HMIN103 - Gestion des Données du Web

Langages pour données et schémas : XML & DTD

Federico Ulliana
UM, LIRMM, INRIA GraphIK

September 10, 2020

Slides collected from J. Cheney, S. Abiteboul, I. Manolescu, P. Senellart, P. Genevès, D. Florescu, D. Suciu, and the W3C.

Fonctionnement du Module

Fonctionnement du Module

- Responsables
  - Federico Ulliana, Pierre Pompidor `{ulliana,pompidor}@lirmm.fr`
- Groupes : 1 (DECOL), 2 (AIGLE)
- Cours : vendredi matin
- MCC : une partie pratique importante
  - 25% TP en binômes à rendre (F. Ulliana) (1 session) :
    1. Données et Schémas XML
    2. XPath et XQuery
    3. XML & Relationnel

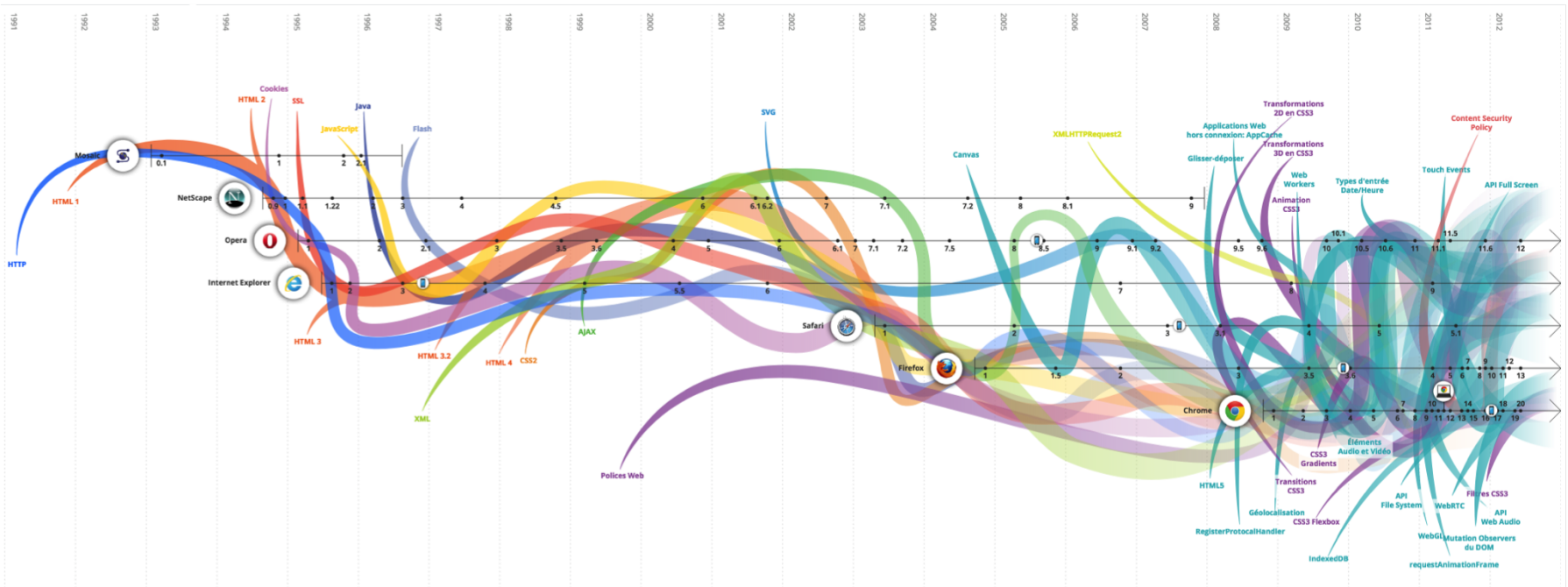
    » Moyenne des trois notes de TP.
    » Éval. : correction réponses, questions bonus, qualité redaction.
  - 25% Projet (P. Pompidor) (1 session)
  - 50% Écrit (2 sessions)
- Moodle : transparents, feuilles de TD/TP, etc...
- Accueil étudiants : rdv par mail

Gestion des données semi-structurées du Web

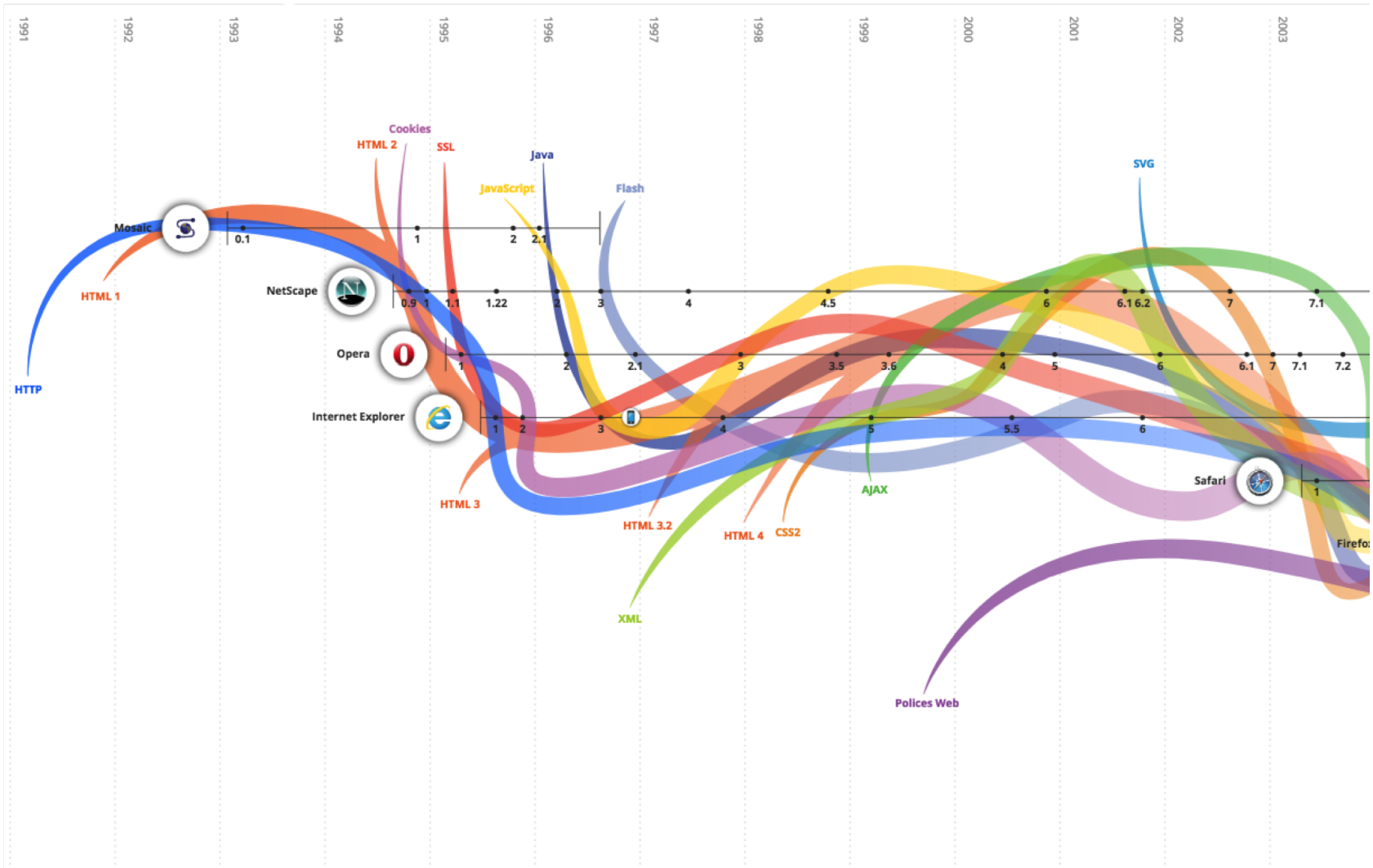
- Des fondements… (Federico Ulliana)
  - XML, DTD, XPath/XQuery, XSLT, JSON

- …aux applications (Pierre Pompidor)
  - HTML5/Node.Js, WebGL

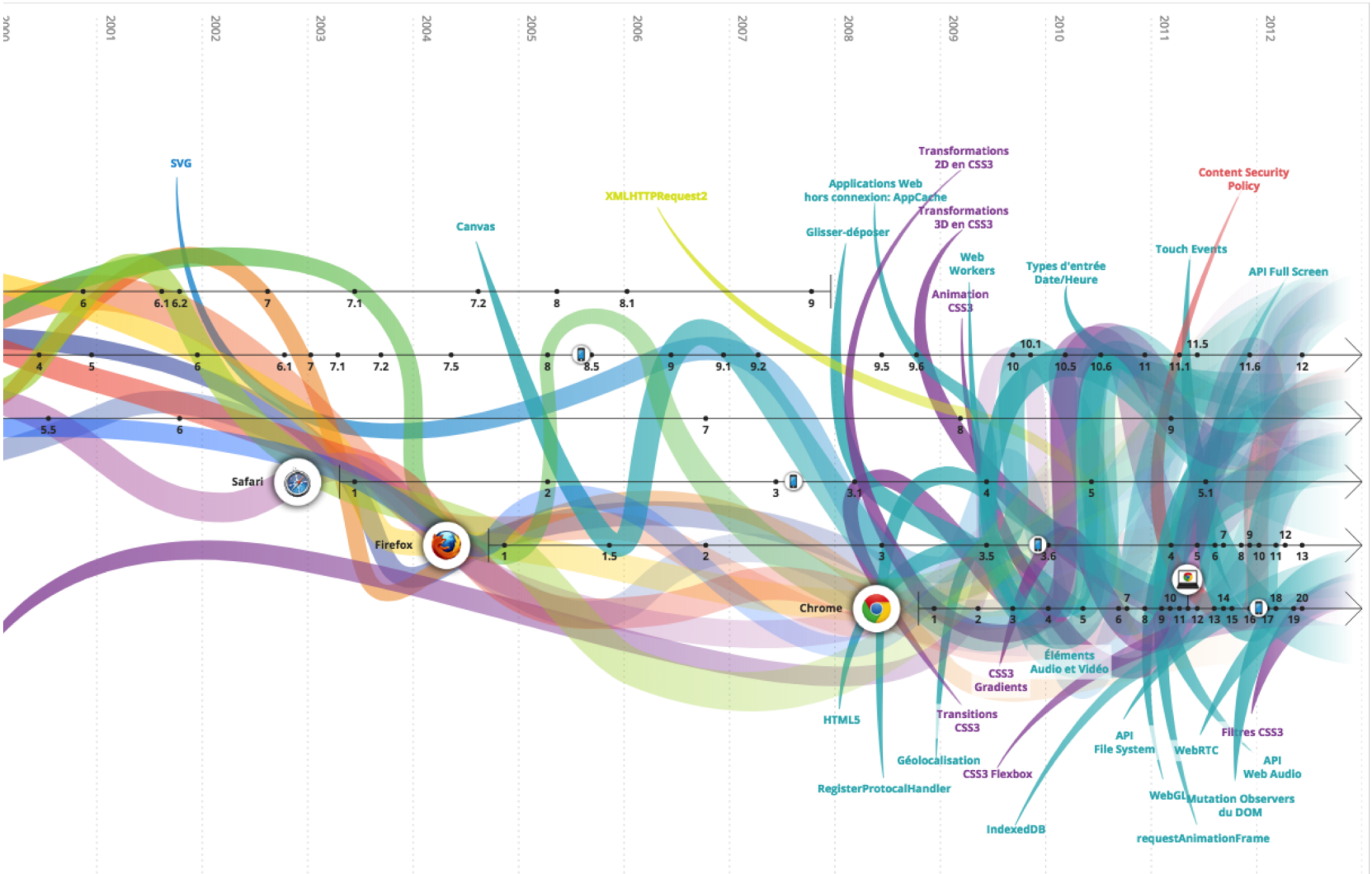
www.evolutionoftheweb.com



www.evolutionoftheweb.com - Partie 1 du Module



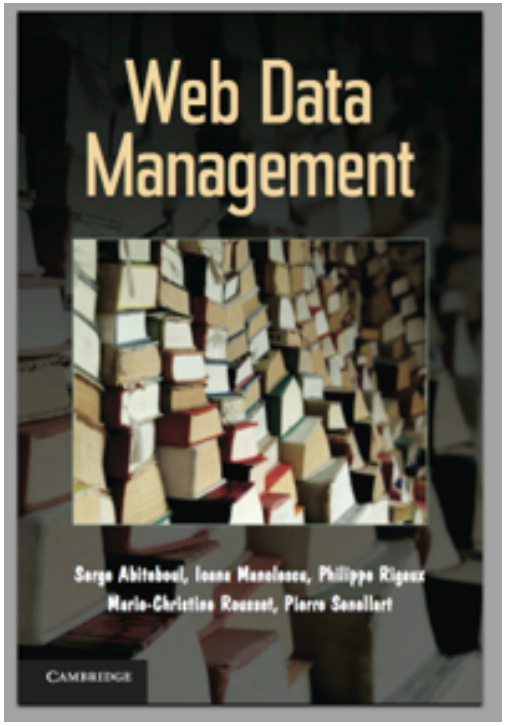
www.evolutionoftheweb.com - Partie 2 du Module



Références Bibliographiques pour la Partie 1

1. Livre (Open Access)
   Web Data Management
   Abiteboul & al.
   - Chapitre data-model link
   - Chapitre schemas (Section 3) link
   - Chapitre XPath link

2. Articles de recherche disponibles sur Moodle pour la partie XML↔Relationnel



Les transparents ne remplacent pas dans aucun cas ces références.

What is XML ?

- XML = eXtensible Markup Language [W3C ‘98]
- Ask five different people, get five different answers…
  - a self-describing data format ?
  - a generalization of HTML ?
  - best thing ever invented ?
  - part of the DNA of information technology ?
  - a meta-language ?
- Every answers carries some truth in it.
- But why would you need a meta-language, which self-describes data, and (also) generalizes HTML ?

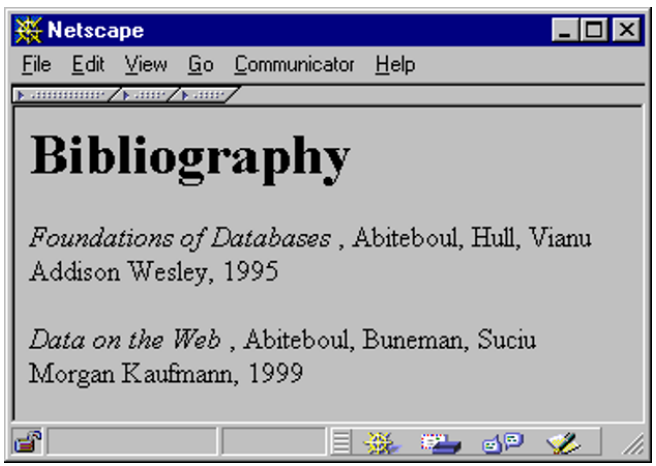
Why XML?

If XML is the solution... then what was the problem ?

- Web data :
  - By far the largest information system ever seen.
  - Billions of textual documents, images, PDF, multimedia files, provided and updated by millions of institutions and individuals.
  - An anarchical process which results in highly heterogeneous data organization, steadily growing and extending.
- The challenges :
  - Master the size and extreme variability of Web information, and make it usable.
  - Ensure long-term access to data. Write documents in 1998 and read them safely in 2087.

Before XML...

- Web data was based on HTML, which describes presentation.
- This is appropriate for humans, but falls short when it comes to software exploitation of data.




```
1  <html>
2
3   <h1>  Bibliography </h1>
4
5   <p> <i> Foundations  of  Databases</i>
6           Abiteboul,  Hull,  Vianu
7           <br>  Addison  Wesley,  1995
8
9   <p> <i> Data  on  the  Web </i>
10           Abiteboul,  Buneman,  Suciu
11          <br>  Morgan  Kaufmann,  1999
12 </html>
```

With XML...

- We have a language to describe content which is both :
- human-readable : verbosity helps the user which reads a document
- machine-readable : content is described by tags (markup language, self-describing data) which promotes machine-to-machine communication and data exchange

```
1  <bibliography>
2   <book>
3   <title>  Foundations of
          Databases      </title>
4   <author>  Abiteboul </author>
5   <author>  Hull      </author>
6   <author>  Vianu     </author>
7   <publisher>  Addison  Wesley
                    </publisher>
8   <year> 1995 </year>
9   </book>
10 </bibliography>
```

```
1  <html>
2
3   <h1>  Bibliography </h1>
4
5   <p> <i> Foundations  of  Databases</i>
6           Abiteboul,  Hull,  Vianu
7           <br>  Addison  Wesley,  1995
8
9   <p> <i> Data  on  the  Web </i>
10           Abiteboul,  Buneman,  Suciu
11          <br>  Morgan  Kaufmann,  1999
12 </html>
```

Did we really need a new language ? (history moment)

- Actually, the importance of markup languages for displaying information within applications had already been recognized - even before the advent of the Web.
- There was already a language called SGML
  - Standard Generalized Markup Language, ISO 8879:1986
- But this language is very (too much) general (see next slide).
- XML = a concise profile (subset) of SGML for the World Wide Web.

Did we really need a new language ? (history moment)

- SGML is very general. To see that, consider that it allows one to generalize (= you can write a SGML document for) :

- Wiki-like syntaxes

  ```
  ''italic'', '''bold''', === Header 2 ===, [[anchor]]
  ```

- RTF (Rich Text Format, Microsoft, 1987-2008).
  Document interchange format for Microsoft products.

  ```
  {\pard This is some \b bold text. \par }
  ```

- HTML

  ```
  <par> This is some <b> bold </b> text <\par>
  ```

- DTD (Document Type Definition) : even schemas for data !

How is XML used?

- XML is a descendant of SGML, and therefore the design of XML has been focussing on documents = information destined for publication/visualization (primarily on the Web).


```
<tweet>
  <ref_user>example</ref_user>          <text> absolutely smashed it at   </text>
  <hashtag>mtvliveclockdown</hashtag> <text>! Catch him at the official </text>
  <ref_user>clubmtivik</ref_user>       <text> after party tonight @ 10pm </text>
</tweet>
                                                (: can you write this in JSON ? :)
```

- This means that XML has all features needed to write documents
  - elements, attributes, entities, mixed content, ordering, etc…

- But this did not limit the range of uses of XML.

How is XML used?

- For example, you can express data = information destined to be processed by a database or a (web) application.

```
1 <reservation>                    (: this document is like a relational tuple :)
2   <passenger>Alice</passenger>
3   <flight_number>AB678</flight_number>
4   <price>1.345</price>
5 </reservation>
```

reservation	passenger	flight_number	price
	Alice	AB678	1.345

- no tag interleaving, no mixed content, order more or less important

How is XML used?

(http://www.snee.com/xml/xml2004paper.html)

- Data-Oriented XML : use XML as an interchange format to relay database or transaction information. Computer sends a message to another, often as part of a transaction, and this message typically carries information that began or ends up in a database (e.g., relational).

- Document-Oriented XML : uses XML to impose structure on information that may not fit into a database - particularly information intended for publishing (e.g., HTML)

- Technically, there is no difference. All XML is data in documents. However, the concepts are still important.

How is XML used ?

- Document-Oriented XML


```
<tweet>
  <ref_user>example</ref_user>          <text> absolutely smashed it at   </text>
  <hashtag>mtvliveclockdown</hashtag> <text>! Catch him at the official </text>
  <ref_user>clubmtivik</ref_user>       <text> after party tonight @ 10pm </text>
</tweet>
                                                (: can you write this in JSON ? :)
```

- Data-Oriented XML

```
<reservation>                        (: this document is like a relational tuple :)
  <passengers>Alice</passenger>
  <flight_number>AB678</flight_number>
  <price>1.345</price>
</reservation>
```

- First : there are many XML-based standards, that we use every day.
  - XML-based = they use XML as syntax
- Document-oriented :
  - Web apps : XHTML5 (the XML serialization of HTML5)
  - Word processing : ODF (OpenOffice.org), OOXML (Microsoft),...
  - Graphics : SVG
  - Content publishing : RSS, Atom, iWork (Apple),PDF/XFA (Adobe) ...
- Data-oriented :
  - Semantic Web : RDF/XML
  - Web apps : AJAX (XMLHttpRequest)
  - Services Oriented Architectures : SOAP, XMPP, XMLRPC
  - Other industry standards : UBL (Universal Business document Language), UPnP (Universal Plug and Play, home electronics & IoT), Health Level 7 (clinical and administrative data), OpenTravel Alliance (travel information), FpML (Financial products Markup Language), ...
  - Schema definition : XSD (schema language for XML with XML syntax)
- Standards are complex! Standardization docs are hundreds of pages

What is XML used for ? (1) XML-based Standards

- XML = describe things/objects (their data, metadata, references,...)
- Publishing
  - Web apps : XHTML5 (XML serialization HTML5)
  - Word processing : ODF (OpenOffice.org), OOXML (Microsoft),...
  - Content publishing : RSS, Atom, iWork (Apple),...
- Interacting/Presenting :
  - Graphics : SVG
- Serialization of other languages :
  - Semantic Web : RDF/XML
- Communication / Messaging :
  - Web apps : AJAX (XMLHttpRequest)
  - Services Oriented Architectures : SOAP, XMPP, XMLRPC
- Data exchange :
  - Other industry standards : UBL, UPnP, Health Level 7 ...
- Modelling :
  - Schema definition : XSD (schema language for XML with XML syntax)
- Although the design of XML focuses on documents, the language is widely used for the representation of arbitrary data structures such as those used in web services.

- Second : XML is used as a serialization format for large datasets
  - large = from MB to GB, anyways bigger than SOAP/HTML/SVG files

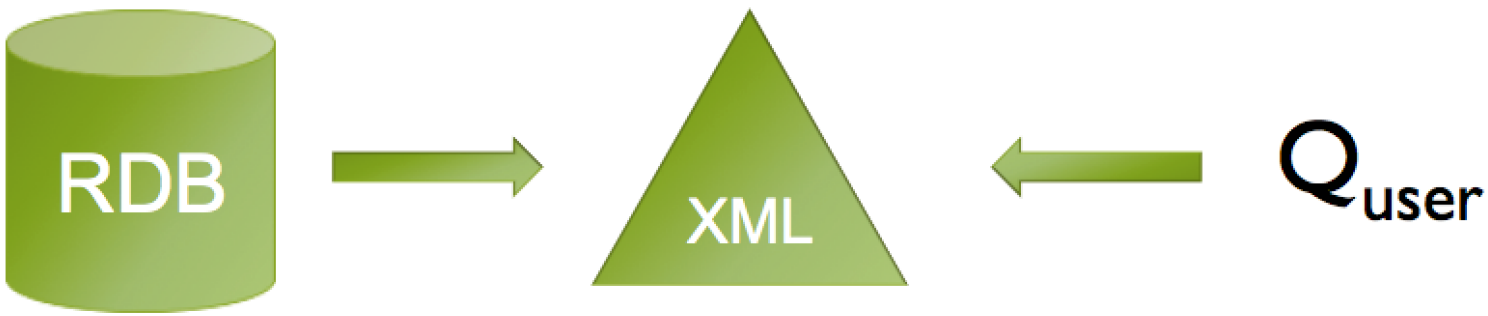
Use cases :

- A company can collect internal/external data coming from different sources and store it as XML.
  - data can stay within the company (undisclosed), or
  - the company may expose a Web interface allowing clients to query
- An institution can publish a Web open dataset. (e.g., Montpellier : https://data.montpellier3m.fr/datasets/commune/montpellier-mediterranee-metropole)
- The content of a database can be exported as XML and shared.
  - across platforms, DBs, enterprises

- Large data Publishing, Exchange, and Integration
  - Massive demand : across platforms, DBs, enterprises
  - Very important issues : DB vendors have products for all of these
- And because of the widespread of the language most commercial RDBMS provide today XML support
  - Oracle
  - IBM DB2
  - Microsoft SQL Server – XML support since 2005
- Even SQL had to evolve to embrace XML (standardization of 2003)
- XML support in Oracle and SQL/XML extensions will be covered later in this course.
- XML (and later JSON) have become the prime standards for data exchange and representation on the Web

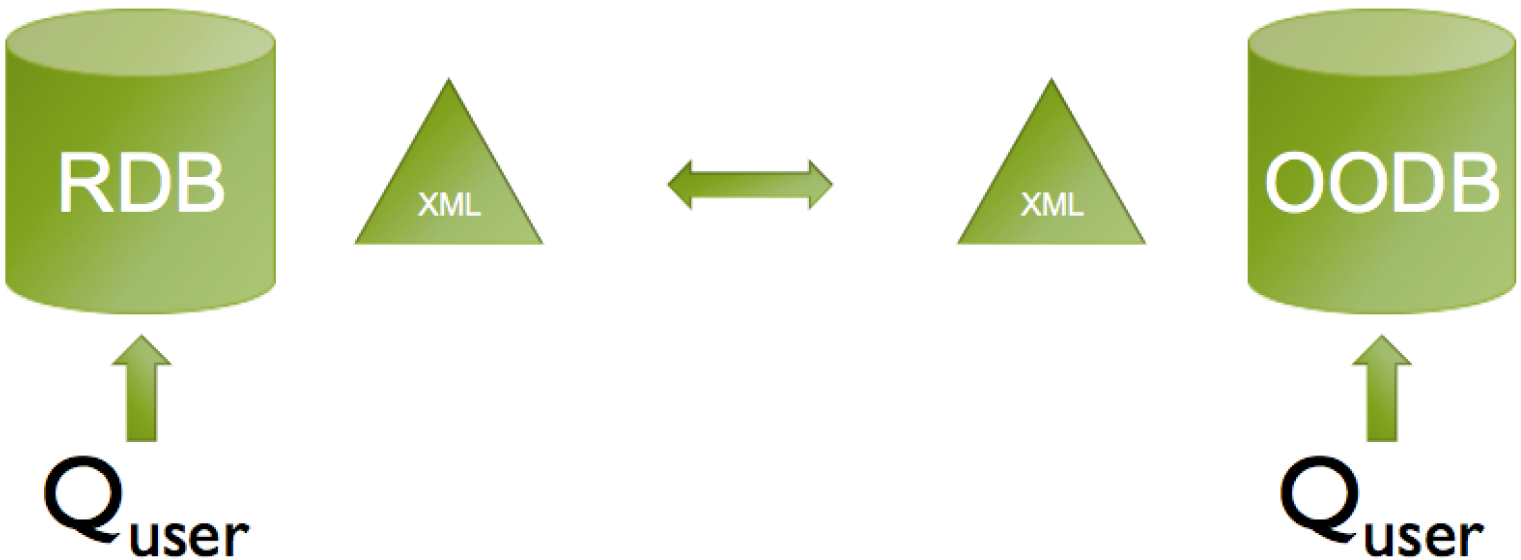
Data publishing

Make available in the Web a dataset which is not XML



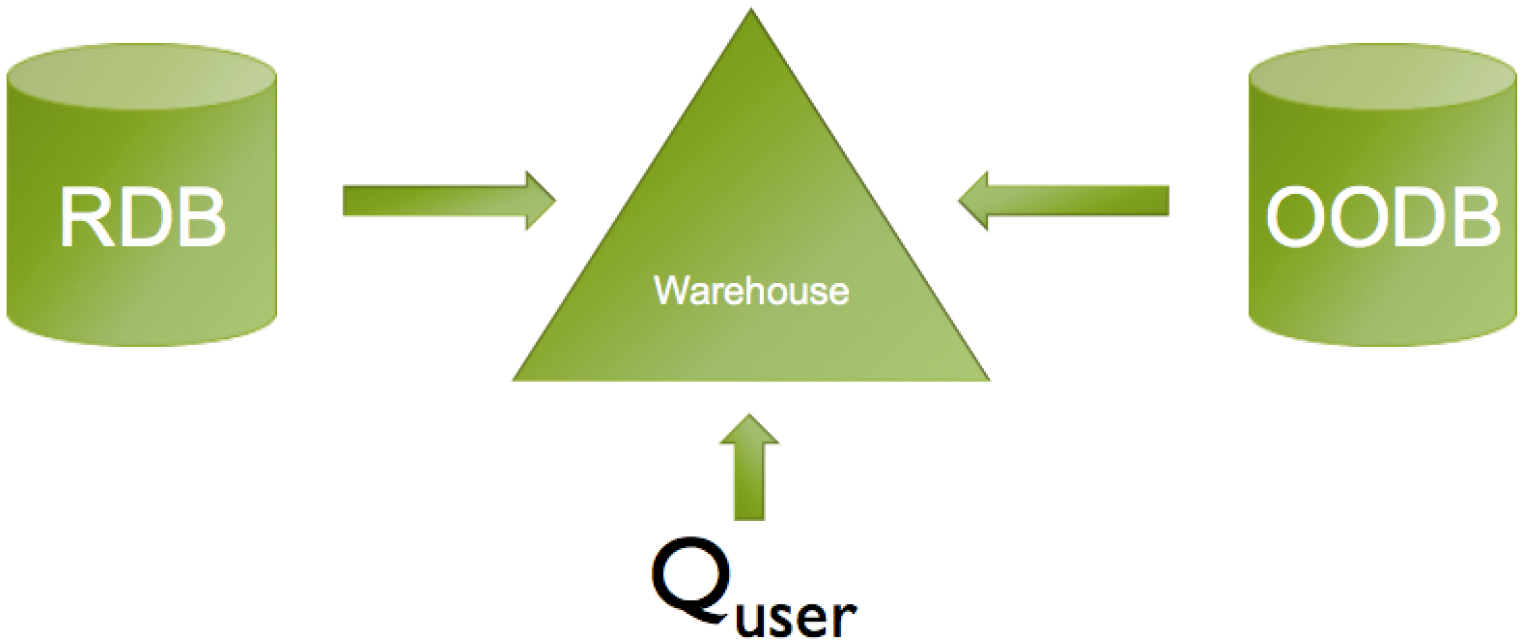
Data exchange

Transfer information between heterogeneous systems



Data integration

Aggregate information from heterogeneous systems



Is there any alternative to XML ?

- Since 2015, JSON (proposal of 2000, then standardized in 2013!) is becoming the dominant format for data-oriented usage. XML was dominant for the previous period 2000-2015.
  - Web apps:
    - » MEAN architecture combines Node.js/Angular.js and MongoDB (storing JSON). This will be covered in the second part of this course.
    - » XMLHttpReq allows one also to transfer JSON (chose AJAX or AJAJ).
  - NOSQL & intensive distributed data-processing :
    - » JSON has been adopted as the data-model of Key-Value stores NOSQL systems (MongoDB, CouchDB, etc) to represent records
    - » Easier to load data in these systems if we use the JSON syntax.
- But XML remains the dominant format for document-oriented usage (XHTML) and remains very important for data-oriented usage (AJAX, export of large datasets) as well as the basis of hundreds of XML-based standards (SVG, ODF, RSS, RDF/XML, SOAP, XSD, etc).
  - We also have many tools for translating (data-oriented) XML to JSON.

Is there any alternative to XML ?

- In regard of JSON, XML offers:
  1. more expressiveness : not all documents do naturally fit in JSON which has been conceived for serializing and exchanging Javascript objects
  2. standardized languages (W3C), with stable and predictable behaviour (DTD, XML Schema, XPath/XQuery, XSLT, etc..). Some warnings :
     - JSON does not have a standardized query language.

       Every NOSQL system provides its own language with arbitrary syntax and semantics. Querying a NOSQL system may not do what you expect !

       An emblematic case for MongoDB is the use of strict equality for nested records only `https://docs.mongodb.com/manual/tutorial/query-array-of-documents/`.

       (quoting the documentation) Equality matches on the whole embedded/nested document require an exact match of the specified document, including the field order.

     - JSON-Schema language (2014) not a standard in 2019 (even if it will).

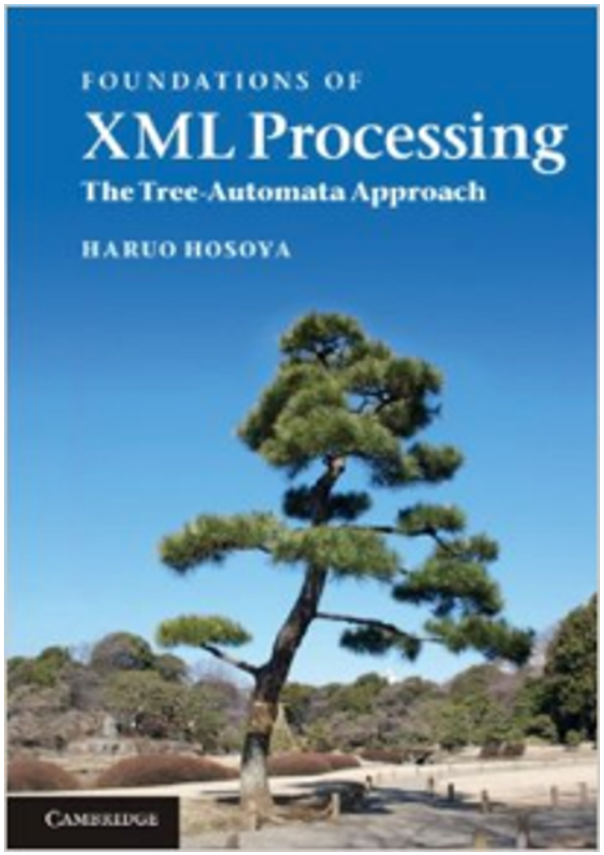
       It is supported by some systems before the standard reaches its final version (MongoDB 4.2 supports the Internet-Draft number 4 of JSON-Schema, expired in 2013). This is good (for users) and bad (for users and system maintenance, if major changes).

       `https://docs.mongodb.com/manual/reference/operator/query/jsonSchema/`

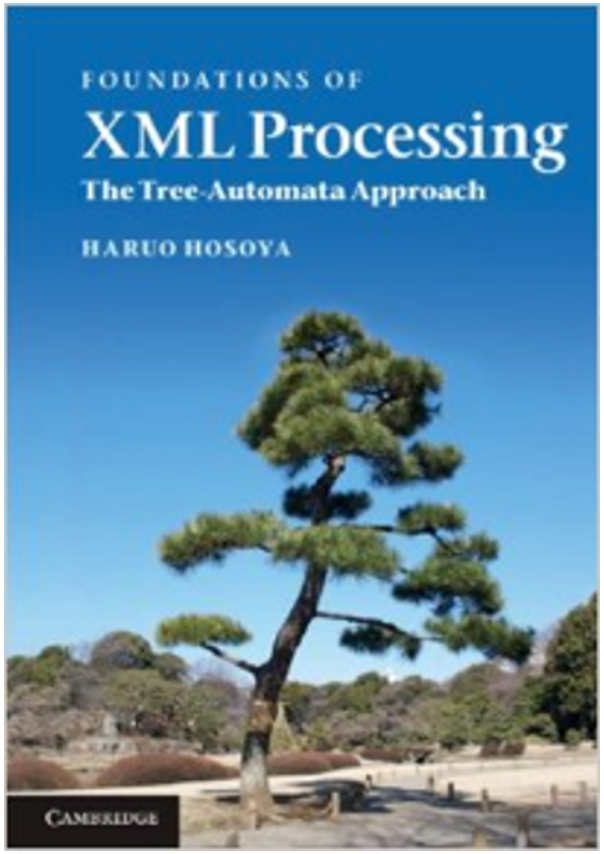
       So if one needs to define and share a schema and wants to validate the data one should go for XML.

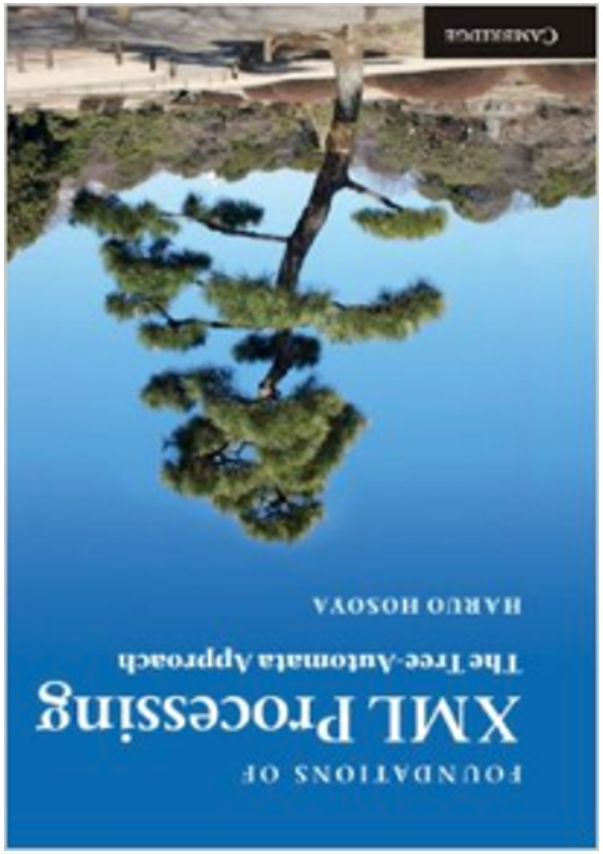
The essence of XML

The essence of XML : Trees



The essence of XML : Trees





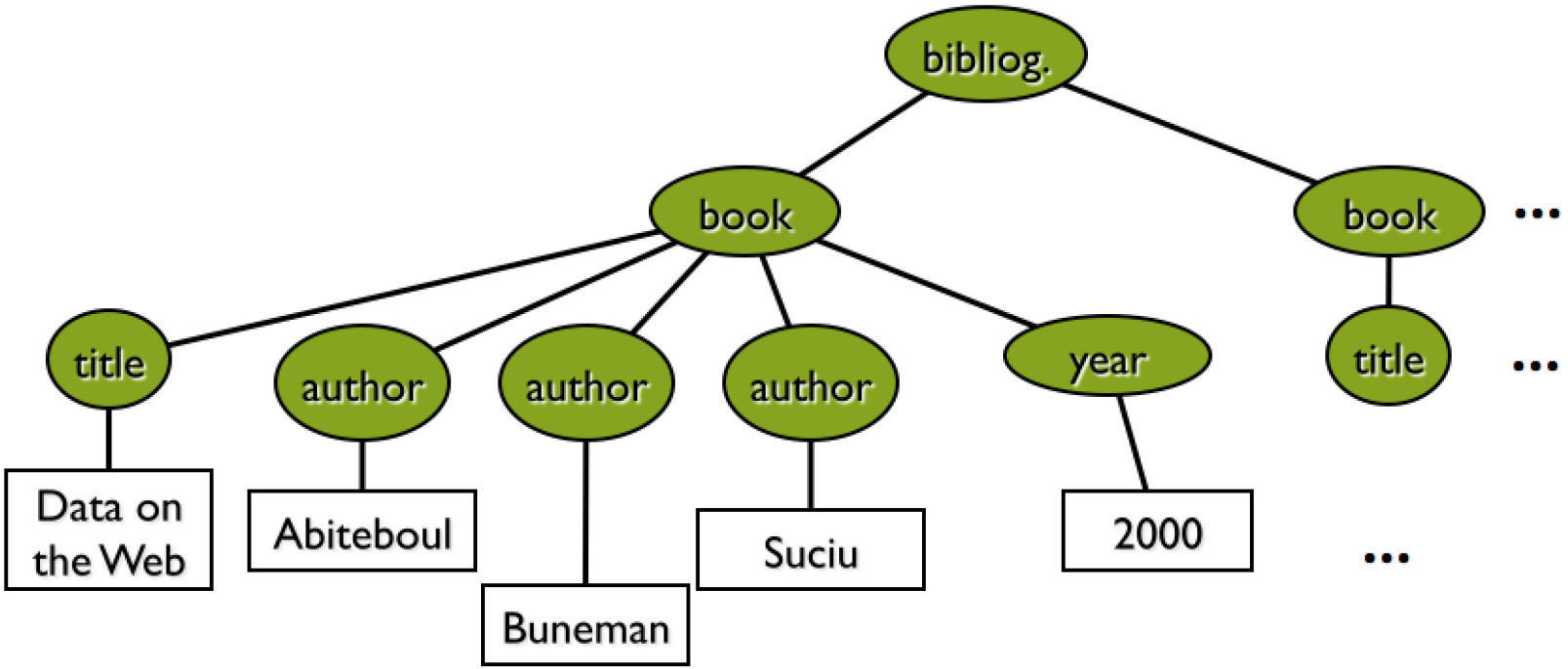
The verbose syntax may make XML looking unappealing.
But, understand that XML is essentially a syntax for trees.

Trees: the essence of XML

```
<bibliography>
   <book>
       <title>Data on the Web</title>
       <author>Abiteboul</author>
       <author>Buneman</author>
       <author>Suciu</author>
       <year>2000</year>
   </book>
   <book>
       <title>...</title>
       ...
   </book>
 ...
</bibliography>
```

Trees: the essence of XML

XML = ordered, node-labelled, unranked trees



Trees for representing data : an old idea (history moment)

- From a human perspective, trees are the most natural way for representing information...
  - Human knowledge tend to be hierarchically organized (think of a book divided in chapters, sections, paragraphs)

- ...but evidence shows that trees are not always the most efficient way of storing information for intensive data processing
  - see the multi-billion industry built around the Relational Model

- Many attempts to work with hierarchical databases
  - From the 60s and IMS (IBM)
    - then, Ted Codd's model won the "Relational War" in IBM ! (1970)
  - From the 80s and object databases
  - Both failed, for various reasons (eg., IMS lacking declarative features, OODB could not beat performances of competitors).

Trees for representing data : an old idea (history moment)

- So why should this idea work on the Web ?
- Because XML not coupled with a system for intensive data processing
- XML is a model for data-exchange and representation.
- It does not constraint data like a data-management systems would do (eg. strong typing), and gives the flexibility needed in the Web context.
  - Self describing data: no separation between schema-vocabulary (tags) and data
  - Meta-language : choose your own tags, and no need of a schema (but add one if needed)

Trees : the essence of XML

Plus ça change, plus c'est la même chose. Oubliez la syntaxe, pensez aux arbres!

Serge Abiteboul ($\sim$ 2010)

Trees : the essence of XML

Plus ça change, plus c'est la même chose. Oubliez la syntaxe, pensez aux arbres!

Serge Abiteboul ($\sim$ 2010)

(JSON docet)

Defining the structure of a tree : XML Node Types

A model for trees

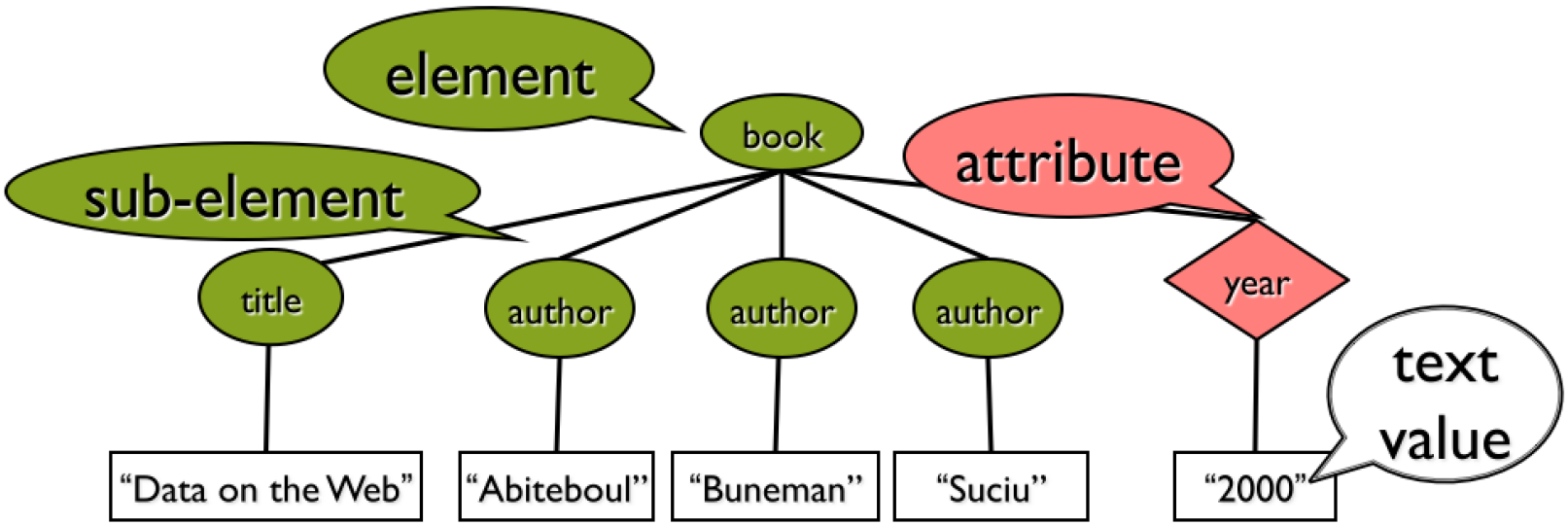
- XML nodes can be of many different types
  - Elements
  - Attributes
  - Text-nodes

  - … but also Processing instructions, Comments, CData Node, Document Node, Entity Node, Entity Reference Node, Notation Node

- We will focus on the node types that provide most of the structure of the XML document.

A closer look

```
<book year="2000">
  <title>Data on the Web</title>
  <author>Abiteboul</author>
  <author>Buneman</author>
  <author>Suciu</author>
</book>
```



Element (Node Type)

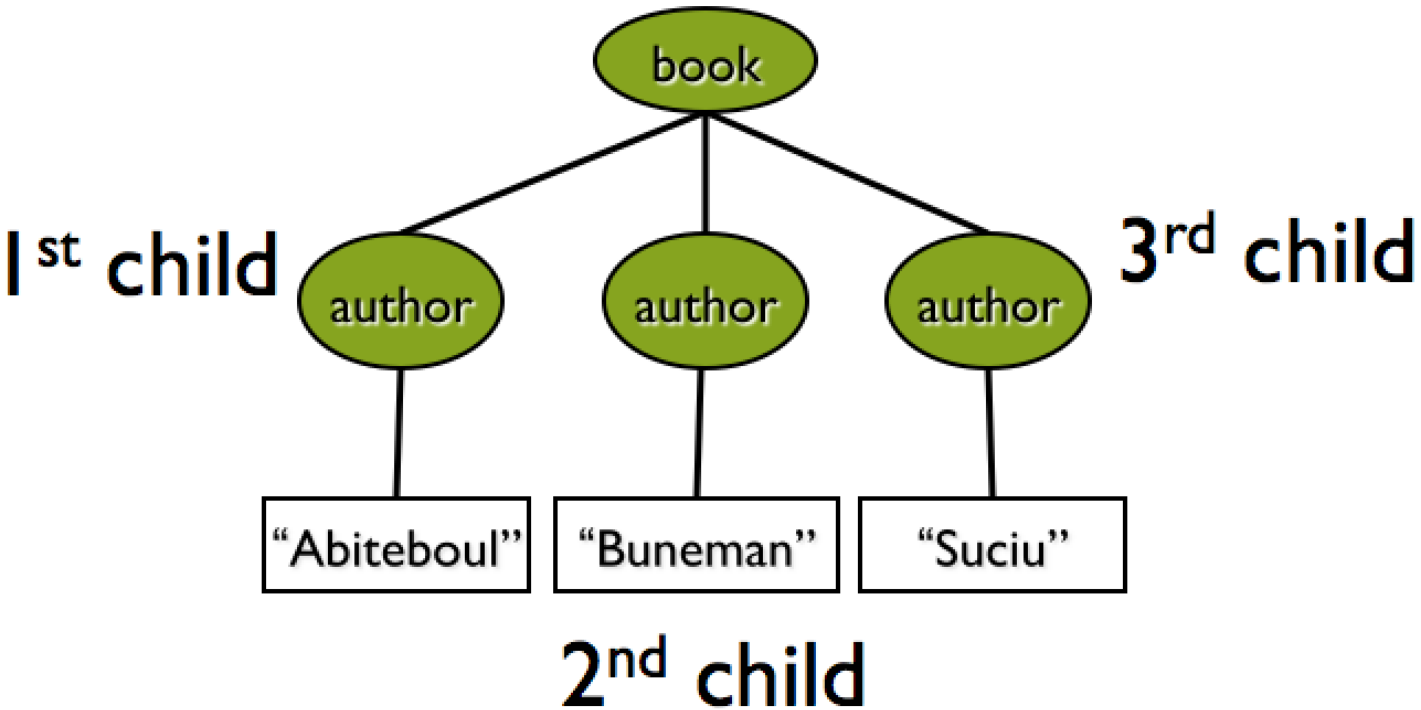
- Segment between a start and (corresponding) end tag
  - trees (not forests) $\Rightarrow$ unique root element
  - elements can be nested to express ‘records’ and ‘lists’

- Opening and closing tags must properly match.

```
1 <root>
2   <a> <b> </a> </b>          (: NO :)
3   <a> <b> </b> </a>          (: OK :)
4 </root>
```

Ordering

- XML elements are (totally) ordered



- How to represent sets in XML ?

Attribute : Syntax

- Start tag may contain attributes describing the element
- XML attributes cannot be nested (they are ‘flat’)

```
1 <picture>
2  <height dim='cm'> 2400</height>
3  <width dim='in'> 96 </width>
4  <data encoding='gif'> M05-+C\$ </data>
5 </picture>
```

Attribute : Structure

- The attributes of an element must be unique.

```
1 <person friend='Blair'  friend='Clinton'>
2   Obama
3 </person>                              (: NO :)
```

```
1 <person friend1='Blair'  friend2='Clinton'>
2   Obama
3 </person>                              (: OK :)
```

Attribute : Structure

- Order does not matter

```
<person id='p1' friend='p2'>
  <name>Bill Clinton </name>
</person>
```

```
<person friend='p2' id='p1'>
  <name>Bill Clinton </name>
</person>
```

Should I use elements or attributes ?

```
<course date="12/11/2022">
 <teacher>FU</teacher>
 <degree>M1</degree>
 <title>Web Data Management
 </title>
</course>
```

```
<course>
 <date>12/11/2022</date>
 <teacher>FU</teacher>
 <degree>M1</degree>
 <title>Web Data Management
 </title>
</course>
```

```
<course>

 <date>
  <day>12</day>
  <month>11</month>
  <year>2002</year>
 </date>

 <teacher>FU
  </teacher>
 <degree>M1</degree>
 <title>Web Data
     Management
 </title>
</course>
```

- It depends on the needs of the applications that will use your XML !

This is not how XML should be used

```
<course
  day='12'
  month='11'
  year='2002'
  teacher='FU'
  class='M1'
  title='Web Data Management'
</course>
```

- Why ? It is principally matter of style. This document is not illegal, and equivalent to the last one, but it departs from the spirit of XML.
- Main problem : everything (too much information) inside attributes.
- We have lost the order (sometimes an issue, sometimes not)
  - if you select all the attributes the querying system may chose to display them in a strange order : day then teacher then year
  - if it is not an issue you can use JSON
- There may be ambiguities when the XML is extended. Suppose later we add another date for the academic-year, the all attributes will be mixed together. Also, suppose that we add the time. The date needs some structure and (day,month,year) attributes grouped together.

Should I use elements or attributes ?

- Be aware of the following things :
  - attributes cannot contain multiple values (elements can)
  - attributes are not are not easily expandable (for future changes)
  - attributes cannot describe structures (elements can)
  - attributes are more difficult to manipulate by program code
- … but this does not mean that attributes must be avoided !

Use attributes for IDs and Keys !

```
<person id_pers = "barak"   friend="bill">
  <name> Barak Obama </name>
  </person>

<person id_pers = "bill"   friend="barak">
  <name> Bill Clinton </name>
</person>
```

- The integrity of ID/IDREF will not be verified ...
- ...unless a DTD is available (we will come back next to this)

Quiz : find the errors

```
1  <books>
2   <book id='b1>
3      <title>Data on the Web</title>
4    <year>2000/year>
5    <authors>
6        <author id="b1">Abiteboul
7        <author id=a2>Buneman
8         </author>
9     <publisher>"Addison-Wesley"</publisher>
10 </books>  <foo>bar</foo>
```

Other kinds of nodes

- entity references: & " >
  - textual substitution; allows escaping special characters
  - you can define your own if you want
- processing instructions: <? foo : bar ?>
  - can be used to pass information to processors
- comments: <!– foo –>
- CDATA sections: - <!CDATA[[I <3 XML]]>
  - allows including raw text (<, >, &, etc. uninterpreted)
- Luckily, these are mostly irrelevant to use of XML for data
  - but you need them when writing reading/writing XML as text

Summing Up

- XML, the standard de-facto for data representation and exchange on the Web.
- Trees are the essence of XML, and there exists a precise syntax to define them.