XML
(eXtensible Markup Language)

CMT220
Databases & Modelling

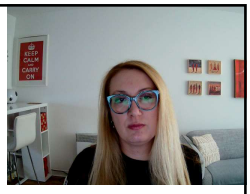
Cardiff School of **Computer Science & Informatics**
http://www.cs.cf.ac.uk

CARDIFF UNIVERSITY
PRIFYSGOL CAERDYDD

1

1

Lecture



- **content**
  - markup languages
  - XML & its basic concepts
  - structuring data with XML
- **learning outcomes**
  - describe the XML data model & outline its basic features
  - understand the advantages of the XML approach to data management

CARDIFF UNIVERSITY
PRIFYSGOL CAERDYDD

2

2

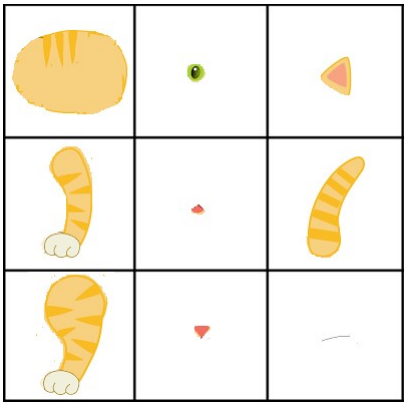
Relational data model

- simple
- relations are the key concept
- primitive data types, e.g. string, integer, date, etc.
- mathematical theory
- great performance: query optimization
- good for administrative and transactional data
- not as good for other types of complex data

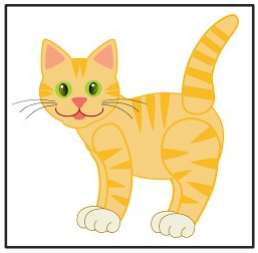
CARDIFF UNIVERSITY PRIFYSGOL CAERDYDD

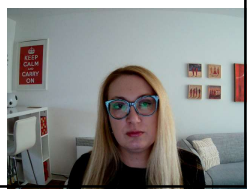
3

Relational vs. object–oriented model



vs.





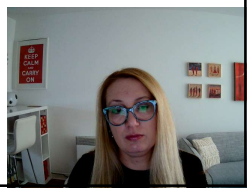
CARDIFF UNIVERSITY PRIFYSGOL CAERDYDD

- at query time, things are put back together

4

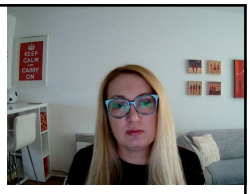
XML: design goals

- represent semi–structured data (data that are structured, but do not fit relational model)
- offer more flexibility than databases, but still do some of the database functionality
- separate syntax from semantics to provide a common framework for structuring information
- allow tailor–made markup for any imaginable application domain
- support internationalisation (Unicode) and platform independence



5

What is XML?

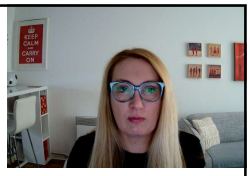


- **XML** = e**X**tensible **M**arkup **L**anguage
- first published in 1997
- a World Wide Web Consortium (W3C) standard
- W3C: the main international standards organisation for the World Wide Web
- XML = "SGML for the Web"
- SGML (Standard Generalized Markup Language): an ISO-standard technology for defining generalised markup languages for text documents

6

6

Markup language



- **markup language**: a system for annotating text in a way that is syntactically distinguishable from the text itself
- three types of electronic markup:

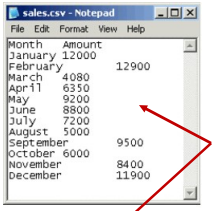
1. **presentational**: achieve a visual effect, e.g. in **HTML**
   red and bold
   red and bold
2. **procedural**: how to process the text, e.g. in **LaTeX**
   \sum_{i=1}^{\infty}\frac{1}{i} $\sum_{i=1}^{\infty} \frac{1}{i}$
3. **descriptive**: provide additional information, e.g. in **XML**
   <NP><DT>a</DT> <JJ>new</JJ> <NN>example</NN></NP>

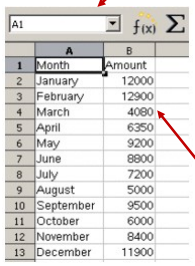
CARDIFF UNIVERSITY
PRIFYSGOL CAERDYDD

7

7

XML tags





- **XML** is a markup language that is used to store data in a self-descriptive manner
- making the data "self-descriptive" is achieved by tagging (annotating or marking up) information
- unlike delimited files or database tables, XML documents are structured by tags
- **tags** look like this: **<TAG>**some data here**</TAG>**

open close

- tags indicate the beginning & ending of the tagged data – text–based & position–independent
- e.g. <sale><month>March</month> <amount>4080</amount> </sale>

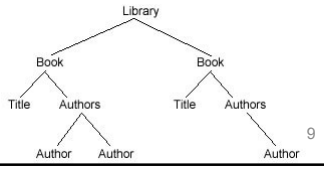
8

XML tags

- the basic structure of XML files:
  - there are tags: ... <TAG>tagged data</TAG> ...
  - tags surround data, or other tags:
    <MOTHER>... <CHILD>nested tag</CHILD> ...</MOTHER >
- **NOTE**: tags can only be **nested** within other tags,
  i.e. they **cannot** overlap partially!

<ONE>...<TWO> BANNED </ONE> ...</TWO>

- therefore, XML documents have **hierarchical** or **tree-like** structure

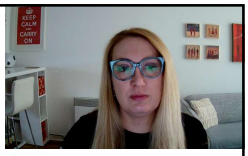


CARDIFF UNIVERSITY
PRIFYSGOL CAERDYDD

9

9

XML tags – example



```
<xml>
  <sender>
    <name>
      <first>John</first>
      <middle>C</middle>
      <last>Doe</last>
    </name>
    <address>
      <line1>87 Victoria Road</line1>
      <line2></line2>
      <line3></line3>
      <town>Wisbech</town>
      <county>Cambridgeshire</county>
      <postcode>PE13 2QL</postcode>
    </address>
  </sender>
</xml>
```

- xml
  - sender
    - name
      - first: John
      - middle: C
      - last: Doe
    - address
      - line1: 87 Victoria Road
      - line2:
      - line3:
      - town: Wisbech
      - county: Cambridgeshire
      - postcode: PE13 2QL

the same!

```
<xml><sender><name><first>John</first>
<middle>C</middle><last>Doe</last>
</name><address><line1>87 Victoria
Road</line1> <line2></line2> <line3>
</line3><town>Wisbech</town><county>
Cambridgeshire</county> <postcode>PE13
2QL</postcode> </address></sender></xml>
```

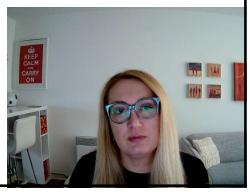
CARDIFF UNIVERSITY
PRIFYSGOL CAERDYDD

10

XML element

- an **XML element**, e.g. <TAG>some data here</TAG>, consists of:

1. the **delimiters**: "<" and ">" (special characters in XML)
2. the **identifier/name**: the "TAG" enclosed by the two delimiters
3. the opening & closing **tags**: "<TAG>" & "</TAG>“ (note the backslash in the closing tag)
4. the **content**: "some data here"

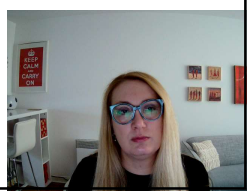


11

XML attribute

- **XML attribute**: specifies **additional information** about an XML element
- an attribute for an element appears within the opening tag: <element **attributeName="value"**>...</element>
- attributes are means of specialising generic elements
- e.g. <temperature **unit="C"**>28</temperature>
- attribute vs. element:

```
<temperature>
    <value>28</value><unit>C</unit>
</temperature>
```

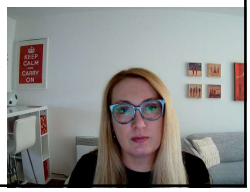


12

XML special characters

- some characters have a special meaning in XML
- e.g. "<" is always interpreted as the start of a new element
- this will generate an XML error:
  <message>if salary < 1000 then</message>
- replace a special character with an entity reference:
  <message>if salary < 1000 then</message>
- 5 predefined entity references in XML:

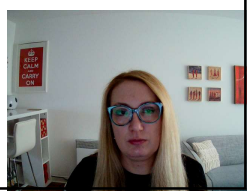
1.	<	<	less than
2.	>	>	greater than
3.	&	&	ampersand
4.	'	'	apostrophe
5.	"	"	quotation mark



13

XML structure – summary

- an XML document is an **ordered, labelled tree**
- each node, i.e. XML element:
  - must have a **name**
  - may have **attributes**, each consisting of a name & a value
  - may have **content**, which may include **child nodes**
- the XML code must be **syntactically correct** or the XML parser will report an error



14

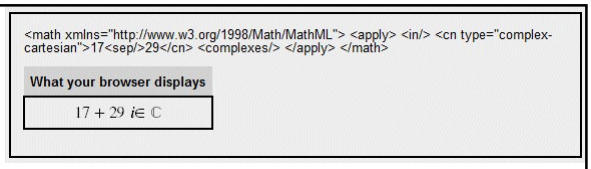
Well–formed XML documents

- an XML document is a text which is well–formed if it conforms to the XML syntax rules:

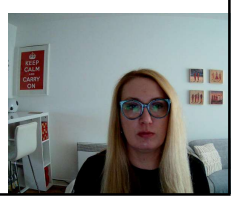
1. it contains only properly encoded legal **Unicode characters**
2. none of the **special syntax characters** (<, >, ", ', &) appear except when performing their markup roles
3. XML elements are **correctly nested**, with none missing & none overlapping
4. the XML tags are **case–sensitive**
5. there is a **single "root" element**, which contains all other elements

15

What is XML?

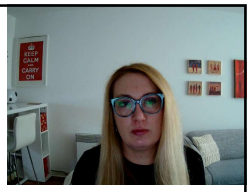


- not a language but a **meta-language**, i.e. a framework for defining markup languages (or dialects)
- **no fixed** collection of markup tags → XML is **flexible**
  - each XML language tuned for a specific application, e.g. MathML is an application of XML for describing mathematical notations
- all XML languages **share common features** → enables building of **generic tools** for processing XML data
  - all XML languages can be processed by a single lightweight parser
- XML is intended for machine processing, but it is still a **human readable format** mostly because the data are structured in tags that use common language, e.g. <phoneNumber>



16

XML vs. HTML



- **XML:** defines **logical structure** only
- **HTML:** the same intention, but has evolved into a **presentation** language; a markup language for a specific purpose – **display in browsers**
- unlike HTML, XML by itself conveys only content & structure, **not** presentation or behaviour
- these can still be associated with XML, but this requires additional mechanisms such as stylesheets, scripts, namespaces, etc.
- **XHTML (eXtensible HyperText Markup Language):** a family of XML markup languages that mirror or extend versions of the widely used HTML

CARDIFF UNIVERSITY PRIFYSGOL CAERDYDD

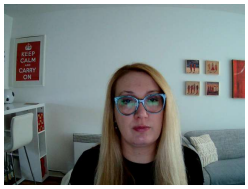
17

17

HTML vs. XHTML

- XHTML is a stricter & cleaner version of HTML
- XHTML is HTML re–designed as an XML language
- Why XHTML?
- many web pages on the Internet contain "bad" HTML:

```
<html>
<head>
<title>This is bad HTML</title>
<body>
<h1>Bad HTML
<p>This is a paragraph
</body>
```



CARDIFF UNIVERSITY PRIFYSGOL CAERDYDD

- XML is a markup language where documents must be marked up correctly & well–formed

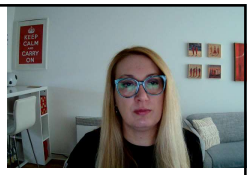
18

18

XML Schema

19

XML schema

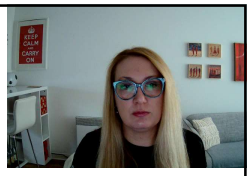


- How are **different XML languages** or dialects specified?
- **XML schema** = syntax definition (i.e. grammar) of an XML language – describes the structure of an XML document
- formal languages for expressing XML schemas:
  - **Document Type Definition (DTD)**
  - **XML Schema**
- they use very **different syntax** to achieve the same task of creating **documentation**:
  - what elements an XML document can contain
  - how they should be used
  - what interactions may take place between parts of a document

20

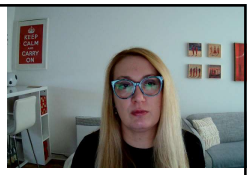
20

XML schema



- **NOTE:** neither DTD nor XML Schema are strictly required for XML development, but they are strongly recommended
- a well–formed XML document is **valid** if it conforms to the associated schema specified in DTD or XML Schema
- the main reason schemas are used in XML is to allow automatic validation of document structure
- creating an XML schema is known as data modelling
- an XML developer (or an application) can then use the schema to check if an XML document adheres to the model
- if not, then the document could prove to be problematic

DTD vs. XML Schema



```
<?xml version="1.0"?>
<!DOCTYPE bookstore [
 <!ELEMENT bookstore (name,topic+)>
 <!ELEMENT topic (name,book*)>
 <!ELEMENT name (#PCDATA)>
 <!ELEMENT book (title,author)>
 <!ELEMENT title (#CDATA)>
 <!ELEMENT author (#CDATA)>
 <!ELEMENT isbn (#PCDATA)>
 <!ATTLIST book isbn CDATA "0">
 ]>
```

DTD "grammar"

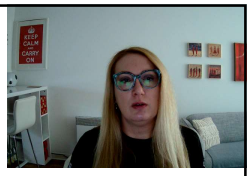
```
<xsd:complexType name="bookType">
 <xsd:element name="title" type="xsd:string"/>
 <xsd:element name="author" type="xsd:string"/>
 <xsd:attribute name="isbn" type="isbnType"/>
</xsd:complexType>
<xsd:simpleType name="isbnType">
 <xsd:restriction base="xsd:string">
   <xsd:pattern value="\[0-9]{3}[-][0-9]{3}[-][0-9]{3}"/>
 </xsd:restriction>
</xsd:simpleType>
```

XML Schema "grammar"

```
<bookstore>
 <name>Mike's Store</name>
 <topic>
   <name>XML</name>
   <book isbn="123-456-789">
     <title>Mike's Guide To DTD's and XML Schemas</title>
     <author>Mike Jervis</author>
   </book>
 </topic>
</bookstore>
```

XML "sentence"

XML data exchange



- XML standardises the concrete **syntax** of **data exchange** in a **text–based** notation designed to be obvious to both people & machines
- XML uses **documents** as the transfer mechanism for data
- XML publishing model **decouples data from processing**, which isolates changes in large systems, making them more flexible & reliable
- XML is suitable for **transactional processing** in a heterogeneous, asynchronous, distributed environment such as the Web

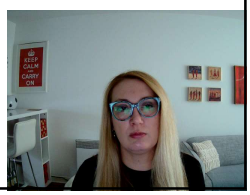
CARDIFF UNIVERSITY PRIFYSGOL CAERDYDD

23

23

XML advantages

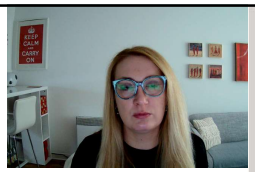
- data representation is text–based & position–independent
- open & extensible
- platform & language independent → portable
- human & machine readable
- self–descriptive data
- no dependence on large software vendors
- no binding to specific tools



CARDIFF UNIVERSITY PRIFYSGOL CAERDYDD

24

XML trade offs



- XML is **not** a slim format: using tags makes data bigger & more complex than a flat file
- **performance**: relational databases are still much faster
- **no centralised control** of data: potential problems with data integrity
- **uniformity**: too many different formats

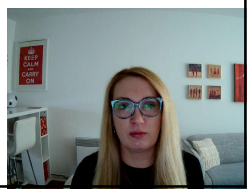
```
<empinfo>
  <employees>
    <employee>
      <name>Scott Philip</name>
      <salary>£44k</salary>
      <age>27</age>
    </employee>
    <employee>
      <name>Tim Henn</name>
      <salary>£40k</salary>
      <age>27</age>
    </employee>
    <employee>
      <name>Long yong</name>
      <salary>£40k</salary>
      <age>28</age>
    </employee>
  </employees>
</empinfo>
```


```
{ "empinfo" :
  {
      "employees" : [
      {
        "name" : "Scott Philip",
        "salary" : £44k,
        "age" : 27,

      },

      {
        "name" : "Tim Henn",
        "salary" : £40k,
        "age" : 27,
      },

      {
       "name" : "Long Yong",
       "salary" : £40k,
       "age" : 28,

      }
                ]
  }
}
```

25

Summary

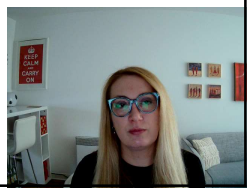
- XML is a W3C standard **meta-language** for defining markup languages
- **markup** language: a system for annotating text
- XML is used to store data in a **self-descriptive** manner using **XML tags**
- XML documents are structured using **XML elements**, which must have a **name**, may have **attributes**, and may have **content**, which may include other XML elements
- an XML document is an **ordered, labelled tree** of XML elements



26

Summary

- XML uses documents as a **data exchange** mechanism
- an XML document is **well-formed** if it is syntactically correct according to the W3C specification
- different markup languages are specified using **XML schemas**
- an XML schema can be expressed in a formal language such as **DTD** or **XML Schema**
- a well-formed XML document is **valid** if it conforms to the associated XML schema



27