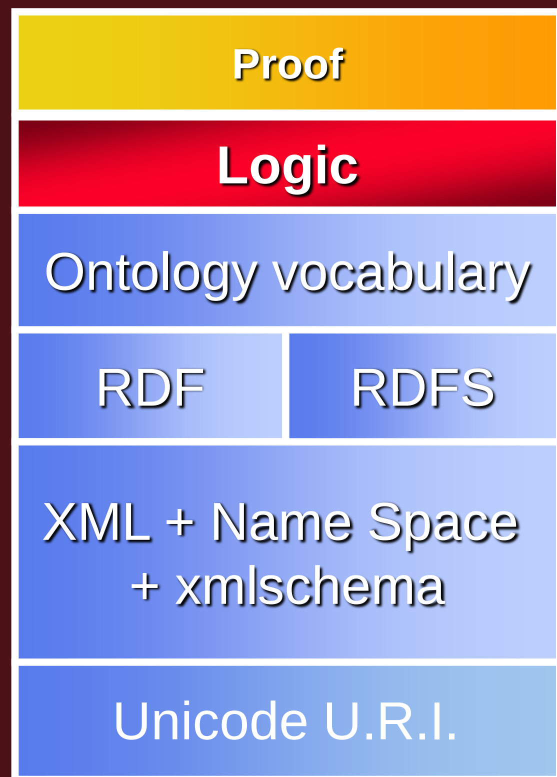
Introduction to the Web Semantic Architecture



SGML Origins

- SGML: Standard Generalized Markup Language
- It comes from GML, an IBM language 1960
- SGML is an ISO standard
- SGML was originally designed to enable the sharing of machine-readable documents in large projects in government, legal and industry (EDI).
- It has also been used in printing and publishing, but his complexity has prevented its widespread expansion.

Several other MarkUp langages derives from SGML

SGML

- **HTML (Web Publishing)** — One use only
- **DocBook (multimedia Publishing)** — Professional Book editing only
- **XML (general Purpose)** — Extensible use

Differences between HTML, DocBook, XML

- ***HTML (Hyper Text Mark-up Language): a language only designed to publish on Web.***

- ***DocBook : a language designed to describe a Book. It enables publication on Print, Web, PDF, etc…***

- ***XML (eXtensible Mark-up Language) : a meta language designed to express any vocabularys needed in application. XML is easier to implement than SGML, so it has replaced SGML almost completely. XML is used for general-purpose applications, such as the Semantic Web, XHTML, SVG, RSS, Atom, XML-RPC and SOAP.***

XML definition and use

Definition Scheme of the XML

```
<schema "http://.../XMLSchema"
   version="1.0">
<simpleType name="dayOfMonth">
   <restriction base="integer">
         <minInclusive value="1"/>
         <maxInclusive value="31"/>
   </restriction>
</simpleType>
```

Valid XML data according to the Schema

```
<dayOfMonth>
12
</dayOfMonth>
```

Invalid XML data according to the Schema

```
<dayOfMonth>
33
</dayOfMonth>
```

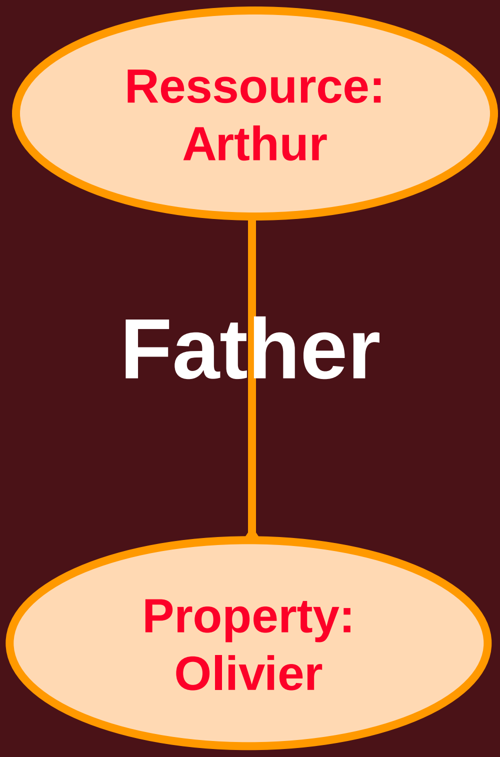
We want to define a type of data for the day of the month

Web Semantic Architechure
the basis Layer, XML

- ***XML permits abstracts expressions, sharable between different computers in different locations.***
- ***Technically, this features are permitted by three ways***
  - ***NS : Name Space and schemas which permit to define a set of vocabulary***
  - ***Unicode, which permit a comprehension of encoding on every computers.***
  - ***URI : Universal Resource Identifier, which permit to identify a resource on the Net***

Web Semantic Architechure the rdf layer, XML

We want to say to the system father(P,Y) which means father of « P » is « Y »



- ***RDF (Resource DescriptionFramework) is a way to express a <u>statement</u> between a <u>ressource</u> and a <u>property</u>:***

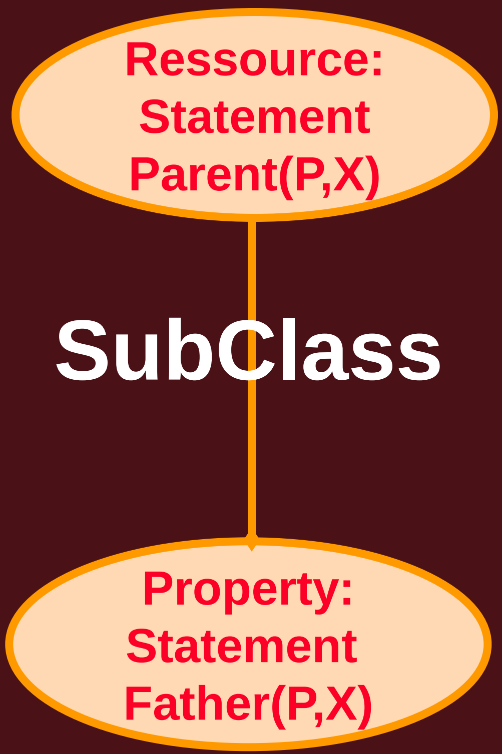
```
<rdf:RDF xmlns:rdf=« … » xmlns:myfamily=« … »>
    <rdf:Description
            rdf:about="http://www.family.picot/arthur">
                        <family:father>
                        Olivier Picot
                        </family:father>
    </rdf:Description>
</rdf:RDF>
```

Web Semantic Architechure
the rdf layer

- *In the previous exemple, we explain that the father of Arthur is Olivier. W’ve done this through an RDF syntax, so a computer can understand and treat this information.*

- ***RDF enables us to explain statements to the system.***
- ***This a new level of Web Semantic***

Web Semantic Architechure
the rdf schema layer, XML



With RDFS, we can assert relationships between statements like “is subclass of”. In our example, we assert that the Statement “Father “ is a subclass of statement “parent”

```
<rdfs:Class rdf:about="#father">
   <rdfs:subClassOf rdf:resource="#parent"/>
</rdfs:Class>
```

Then the system knows that

if “Olivier” is the father of “Arthur” then

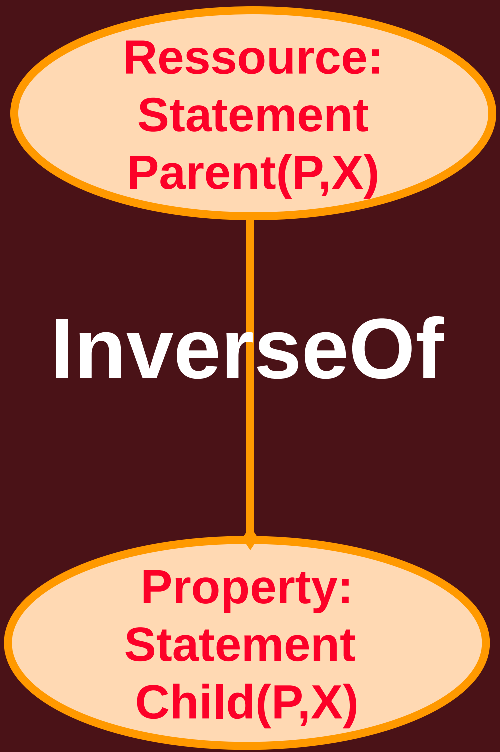
“Olivier” is a parent of “Arthur”

Web Semantic Architechure the rdfs layer

- *RDFS allows us to organize our statements and a lot of others relationships, that are not shown here*

- ***RDFS enables us to explain relationships between statements.***
- ***Then the system knows some new assertions(i.e. Olivier is the parent of Athur)***

Web Semantic Architechure
the ontology layer



- ***It's impossible to assert, in rdfs some kind of relationship between two Statements. To do this we have to use ontology langage: OWL***

```
<owl:ObjectProperty rdf:ID="Parent">
    <rdfs:range rdf:resource="#human"/>
    <rdfs:domain rdf:resource= "#family"/>
    <owl:inverseOf rdf:resource="#Child"/>
</owl:ObjectProperty>
```

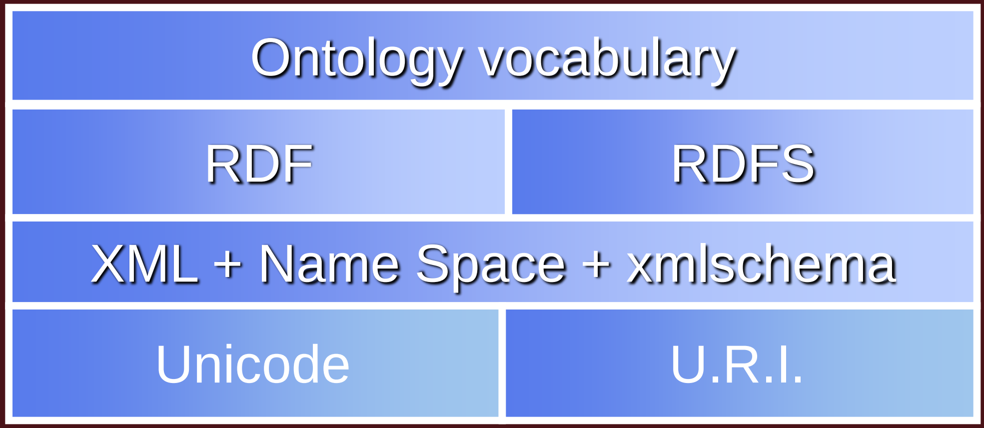
With this new relationship between Statements, the system can deduce that, if Arthur Parent is Olivier

Then Olivier Child is Arthur

« Well done !!! ;-) »

Web Semantic Architechure
the ontology layer (owl)

- **OWL allows us to assert some more complicated relationship than rdfs can do.**



- *OWL allow us to declare some refined relationship between statement and propertys, like:*
  - *Inverse*
  - *Equivalent*
  - *Restrictions*
  - *…*

Web Semantic Architechure
the logic and inference rules layer

- We dispose of those statements
  - S1: male(x) ⇔ x is a male
  - S2: father(P,x) ⇔ x is the father of P
  - S3: parent(P,x) ⇔ x is the parent of P
  - S4: notSame(X,Y) ⇔ x is not the same than Y
  - S5: brotherOrSister(X,Y) ⇔ x is the brother or the sister of y
- We can now assert some rules to the system
  - R1: male(X), parent(P,X) → father(P,X)
  - R2: father(P,X),parent(P,Y),notSame(X,Y) → mother(P,Y)
  - R3: parent(P,X),brotherOrSister(P,Q) → parent(Q,X)
  - R4: brotherOrSister(P,Q) → brotherOrSister(Q,P)

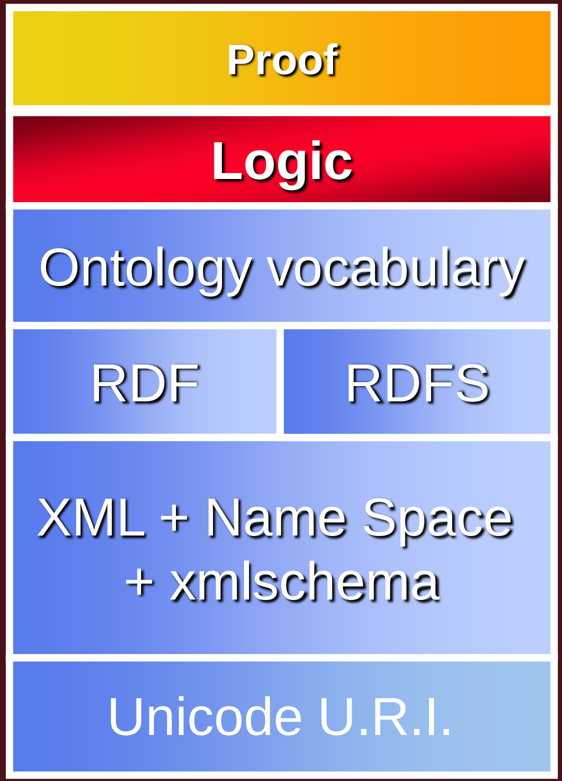
Web Semantic Architechure
the logic and inference rules layer

- ***Rules reminder:***
  R1: male(X), parent(P,X) → father(P,X)
  R2: father(P,X),parent(P,Y),notSame(X,Y) → mother(P,Y)
  R3: parent(P,X),brotherOrSister(P,Q) → parent(Q,X)
  R4: brotherOrSister(P,Q) → brotherOrSister(Q,P)
- ***If we assert to the system:***
  Parent(Arthur,Olivier)
  Male(Olivier)
  BrotherOrSister(Arthur, Leonard)
  Parent(Leonard,Celine)

Using rules, the system can deduce that

R1: male(Olivier), parent(Arthur,Olivier) → Father(Arthur,Olivier)
R4: brotherOrSister(Arthur,Leonard) → brotherOrSister(Leonard, Arthur)
R3:Parent(Leonard,Celine),
brotherOrSister(Leonard, Arthur) → Parent(Arthur,Celine)
R2:father(Arthur,Olivier),parent(Arthur,Celine),
notSame(Celine,Olivier) → Mother(Arthur,Celine)

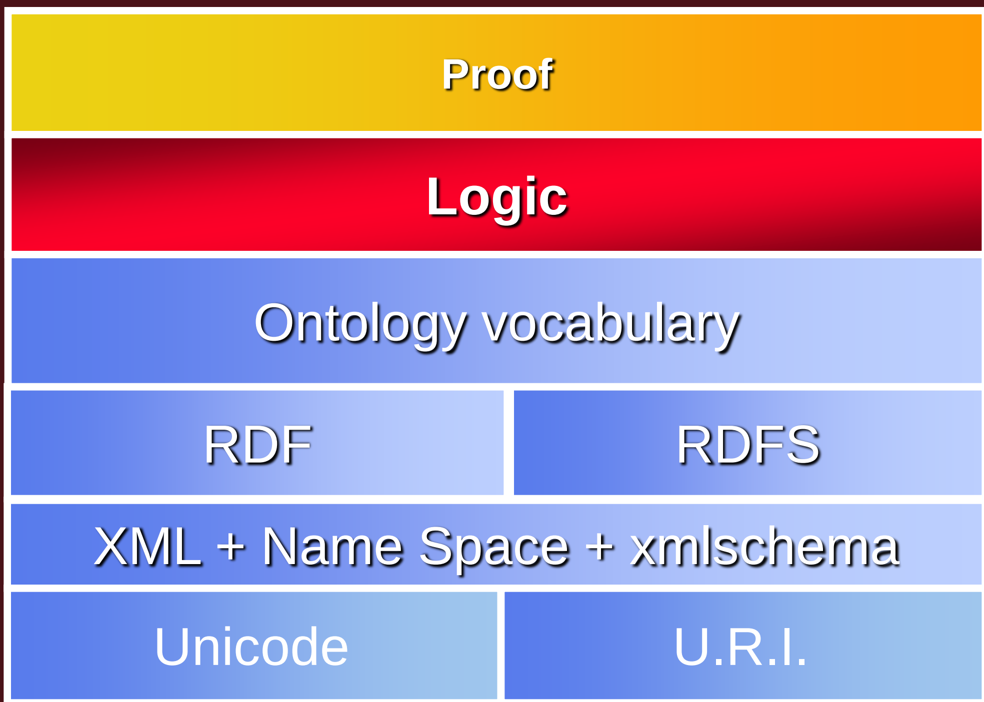
Web Semantic Architechure
Conclusion



- *We’ve quickly seen the tools corresponding to the different nivels of the semantic web architecture.*
- *If we organize safetly our indexes, or any kind of meta data, we can afterwards engage a process of asserting statements and rules, then starting a step of inducing and deducing new assertions*
- *In a context of web analysis, we prepare our data to future use.*
- *As you‘ve seen, AXIS is a way to do so*

Thank You for you attention

Web Semantic Architechure
the logic and inference layer



- ***Asserting some rules to the system, we can induce some informations afterward.***

This is the upper nivel of our architecture

Web Semantic Architechure Reference site

- *You see in English and in French a more detailed presentation of semantic web at:*

  - ***http://***

Please feel free to send us comments on this brand an new translation done by Titan