

Web 3.0 - beyond the Semantic Web, a way to global SOA?

Seeing Web 2.0's advantages, we could ask from ourselves what could be the next shift? Try to imagine the following situation: You have not seen any new movies in a while and feeling all energetic, you make up your mind to go see a movie and have a late night dinner afterward. You are in the mood for some action adventure and an Italian delicacy. At first instant, you pull out your tablet, turn it on, open a web browser and immediately search for cinema, movies, and restaurant information. Without knowing what movies are showing in cinemas near you, you spend time reading short descriptions about movies which fall under action adventure before deciding. You sometimes even watch trailers about each movie showing to help make your choice easier. Although this might sway your decision in what movies you decide to watch-if there are less movies in the category you have decided-, you proceed anyway. Also, you may want to check for location, customer reviews and ratings for possible nearby restaurants. In all, you end up visiting several websites with a near or final conclusion in mind before heading out the door.

Some web experts are quite certain that Web 3.0-the next generation of the web after Web 2.0- will make such task like searching for movies or restaurants quicker, faster and easier. They believe that multiple searches will be a thing of the past and with complex search terms, the web can do the rest. Using the previous example, one could type "I would like to see an action adventure movie and then have dinner at an Italian restaurant. What possibilities do I have?" In this scenario, your response would be analyzed by the Web 3.0 browser after searching for all possible answers that match your criteria providing an organized search result for you.

Anyway, there is more to it. Most of these internet expert are certain that Web 3.0 browser will act like a personal assistant which is attentive in learning what one's interest is. They believe that the more you use your browser, the more your browser becomes knowledgeable about your questions. In the end, you might even be able to ask open questions to your browser such as "where is the best place for dinner nearby?" or "where is the best Italian restaurant in town" Looking up your records and taking into account your likes and dislikes, and also using your current location and geo-tagging, your browser would then suggest a list of possible nearby restaurants or eateries.

Graph Search - an other way of search

If you type in google what graph search is, you would most probably get results related to Facebook graph search. Though searching has been around for some time, it is not as natural as we would want. It is still dependent on keywords which results in articles on the web related or unrelated to our intended search. With this in mind, future web pioneers had to think outside the box. Facebook as a leading online presence moved a step further and developed Facebook graph search. To understand how graph search is different from normal searches, I'd like to shed more light on it.

Graph Search, popularly known as Facebook graph search, is a search engine combined with Facebook's social graphs. Using the search engine, natural language queries are processed from raw data and return information based on a user's network of friends, connections, or related information depending on the search. Current usage of graph search include but not limited to online marketing, job searches, common interest, dating, to name a few.

Below are a few examples;

- Most liked restaurants by friends living in Debrecen.
- Games fans of Harry Potter like.
- Debrecen alumni who like Titanic.

- Single ladies in Kassai utca.
- People in Debrecen who like Arsenal.

With graph search, several concepts of a search can be shared and correlated. These ties, which consist of search variables which are dependent on each other include education, hobbies, location, jobs, employer, marital status, gender, religion, interest and age. With graph search, organisations and individuals act as nodes which can be linked to one another.

With the idea of an emerging Web 3.0, the future looks promising for Facebook's graph search.

The Road to Web 3.0 through Web 2.0

Several jargon and internet buzzwords have made it to public consciousness and sub consciousness but of all these that evolved, Web 2.0 is by far the best known. Though most people may have heard of it in several ways than one, only a few have an idea what it really means. While several of those who have no idea of what it is all about suggest it is nothing more than a strategy but online marketers created to persuade venture capitalist (according to investopedia, "An investor who makes available capital either to startup ventures or supports small companies that wish to expand but do not have access to public funding") into investing millions of dollars into websites or startups. Without disputing the fact that Dale Dougherty of O'Reilly Media coined the phrase "Web 2.0" in 2004, there was never a decision if a "Web 1.0" existed.

Characteristics of Web 2.0 include but not limited to:

- Users and sometimes visitors have the ability to add some changes to webpages. Popular sites such as amazon, zappos, ebay allows shoppers to leave reviews about products. This helps future visitors get information that can be easily read.
- With the emergence of web 2.0, content can be easily share. Another example is youtube which allows users to create and upload a video to its site for visitors to watch.
- With a good internet connection, users who subscribe to websites can receive notifications via RSS (Really Simple Syndication) feeds.
- Access to the internet using handheld devices like smartphones and tablets. This way, internet has moved beyond mere desktop computers.
- Using interactive web pages to link people together thus bridging the gap of face to face meeting. Facebook-a popular social networking site- makes it easier for users to keep in touch with one other. It also helps users make new friends and find new friends too.
- Content which were inaccessible digitally is now easily accessible and available.
- With the emergence of "mashup" capability, users who are not professionals can create different applications using a mix of several software. Google maps is a popular example as it can be incorporated in different web applications and websites.

Moreover, think of Web 1.0 as the earlier stage of the World Wide Web which consisted of webpages that were connected by hyperlinks. Think of it as a source of information which information can be gotten but no change or contribution to such information is allowed. The exact definition of Web 2.0 has evolved over time, but with social networking and online interactions, Web 2.0 is focused on the ability of users to share and contribute information through social media, blogs, etc.

Folksonomy and Collabulary

As we listed in one of the Web 2.0's key principles, Folksonomies are the first step for the semantic version of the Web. Its an Internet-based Information Retrieval methodology or in other words, a collaboratively open-ended labels for categorizing content (webpages, photographs, links, etc.). The labels have a new

name: Tags, and labeling have Tagging. It could be threatened as people's classification management, where a folksonomy is accessible as a shared vocabulary which is familiar to its primary users.

It has several advantages, like dramatically lower categorization costs and quick respond to changes. Folksonomies are unsystematic, unsophisticated and open-ended (tags are created and applied on the fly). In spite of the various tagging abilities, the global process usually produces results comparable to the best professionally designed systems. Moreover, in enterprise level, the "emergent enterprise taxonomy" created by the employees could be seen easily.

However, there are disadvantages as well. The criticism shows several problems with

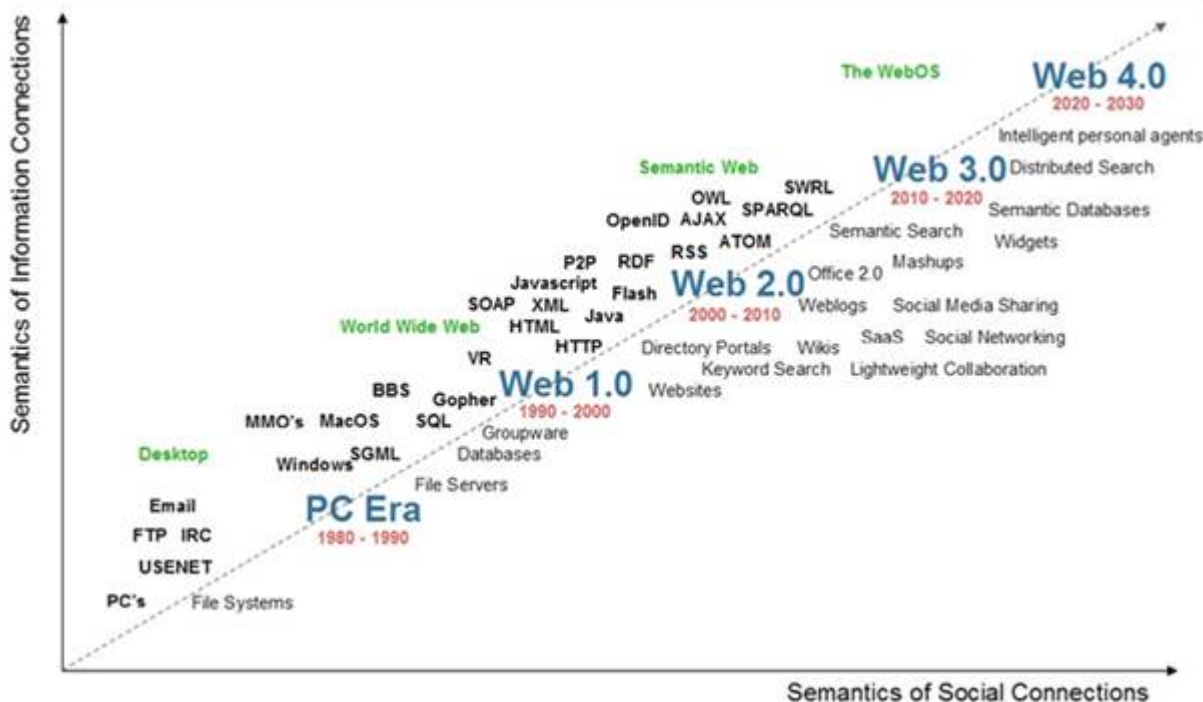
- * polysemy (words with multiple meaning),
- * synonyms (words with the same or similar meaning).

Over these, there are other factors, like plural words between the tags, and the "meta noise" which decreases the system's information retrieval with the false tagging.

The solution is a compromise between folksonomies and taxonomies (controlled vocabularies). This is the Collabulary. A collabulary arises similarly to what a folksonomy does but is developed collaborating with domain experts. It could avoid errors that inevitably arise in native, unsupervised folksonomies.

Basics of Web 3.0

Even though most users have not yet gotten a grasp on what Web 2.0 is about, others are already thinking ahead trying to figure out what comes next. There have been several questions to which we do not know the answer but yet they are still being asked. What exactly will Web 3.0 have that would separate it from Web 2.0? Would it be different from how we use the web today? Would it happen like a boom that we would not even notice it has already begun?



Timeline of the Web

[Source: solutions.wolterskluwer.com/blog](https://solutions.wolterskluwer.com/blog)

People with extensive understanding about the internet believe that Web 3.0 would be like having an assistant who knows almost practically everything about a person with answers to unrestricted information stored on the internet. These experts also believe that even though Web 2.0 connects people using the internet, Web 3.0 will use the information from the internet to make the connection. However, questions

have often been asked as to whether Web 3.0 will replace the current web as we have it or if it would exist separately.

As complicated as this concept may sound, an example may shed more light. Everyone loves to pamper themselves once in a while. Maybe going to the sauna or taking a nice vacation so as to watch relax on the beach and watch the sun rise and set. As usual, you set aside a budget of €1,500 for your vacation. Your desire is to find a flight deal, a nice restaurant and a comfortable place to stay without entirely spending all of your budget.

Currently, all you have to do is a little or sometimes more research to find the best holiday options available. You would research destinations, holiday deals, cheap flights, and budget meals until you find the right option that best suits you. Often, you may browse hotels, hostels, and car rentals as presented by several search engines. The entire search process of getting a holiday that satisfies your needs may take a few hours, days or weeks depending on your effort.

However, internet experts believe that Web 3.0 will let you sit back while the internet does all the work. The painstaking process of Web 2.0 will be a thing of the past. Using a search engine, Web 3.0 will be able to narrow down your search by gathering your data, analyzing those data and presents the data back to you in such a way that compares it quickly. With this, Web 3.0 will be intelligent enough to understand information from the web.

Currently, a search engine on the web, for example google, is not quite intelligent enough to be able to understand search. What it actually does is browsing through millions of webpages that contain relevant keywords related to your search terms. Search engines are unable to differentiate relevant or irrelevant webpages related to your search. What it does is display a webpage with a keyword in your search term. For example, if you typed in a search engine the word "Carina", your results would be from webpages about the car "Toyota Carina" or names, products, and things bearing Carina.

It is believed that with a Web 3.0 search engine, not only would you find related keywords but the context of your request would be interpreted as well. Using our holiday destination as an example, if you do decide to search for "Budget holiday for a week under €1,500", aside just displaying the results, information about restaurants, or upcoming activities related to your search might also be displayed in a Web 3.0 browser. The entire internet would not only be treated as an information centre but also as a massive database.

Approaches to Web 3.0 - APIs, SOA and semantics

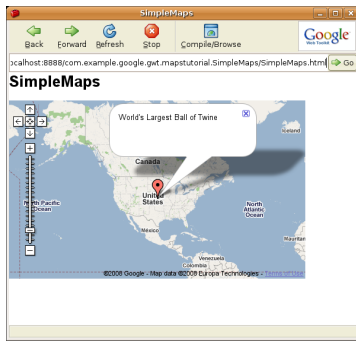
The future of the web is unknown to all. Most web experts believe that the experience will be relevant as well as enabling users have a distinctive profile. In this case, such distinctive profile will be based on individual's browser history. With this concept, it would be easier to modify searches to suit every individual. They believe that if two users performed that same search using the same search term, both would get different result based on their individual profiles. It is slightly similar to graph search but they are not the same.

Presently, technologies for applications such as these are not yet mature. They are still in production and testing phase. Most still use a trial and error method which is not as efficient as what the future of Web 3.0 is intended to be. Experts believe that Web 3.0 will be founded on **Application Programming Interface (APIs)** to achieve a global SOA (Service-Oriented Architecture).

What is an API?

An **API** can be defined as an interface designed in a way that allows developers to create applications and take advantage of a certain set of resources. Many Web 2.0 sites include APIs which present developers with access to certain capabilities and unique site's data. Popular social networking site Facebook allow developers to use their API for reviews, games, etc.

Among all current trends of the web as we have come to know it, mashup seems most likely the trend that would aid in the fast development of Web 3.0. A mashup is the process of combining two or more applications making them become one. An example is combining google maps with a hotel review site. Now this new site would not only show reviews about hotels but also display their locations for visitors to see.



An example of mashup using API

What is SOA?

The SOA term becoming widely used, but there's not a lot of precision in the way that it's used. There are at least two different approaches what we could identify. At first, the technological side which is a very technical perspective in which architecture is considered a technical implementation. The World Wide Web Consortium (W3C) for example refers to SOA as

A set of components which can be invoked, and whose interface descriptions can be published and discovered

Second, a much more general interpretation, based on the Component Based Development and Integration (CBDI) forum's definition:

The policies, practices, frameworks that enable application functionality to be provided and consumed as sets of services published at a granularity relevant to the service consumer. Services can be invoked, published and discovered, and are abstracted away from the implementation using a single, standards-based form of interface. (CBDI)

It highlights that SOA is much more than just an architectural pattern, it is a style. It highlights that any form of service can be exposed with a Web services interface. However higher order qualities such as reusability and independence from implementation, will only be achieved by employing some science in a design and building process that is explicitly directed at incremental objectives beyond the basic interoperability enabled by use of Web services. The CBDI forum advises that we need to think on it as a framework to understand what constitutes a good service. Two obvious sets could be identified here:

- Interface related principles—Technology neutrality, standardization and consumability.
- Design principles—These are more about achieving quality services, meeting real business needs, and making services easy to use, inherently adaptable, and easy to manage.

Business management and IT management may have a better understand the cost and benefits after they realized what are the difference when a system is not designed for this purpose. It is important to know if a service is to be used by multiple consumers, (as is typically the case when a SOA is required), the specification needs to be generalized, the service needs to be abstracted from the implementation (as in the earlier dotcom case study), and developers of consumer applications shouldn't need to know about the underlying model and rules.

If a service is SOA enabled we can say it is:

- reusable
- abstracted
- formal
- relevant
- published

With SOA it is critical to implement processes that ensure that there are at least two different and separate processes—for provider and consumer. For the consumer, the process must be organized such that only the service interface matters, and there must be no dependence upon knowledge of the service implementation. If this can be achieved, considerable benefits of flexibility accrue because the service designers cannot make any assumptions about consumer behaviours. For the provider, it needs to develop and deliver a service that can be used by the Service Consumer in a completely separate process. The focus of attention for the provider is therefore again the interface—the description and the contract.

CBDI concludes that there are three major process areas which we need to manage:

- **The process of delivering the service implementation.**
 - 'Traditional' Development
 - Programming
 - Web Services automated by tools
- **The provisioning of the service—the life cycle of the service as a reusable artefact.**
 - Commercial Orientation
 - Internal and External View
 - Service Level Management
- **The consumption process.**
 - Business Process Driven
 - Service Consumer could be internal or external
 - Solution assembly from Services, not code
 - Increasingly graphical, declarative development approach
 - Could be undertaken by business analyst or knowledge worker

This process view that we have seen is a prerequisite to thinking about the type of architecture required and the horizons of interest, responsibility and integrity. From the architectural point of view, Service-oriented architecture is a design pattern that consists of discrete pieces of software with some functionalities and other applications can utilize these functionalities. This pattern does not require us to use some specific product or a platform. A service provides some functionality and this can be used by other large software applications to complete its use. A service is the base of this architecture. Services are the structures which have ability to interact with each other. In other words they are the listener of the other side of the phone which is an endpoint. In the classical layered architecture layers interact with each other. This interaction and hierarchy of this system should be constructed very properly for this system to work proper. Therefore we can say that SOA simply means that these layers are created as services. For example if we make a service which provides us the data, then we have accomplished the functionality of data layer by this. Afterwards we can also use this service from other applications as well. Achieving this we have a flexible architecture to use.

However, not all experts agree. Some however do believe that Web 3.0 will start from scratch. They believe that it would rely on some new programming language and not HTML. They suggest that starting from the scratch would be a lot easier than following the current trend of Web 2.0. The man responsible for the web as we have come to know it has a different theory of what he feels the future of the World Wide Web will be like. He refers to it as a semantic web and his work is mainly referenced when experts talk about Web 3.0. But then again, what exactly do we mean by a semantic web?

One aspect for the Web's future: Semantic Web

According to semanticweb.org,

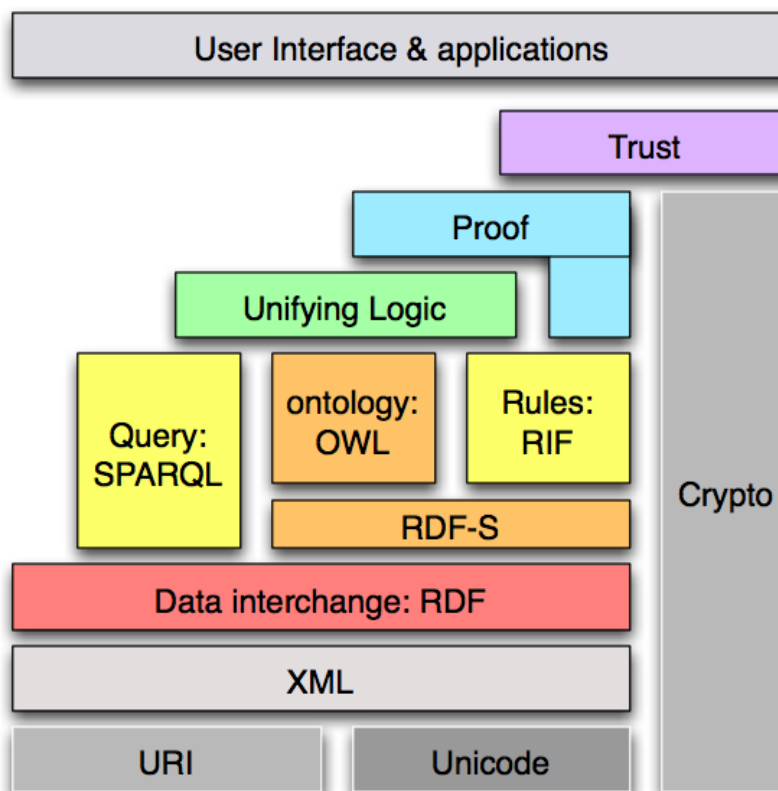
The **Semantic Web** is the extension of the World Wide Web that enables people to share content beyond the boundaries of applications and websites.

However, w3.org also defines it as

The **Semantic Web** provides a common framework that allows **data** to be shared and reused across application, enterprise, and community boundaries. It is a collaborative effort led by W3C with participation from a large number of researchers and industrial partners. It is based on the Resource Description Framework (RDF).

The World Wide Web was invented by Tim Berners-Lee in 1989. This happened 20 years after the first connection was established over what we know today as the internet. At that time, Tim worked as a scientist at the European Organization for Nuclear Research (CERN) and realized that it was difficult for scientists to share information with one another. With this discovery, the need for potential computers to be connected together was born. He disputes that Web 1.0 or Web 2.0 is nothing more than a meaningless jargon by maintaining claims that the intention of the World Wide Web is to do everything that both Web 1.0 and Web 2.0 is supposed to do.

The **semantic web** is a vision of what Tim Berners-Lee future of the web is supposed to be like. Presently, the web is structured for humans and not computers. Computers are unable to interpret information on the web but with semantic web, computers can interpret information on the web using software agents crawling around the web searching for relevant information. In Berners-Lee concept, semantic web would consist of metadata which would make the task of search by computers intelligent and more comprehensive. The most well-known figure describing the SemanticWeb's layered architecture is the following (from the talk of Tim B.-L. in 2006) :



Semantic Web's layered architecture

Before going any further, I would like to walk us through.

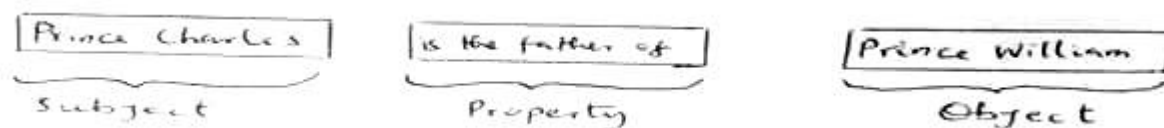
Why use Semantic Web?

With our previous example of searching for a movie, we have come concluded that most of the work done would be by the user. However, with Semantic web search, your choice would be directly entered and the web searches for the best choice to suit your need. In this case, the web would be intelligent enough to know your preferences and if you did have a bad experience from a particular theatre or restaurant, it would not use it when compiling your search result.

In this case, search in Web 3.0 would not be done by reading descriptions or reviews like a person would do but by a thorough search through its **metadata** that defines what the Web needs to present your data. According to whatis.com "**Metadata** is defined as data which describes other data. It is also a prefix used in information technology which means "an underlying definition or description"". Metadata sums up basic information about data thus making it easier to find and work with specific occurrence of data. For example, a metadata for a document would include title, author, file size, date created, etc. In webpages, metadata contain descriptions of the page's content as well as keywords linking the content to the page.

In Semantic Web, experts believe that the metadata of a page are not visible while the page is being read but visible to computers. Metadata, according to Tim Berners-Lee would let the web become a giant database.

XML, RDF and URI



A simple relationship analogy

From the photo above, it is easier to figure out what the sentence "Prince Charles is the father of Prince William". We know that they are both from the royal family, and there exist a relationship between them. We can also tell that father is a parent, and that William is the son of Charles. It is easier for humans but not so easy for the computer to understand such text. To help the computer understand what such sentence means, machine-readable information describing who Prince Charles and Prince William are needs to be added to make their relationship clearer. The tools needed are the **eXtensible Markup Language (XML)** and **Resource Description Framework (RDF)**.

XML is a markup language used to create common information formats, share the format and data on the web. It is a standard way of describing data that enables a user to request a program, gather data and make comparison. Its a markup language and may seen similar for HTML at first sight, they need to be separated. While HTML used to describe the content of a web page and how it is to be displayed and interacted with, XML describes the content - maybe for a web page - in terms of what data is being described. XML and HTML are used together in many web applications and sometimes, XML may appear within an HTML page.

RDF on the other hand, does exactly what its name suggest. It provides a framework which describes resources using XML tags. An RDF description, sometimes referred to as "data about data" can include the authors of the resource, date of creation, updating, sitemap, information that describes content in terms of audience or content rating, keywords for search engines, subject categories, etc. By identifying the resource, using the above diagram as an example, the computer would not confuse Prince William with Prince Harry or Princess Diana, or with other members of the royal family.

To achieve this, RDF makes use of **triples** written as XML tags to express such information. These triples are the **subject, property and object**. Very often, people call these the subject, predicate and object. Presently, RDF exists in the web and it is part of the RSS feed creation.

Benefits of RDF

Some likely benefits of RDF include but not limited to;

- By providing a consistent framework, RDF will encourage the providing of metadata about Internet resources.
- Because RDF will include a standard syntax for describing and querying data, software that exploits metadata will be easier and faster to produce.
- Applications will be able to exchange information more easily due to its standard syntax and query capability.
- Searchers will get more precise results from searching, based on metadata rather than on indexes derived from full text gathering.
- Intelligent software agents will have more precise data to work with.

In the above example, the computer understands that there are there exist a relationship between the two objects. However, it does not know what the objects are nor how they relate to one another. Let's see how these objects relate to one another.

Identifying resources: URI

With the framework that XML and RDF provide, a computer still needs a very direct, specific way of understanding what these resources are and to do this, RDF uses **Uniform Resource Identifiers (URIs)** to direct the computer to a document (object) that represents the resource. The most common form of URI is the Uniform Resource Locator (URL), which begins with `http://`.

According to Margret Rouse,

A URI is the way you identify any of those points of content, whether it be a page of text, a video or sound clip, a still or animated image, or a program. The most common form of URI is the Web page address, which is a particular form or subset of URI called a Uniform Resource Locator ([URL](#)).

An example of URI is http://www.w3.org/Images/WWW/w3c_main.gif

The above web address identifies a file that can be accessed using the Hypertext Transfer Protocol ("`http://`") which resides on a computer named "www.w3.org" that can be mapped to a unique Internet address. The computer's directory structure indicates that the file is located at the pathname of `/Images/WWW/w3c_main.gif`.

Character strings that identify File Transfer Protocol FTP addresses and e-mail addresses are also URIs (and, like the HTTP address, are also the specific subset of URI called a URL). Other sources define a **Uniform Resource Locator (URL)** as "the unique address for a file that is accessible on the Internet".

Another kind of URI is the **Uniform Resource Name (URN)**. A URN is a form of URI that has "institutional persistence," which means that its exact location may change from time to time, but some agency will be able to find it. Very often, people get confused when speaking about URL and URN. A URN functions like an identity or person's name, while a URL resembles that person's street address. In other words, the URN defines an item's identity, while the URL provides a method for finding it.

More exactly, as defined in 1997 in RFC 2141, URNs were intended to serve as persistent, location-independent identifiers for web resources, allowing the simple mapping of namespaces into a single URN namespace. The existence of such a URI does not imply availability of the identified resource, but such URIs are required to remain globally unique and persistent, even when the resource ceases to exist or becomes unavailable.

Example,

urn:issn:0167-6423

The scientific journal Science of Computer Programming, identified by its serial number.

urn:isbn:0451450523

The 1968 book The Last Unicorn, identified by its book

number.

urn:lex:eu:council:directive:2010-03-09;2010-19-UE

A directive of the European Union, using the Lex URN namespace.

Since RFC 3986 in 2005, the use of the term has been deprecated in favor of the less-restrictive "URI", a view proposed by a joint working group between the World Wide Web Consortium (W3C) and Internet Engineering Task Force (IETF). Both URNs and uniform resource locators (URLs) are URIs, and a particular URI may be a name and a locator at the same time.

URNs were originally intended to be part of a three-part information architecture for the Internet, along with URLs and uniform resource characteristics (URCs) as a metadata framework. However, URCs never progressed past the conceptual stage, and other technologies such as the Resource Description Framework (RDF) later took their place and became the "official language" of the Semantic Web but they cannot make the web accessible to the computer on their own. There is a growing need for more languages.

Semantic Web languages: RDFS, OWL and SKOS

One of the many obstacles of Semantic Web is that computers and humans do not have the same kind of vocabulary. We have used natural languages our whole life so it is almost completely easy to understand the connections between different words and concept and make a meaning out of it. Unfortunately for humans, we cannot just give the computer a dictionary, an almanac and a set of encyclopedia and let it learn all of it on its own. For the computer to understand words and the relationship between them, it needs to have **documents** describing all the words and **logic** in order to make required connections.

In the world of Schematic Web, this relationship come from **schemata** and **ontologies** which are the related tools which helps the computer in understanding human vocabulary. A schema is a method for organizing information while an ontology is the language that describes objects and the relationship between them. With RDF tags, a documents creator must declare which ontologies are referenced at the beginning of the document by providing access to schemata and ontologies included in these documents as metadata.

Below are the schema and ontology tools used on Semantic Web:

- **RDF Vocabulary Description Language Schema (RDFS)**

"RDFS is extending RDF vocabulary to allow describing taxonomies of classes and properties. It also extends definitions for some of the elements of RDF." (Maret Obitko, 2007). For example, the resource Prince William is a subclass of the class Royals. A property of Prince William could be handsome.

- **Simple Knowledge Organization System (SKOS)**

"is a W3C standard, based on other Semantic Web standards (RDF and OWL), that provides a way to represent controlled vocabularies, taxonomies and thesauri." (Juan Sequeda, October 31, 2012). A **controlled vocabulary** is a list of term agreed upon by an organization or community such as Monday, Tuesday, Wednesday are days of the week. A **taxonomy** is a controlled vocabulary organized in hierarchy such as Phones as concepts and smartphones or mobile phones as subclasses because they are both phones while a **thesauri** is a taxonomy with more information about each concept including preferred and alternative terms ("Phone" in English, "Telefon" in Hungarian).

- For example-in reference to SKOS-, with the Royal Family, a narrower term for Prince William could be William, while a broader term could be The Duke of Cambridge.

- **Web Ontology Language (OWL)**

extends RDF and RDFS and its primary aim is to bring the expressive and reasoning power of description logic to the semantic web. With OWL, new classes can be

constructed based on existing information. There are three levels of complexity in OWL namely Lite, Description Language (DL) and Full.

However, ontologies are very difficult to create, implement or maintain and depending on their scope, they can be enormous thus defining a wide range of concepts and relationships making it difficult for developers to focus on them and not logic or rules. Another pitfall for the semantic web would be the disagreement over what roles these rules play. Even so, some critics believe that such project is extremely impractical.

Going back to our original example about deciding on watching a movie, here is how the Semantic web could make the steps easier;

- Each site would have pictures and text so people can easily read them, and metadata so computers can read them describing available movies from different cinemas.
- Every metadata would make use of RDF triples and XML tags so that all attributes of the movie (like showing time, cast, and languages) would be machine-readable.
- Businesses would give the computer the vocabulary needed using ontologies to describe these objects and their attributes. Several e-commerce sites would use the same ontologies thus making all metadata a common language.
- Each cinema showing a movie would use appropriate security and encryption measures to protect customer info.
- All metadata found on different would be read by computerized applications and enable comparison of information and verify that the sources of such information is accurate and trustworthy.

In conclusion, much of Web 3.0 is far from reality but more of theory yet this has not stopped experts from guessing what might come next. Since the web is enormous, adding all this metadata to existing pages would be quite tasking

References

Bibliography

Web resources

[SOA] *Understanding Service-Oriented Architecture* - David Sprott and Lawrence Wilkes CBDI Forum. [Understanding Service-Oriented Architecture](#).

[Web 3.0] *World Wide Web 3.0*. [What is Web 3.0? What Will Web 3.0 Be Like?](#). [Web 3.0: The Third Generation Web is Coming](#). [A Brief Introduction to Web 3.0](#).

[Web 2.0] *World Wide Web 2.0*. [What is Web 2.0? - Definition from Techopedia](#). [What is Graph Search? - Definition from WhatIs.com](#). [What is metadata? - Definition from WhatIs.com](#). [Utilizing Web 2.0 in business](#).

[Semantic Web] *Semantic Web*. [W3C Semantic Web Activity Homepage](#). [W3C Semantic Web Activity Homepage](#). [semanticweb.org](#). [Semantic Web - W3C](#). [What is Resource Description Framework \(RDF\)? - Definition from WhatIs.com](#). [RDF Schema RDFS - Introduction to ontologies and semantic web - tutorial](#). [Introduction to: SKOS - Semanticweb.com](#). [Web Ontology Language OWL - Introduction to ontologies and semantic web - tutorial](#).

[Web 1.0] *The World Wide Web Consortium*. [History of the Web - World Wide Web Foundation](#). [Uniform resource identifier - Wikipedia, the free encyclopedia](#). [What is URL \(Uniform Resource Locator\)? - Definition from WhatIs.com](#).

[RIA] *Rich Internet applications*. [QuickStudy: Rich Internet applications](#). [About RIAs](#). [What is Rich Internet Application \(RIA\)](#).

