Peer-to-Peer Computing: A Vision*, WebDB, Workshop on Databases and the Web, 2002.

[17] G. Bell, J. Gray, A. Szalay, *Petascale Computational Systems: Balanced CyberInfrastructure in a Data-Centric World*, Computer, Volume 39, Issue 1, January 2006.

[18] G. Bell, T. Hey, A. Szalay, *Beyond the Data Deluge*, Science, Vol. 323, March 2009.

[19] Ph. Bernstein, M. Brodie, S. Ceri, D. DeWitt, M. Franklin, H. Garcia-Molina, J. Gray, J. Held, J. Hellerstein, H. V. Jagadish, M. Lesk, D. Maier, J. Naughton, H. Pirahesh, M. Stonebraker, J. Ullman, *The Asilomar report on database research*, ACM SIGMOD Record, Volume 27, Issue 4, Dec. 1998.

[20] Ch. Bizer, T. Heath, T. Berners-Lee, *Linked Data - The Story So Far*, International Journal on Semantic Web and Information Systems, 3 (2009).

[21] P. Buneman, *Semistructured data*, ACM PODS, 1997.

[22] T. Bray, J. Paoli, C. M. Sperberg-McQueen, C. M. 1998. *Extensible Markup Language (XML) 1.0*, W3C Recommendation 10, (February 1998). `http://www.w3.org/TR/1998/REC-xml-19980210`

[23] S. Brin, L. Page *The anatomy of a large-scale hypertextual Web search engine*, Computer Networks and ISDN Systems, 1998.

[24] M. Cai, M. Frank, *RDFPeers: a scalable distributed RDF repository based on a structured peer-to-peer network*, Proc. WWW'04, 2004.

[25] *DATA.gov project*, `http://www.data.gov/`

[26] O. Erling, I. Mikhailov, *Towards Web Scale RDF*, 4th International Workshop on Scalable Semantic Web Knowledge Base Systems (SSWS2008), 2008.

[27] R. T. Fielding, *Architectural Styles and the Design of Network-based Software Architectures.* Doctoral dissertation, University of California, Irvine, 2000. `http://www.ics.uci.edu/~fielding/pubs/dissertation/top.htm`

[28] R. T. Fielding, R. N. Taylor, *Principled design of the modern Web architecture*, ACM Trans. Internet Technol., vol. 2, 2, May 2002.

[29] J. Gray, *The Transaction Concept, Virtues And Limitations*, Proceedings of 7th VLDB, Cannes, France, 1981.

[30] A. Szalay, J. Gray, *Science in an Exponential World*, Nature, Vol. 440, March 2006.

[31] G. Grahne, L. V. S. Lakshmanan, *On the Difference between Navigating Semi-structured Data and Querying It*, Research Issues in Structured and Semistructured Database Programming, LNCS Vol. 1949 / 2000.

[32] T. Green, V. Tannen, *Models for Incomplete and Probabilistic Information*, EDBT Workshops, Munich, Germany, March 2006.

[33] S. Gribble, A .Halevy, Z. Ives, M. Rodrig, D. Suciu, *What Can Databases Do for Peer-to-Peer?*, WebDB, Workshop on Databases and the Web, 2001.

[34] T. Guan, L. Saxton, *A complexity model for web queries*, Fundamentals of Information Systems, Ch. 1, Kluwer, 1999.

[35] S. Muoz, J. Prez, C. Gutierrez: *Simple and Efficient Minimal RDFS*. J. Web Sem. 7(3): (2009).

[36] A. Y. Halevy, M. J. Franklin, D. Maier, *Principles of dataspace systems*, PODS 2006.

[37] O. Hartig, C. Bizer, J.-C. Freytag, *Executing sparql queries over the web of linked data*, Proc. ISWC '09, 2009.

[38] M. Hausenblas, M. Karnstedt, *Understanding Linked Open Data as a Web-Scale Database*, 1st Internat. Conf. on Advances in Databases, 2010.

[39] T. Heath , Ch. Bizer, **Linked Data: Evolving the Web into a Global Data Space**, Synthesis Lectures on the Semantic Web: Theory and Technology, Morgan & Claypool, 2011. Online version: `http://linkeddatabook.com/editions/1.0/`

[40] J. M. Hellerstein, *The Declarative Imperative. Experiences and Conjectures in Distributed Logic*, SIGMOD Record, vol. 39, No. 1, March 2010.

[41] J. Hoem, *Openness in Communicaton*, First Monday, Volume 11, Number 7, 3 July 2006. `http://firstmonday.org/htbin/cgiwrap/bin/ojs/index.php/fm/article/viewArticle/1367/1286`

[42] G. Klyne, J. Carroll, *Resource Description Framework (RDF) Concepts and Abstract Syntax*, W3C Recommendation, 2004. `http://www.w3.org/TR/2004/REC-rdf-concepts-20040210/`

[43] R. Kumar, P. Raghavan, S. Rajagopalan, D. Sivakumar, A. Tompkins, E. Upfal, *The Web as a Graph*, Proc. PODS 2000.

[44] LinkedData Project, `http://www.linkeddata.org`

[45] *Workshops and academics events on Linked Data*, `http://linkeddata.org/calls-for-papers`

[46] M. Loukides, *What is data science?*, `http://radar.oreilly.com/2010/06/what-is-data-science.html`

[47] R. Himmeröder, G. Lausen, B. Ludäscher, Ch. Schlepphorst, *On a Declarative Semantics for Web Queries*, DOOD'97, LNCS 1341 / 1997.

[48] M. Lesk, *Encouraging Scientific Data Use*, Posted on *The Fourth Paradigm* on Feb 7, 2011. `http://blogs.nature.com/fourthparadigm/2011/02/07/encouraging-scientific-data-use-michael`

[49] W. Lipski Jr., *On Databases with incomplete information*, Journal of the ACM (JACM) JACM Homepage archive Volume 28 Issue 1, Jan. 1981.

[50] D.L. McGuinness, F. van Harmelen, *OWL Web Ontology Language Overview*, W3C Recommendation 10 February 2004, `http://www.w3.org/TR/owl-features/`

[51] M. Arenas, M. Consens, A. Mallea, *Revisiting Blank Nodes in RDF to Avoid the Semantic Mismatch with SPARQL*, W3C Workshop: RDF Next Steps, Palo Alto, CA, 2010.

[52] J. Madhavan, Sh. R. Jeffery, Sh. Cohen, X. Dong, D. Ko, C. Yu, A. Halevy, *Web-scale Data Integration: You Can Only Afford to Pay As You Go*, Proc. of Third Conference on Innovative Data System Research (CIDR 2007), 2007.

[53] A. O. Mendelzon, *The Web is not a Database*, Workshop on Web Information and Data Management 1998.

[54] A. O. Mendelzon, T. Milo, *Formal Models of Web Queries*, Proc. PODS 1997.

[55] A. O. Mendelzon, T. Milo, *Formal Models of Web Queries*, Inf. Syst. 23(8): 1998.

[56] A. O. Mendelzon, G. A. Mihaila, T. Milo, *Querying the World Wide Web*, Intl. Journal of Digit. Libr., 1 (1997).

[57] Sh. B. Navathe *Evolution of data modeling for databases*, Communications of the ACM, Volume 35 Issue 9, Sept. 1992.

[58] A. Rubinstein, *Modeling Bounded Rationality*, MIT Press, 1998.

[59] D. Konopnicki, O. Shmueli, *Bringing Database Functionalities to the WWW*, The World Wide Web and Databases, LNCS 1590 / 1999.

[60] No SQL, `http://nosql-database.org/`

[61] Y. Papakonstantinou, H. Garcia-Molina, J. Widom, *Object exchange across heterogeneous information sources*, 11th International Conference on Data Engineering (ICDE), 1995.

[62] *Seminar on Open Government data (Open Government Working Group)*, Dec 7-8, 2007. `http://resource.org/8_principles.html`

[63] D. Suciu, *Probabilistic Databases*, Database Theory Column, SIGMOD Record, 2008.

[64] M. Spielmann, J. Tyszkiewicz, J. Van den Bussche, *Distributed computation of web queries using automata*, Proc. PODS 2002.

[65] M. Stonebraker, S. Madden, D. J. Abadi, S. Harizopoulos, N. Hachem, and P. Helland, *The end of an architectural era: (it's time for a complete rewrite)*, Proc. VLDB '07, 2007.

[66] T. O'Reilly, *What Is Web 2.0*, `http://oreilly.com/web2/archive/what-is-web-20.html`

[67] M. Stonebraker, P. M. Aoki, W. Litwin, A. Pfeffer, A. Sah, J. Sidell, C. Staelin, A. Yu, *Mariposa: a wide-area distributed database system*, The VLDB Journal, Volume 5 Issue 1, January 1996.

[68] E. Terzi , M.-S. Hacid , A. Vakali , S. Hacid, *Modeling and Querying Web Data: A Constraint-Based Logic Approach*, Information modeling for internet applications book contents, 2003.

[69] J.T. Horng, Y.Y. Tai, *Pattern-based approach to structural queries on the World Wide Web* Proc. Natl. Sci, Counc. ROC(A). Vol. 24, No. 1, 2000.

[70] P. Valduriez, E. Pacitti, *Data Management in Large-scale P2P Systems*, VECPAR 2004.

[71] *Webdam Project*. Foundations of Web Data Management. `http://webdam.inria.fr`