IRS- Part 2
XML Retrieval

What is XML?

- eXtensible Markup Language
- A framework for defining markup languages
- No fixed collection of markup tags
- Each XML language targeted for application
- All XML languages share features
- Enables building of generic tools

- XML - the eXtensible Markup Language has recently emerged as a new standard for data representation and exchange on the Internet.
- An XML document is an ordered, labeled tree. Each node of the tree is an element
- *XML element and is written with an opening and closing tag. An element can* have one or more *XML attributes.*

XML

- The standard for accessing and processing XML documents is the XML Document Object Model or *DOM.*
- *The DOM represents elements, attributes* and text within elements as nodes in a tree.
- Example:

XML Document

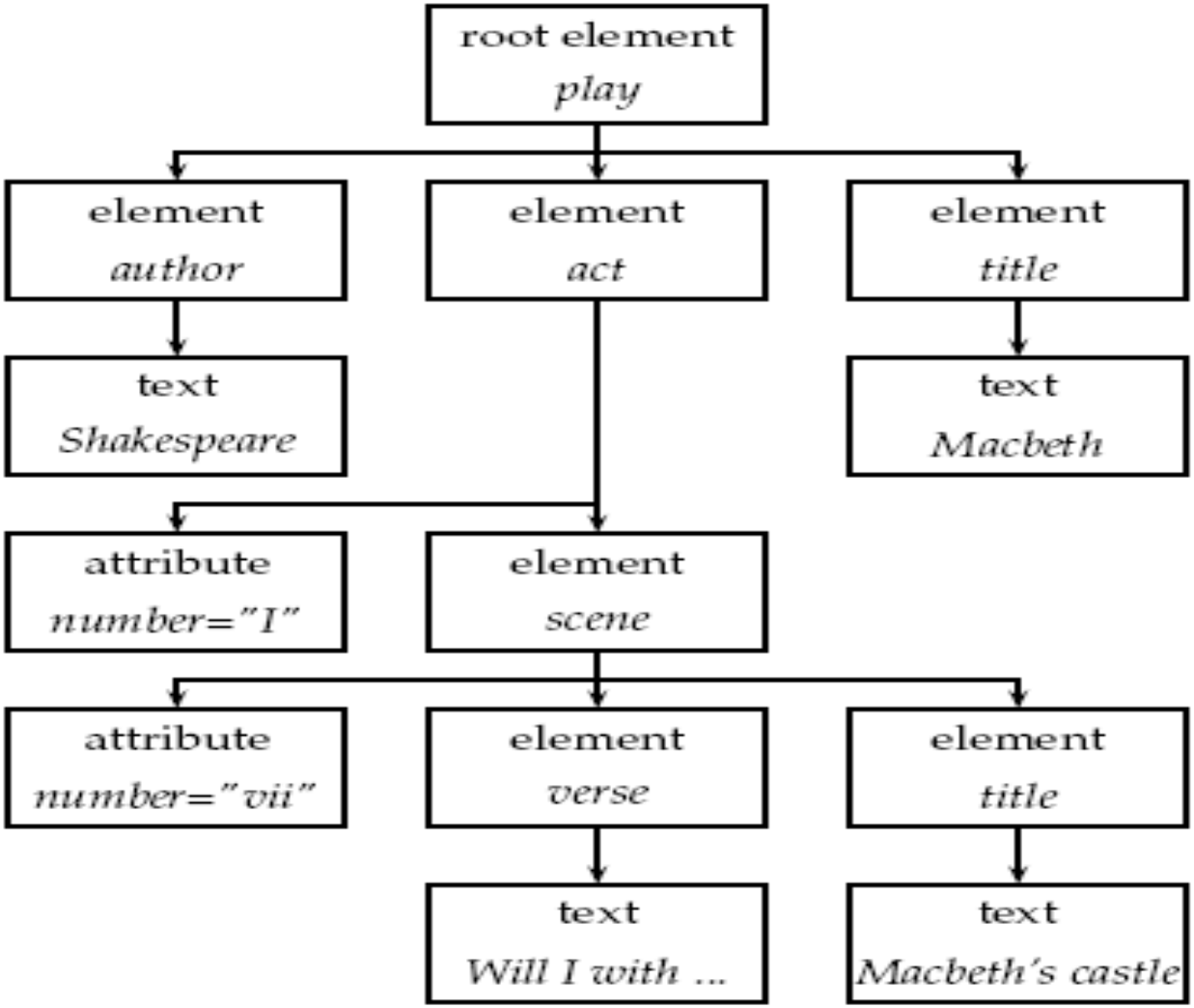
- <play>
- <author>Shakespeare</author>
- <title>Macbeth</title>
- <act number="I">
- <scene number="vii">
- <title>Macbeth’s castle</title>
- <verse>Will I with wine and wassail ...</verse>
- </scene>
- </act>
- </play>

XML Document

- Ordered, labeled tree
- Each node of the tree is an XML element, written with an opening and closing XML tag (e.g. <title...>, </title...>)
- An element can have one or more XML attributes (e.g. number)
- Attributes can have values (e.g. vii)
- Attributes can have child elements (e.g. title, verse)

```
<play>
<author>Shakespeare</author
<title>Macbeth</title>
<act number="I">
<scene number="vii">
<title>Macbeth's
castle</title>
<verse>Will I with wine
...</verse>
</scene>
</act>
</play>
```

XML document as DOM object



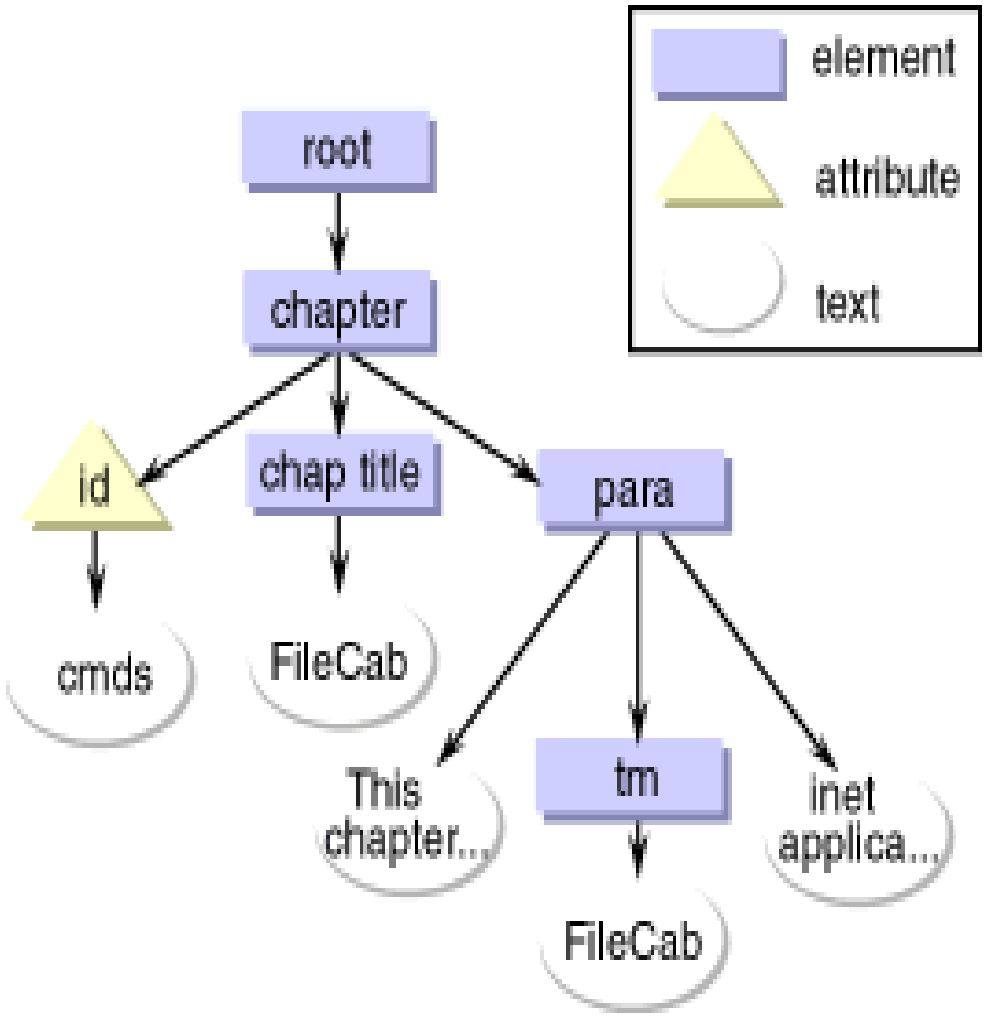
Basic Structure

- An XML document is an **ordered, labeled tree**
- **character data** leaf nodes contain the actual data (text strings)
- **element nodes**, are each labeled with
  - a name (often called the element *type*), and
  - a set of **attributes**, each consisting of a name and a value,
  - can have child nodes

XML properties

- hierarchical structure: nesting of elements
- element: start-tag – content – end tag
  `<tag-name> content </tag-name>`
- tag-name: logical name of element
- content: data or other elements
  (nesting of elements)
  `<author><first>John</first>`
  `<last>Smith</last></author>`
- attributes: assigned to elements
  (specified in start tag)
  pair of (*attribute name*, *attribute value*),
  e.g. `<date format="ISO">2000-05-01</date>`

XML Example




```
<chapter id="cmds">
        <chaptitle>FileCab</chaptitle>
        <para>This chapter describes
        the commands that manage
        the
        <tm>FileCab</tm>inet
        application.
        </para>
</chapter>
```

Design tree structure?

```
<book class="H.3.3">
  <author>John Smith</author>
  <title>XML Retrieval</title>
  <chapter>
    <heading>Introduction</heading>
    This text explains all about XML and IR.
  </chapter>
  <chapter>
    <heading> XML Query Language XQL </heading>
    <section>
        <heading>Examples</heading>
    </section>
    <section>
        <heading>Syntax</heading>
        Now we describe the XQL syntax.
    </section>
  </chapter>
</book>
```

XML applications

- XML application refers to an application of XML to a specific domain?

XML applications cont..

- CML- Chemists can create and publish molecule specifications for easy interchange.
- MML- To publish equations and all kind of mathematical terms
- CDF- introduce the idea of webcasting
- -send documents to the user rather than waiting for the user to come and get them

XML applications cont..

- SMIL- has become a core part part of the RealNetworks streaming software(Apple quicktime)

SMIL lets you create fast cuts and true multimedia presentations.

SVG- Scalable Vector Graphics-implementation in Adobe products, Corel draw)—One can draw 2 dimensional graphics using markup

XML applications cont..

- XUL- XML based user Interface Language—Enable you to describe what user interface elements you want those browsers to display.
- XBRL- Extensible Business Reporting language-To describe financial statements.
- RDF- resource description framework-XML application that specializes in meta-data. One use RDF to specify info. about other resources.
- One can create vocabulary that describe resources

Uses of XML

- XML has a variety of uses, including:
- **Web publishing**: XML allows you to create interactive pages, allows the customer to customize those pages, and makes creating e-commerce applications more intuitive. With XML, you store the data once and then render that content for different viewers or devices based on style sheet processing using an XSL/XSLT processor.

Uses of XML

- **Web searching and automating Web tasks**: XML defines the type of information contained in a document, making it easier to return useful results when searching the Web:
  - For example, using HTML to search for books authored by Tom Wolf is likely to return instances of the term 'wolf' outside of the context of author. Using XML restricts the search to the correct context (say, the information contained in the <author> tag) and returns only the required type of information. Using XML, Web agents and robots (programs that automate Web searches or other tasks) will be more efficient and produce more useful results.

Uses of XML

- **General applications**: XML provides a standard method to access information, making it easier for applications and devices of all kinds to use, store, transmit, and display data.
- **e-business applications**: XML implementations make electronic data interchange (EDI) more accessible for information interchange, business-to-business transactions, and business-to-consumer transactions.

Uses of XML

- **Metadata applications**: XML makes is easier to express metadata (Unified Modeling Language design models or user interface properties, for example) in a portable, reusable format.
- **Pervasive computing**: XML provides portable and structured information types for display on pervasive (wireless) computing devices such as PDAs, cellular phones, and others.
  - For example, WML (Wireless Markup Language) and VoiceXML are currently evolving standards for describing visual and speech-driven wireless device interfaces.

Uses of XML

- **Create other Languges**
  Many languages are created using XML. Examples are WML, MathML etc.

- **Use in Databases**
  Modern database such as Oracle and MSSQL server is providing the XML support in their databases.

XML-related technologies.

- **DTD (Document Type Definition)** is used to define the legal elements in an XML document.
- well-formed XML: proper nesting of elements
- (e.g. <a> is forbidden
- valid XML: document is well-formed and conforms to document type definition

Declaration of DTD

- in the document header:
- <!DOCTYPE name PUBLIC publicid systemid>
- <!DOCTYPE name SYSTEM filename>

Example XML document with system DTD

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE book SYSTEM  "/services/dtds/book.dtd">
<book class="H.3.3">
  <author>John Smith</author>
  <title>XML Retrieval</title>
  <chapter>
    <heading>Introduction</heading>
    This text explains all about XML and IR.
  </chapter>
  <chapter>
    <heading>
      XML Query Language XQL
    </heading>
    <section>
        <heading>Examples</heading>
    </section>
    <section>
        <heading>Syntax</heading>
        Now we describe the XQL syntax.
    </section>
  </chapter>
</book>
```

DTD for example document

```
<!ELEMENT book (author, title, chapter+)>
<!ELEMENT author (#PCDATA)>
<!ELEMENT title (#PCDATA)>
<!ELEMENT chapter (heading,#PCDATA?,section*)>
<!ELEMENT section (heading,#PCDATA?)>
<!ELEMENT heading (#PCDATA)>
<!ATTLIST book
    class  CDATA #REQUIRED
    crdate CDATA #IMPLIED
    type   (monograph|collection|proceedings) "monograph">
```

XML related technologies

- **XML Document Object Model (XML DOM):** standard for accessing and processing XML documents
- The DOM represents elements, attributes and text within elements as nodes in a tree.
- With a DOM API, we can process an XML document by starting at the root element and then descending down the tree from parents to children.
- **XML DOM (XML Document Object Model)** defines a standard way for accessing and manipulating XML documents.

XML related technologies

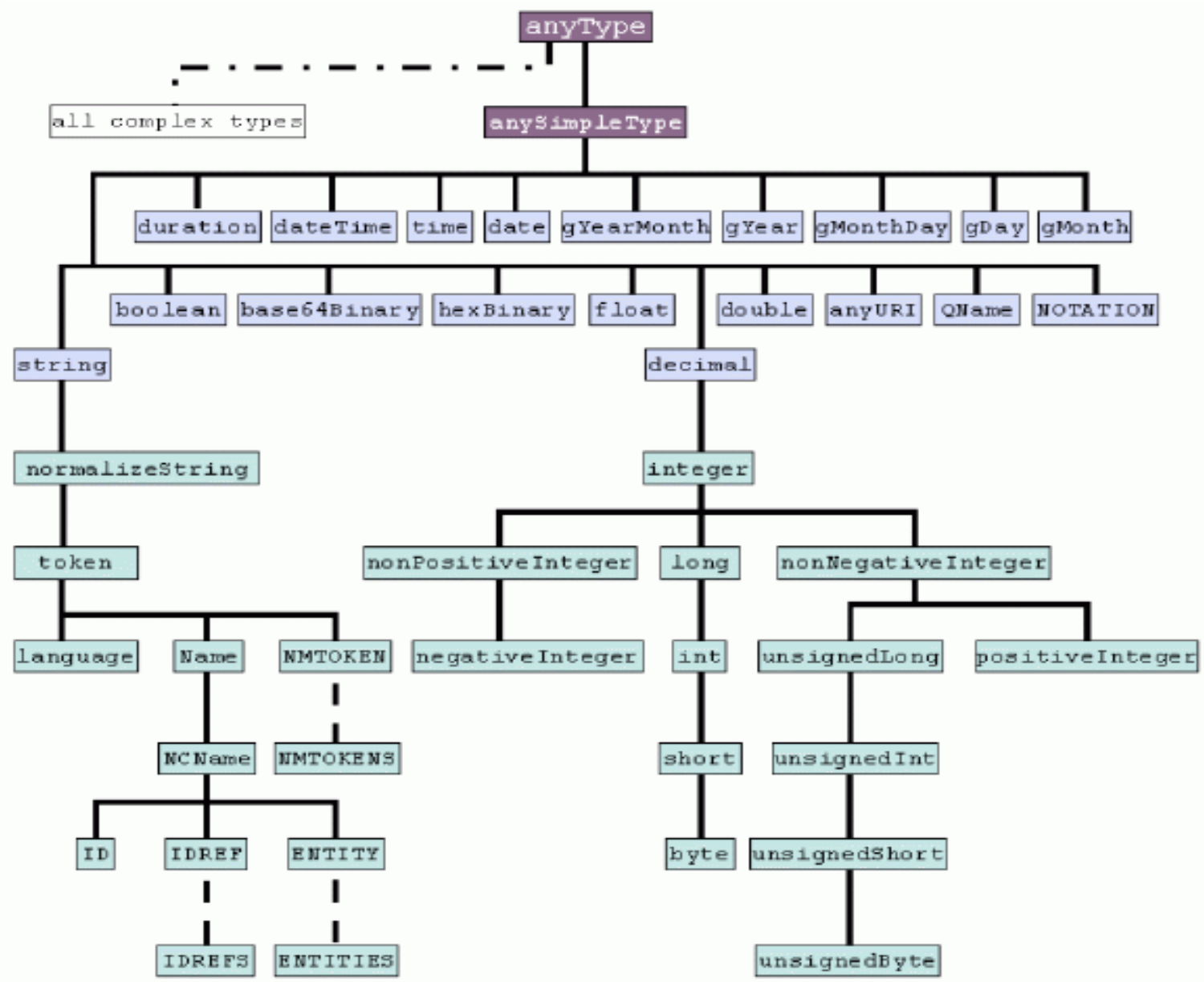
- **XSD (XML Schema)** is an XML-based alternative to DTDs.

XML Schema

resolves weaknesses of DTDs wrt. formatted data:

- support for data types
  (DTDs: only child elements, PCDATA, mixed content and EMPTY)
- specification of structured data
  (e.g. arrays with lower/upper bounds)
- reuse of data-types
  (explicit support for elements only, but not for attributes)
- extensible type system
  (user-defined types / subtypes)
- support for namespaces

XML Schema


anyType
all complex types
anySimpleType
duration
dateTime
time
date
gYearMonth
gYear
gMonthDay
gDay
gMonth
boolean
base64Binary
hexBinary
float
double
anyURI
QName
NOTATION
string
decimal
normalizeString
integer
token
nonPositiveInteger
long
nonNegativeInteger
language
Name
NMTOKEN
negativeInteger
int
unsignedLong
positiveInteger
NCName
NMTOKENS
short
unsignedInt
ID
IDREF
ENTITY
byte
unsignedShort
IDREFS
ENTITIES
unsignedByte

XML related technologies

- **XHTML (Extensible HTML)** is a stricter and cleaner version of HTML.
- **XSL (Extensible Style Sheet Language)** - XSL consists of three parts: XSLT - a language for transforming XML documents, XPath - a language for navigating in XML documents, and XSL-FO - a language for formatting XML documents.
- **XSLT (XSL Transformations)** is used to transform XML documents into other XML formats, like XHTML.

XML-related technologies.

- **XPath** is a language for navigating in XML documents.
- **XSL-FO (Extensible Style Sheet Language Formatting Objects)** is an XML based markup language describing the formatting of XML data for output to screen, paper or other media.
- **XLink (XML Linking Language)** is a language for creating hyperlinks in XML documents.
- **XPointer (XML Pointer Language)** allows the XLink hyperlinks to point to more specific parts in the XML document.
- **XForms (XML Forms)** uses XML to define form data.
- **XQuery (XML Query Language)** is designed to query XML data.

XML-related technologies.

- **SOAP (Simple Object Access Protocol)** is an XML-based protocol to let applications exchange information over HTTP.
- **WSDL (Web Services Description Language)** is an XML-based language for describing web services.
- **RDF (Resource Description Framework)** is an XML-based language for describing web resources.
- **RSS (Really Simple Syndication)** is a format for syndicating news and the content of news-like sites.
- **WAP (Wireless Application Protocol)** was designed to show internet contents on wireless clients, like mobile phones.
- **SMIL (Synchronized Multimedia Integration Language)** is a language for describing audiovisual presentations.
- **SVG (Scalable Vector Graphics)** defines graphics in XML format.
- ***XML parser is used to read, update, create and manipulate an XML document.***

XML Query Languages

XPath Locator Languages

restricted XML query language
retrieves complete elements (subtrees) of XML documents

used in

XSLT (Extensible Style Sheet Language Transformations)
for specifying argument of a transformation

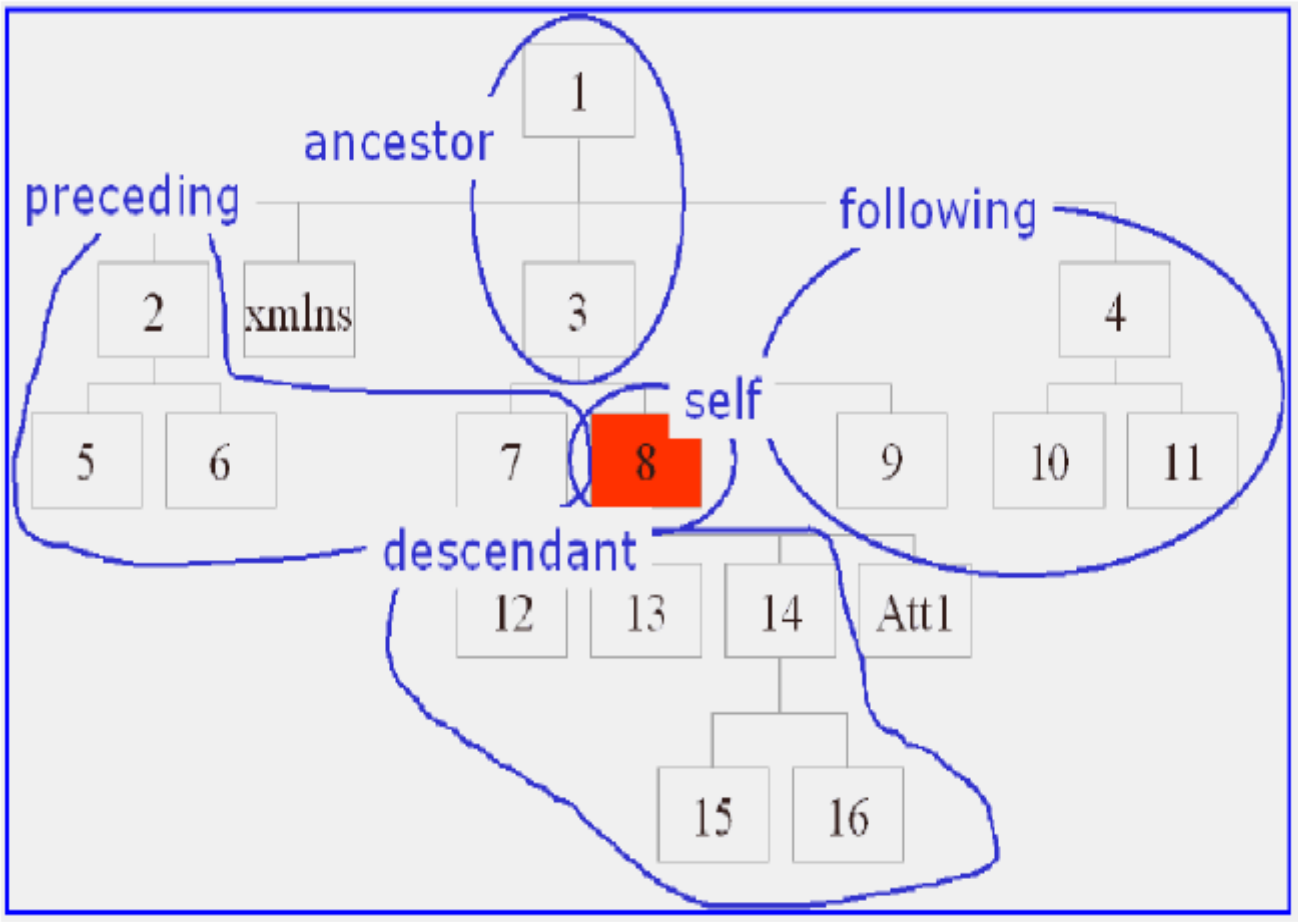
XPointer (XML Pointer)
for defining sources / targets of links

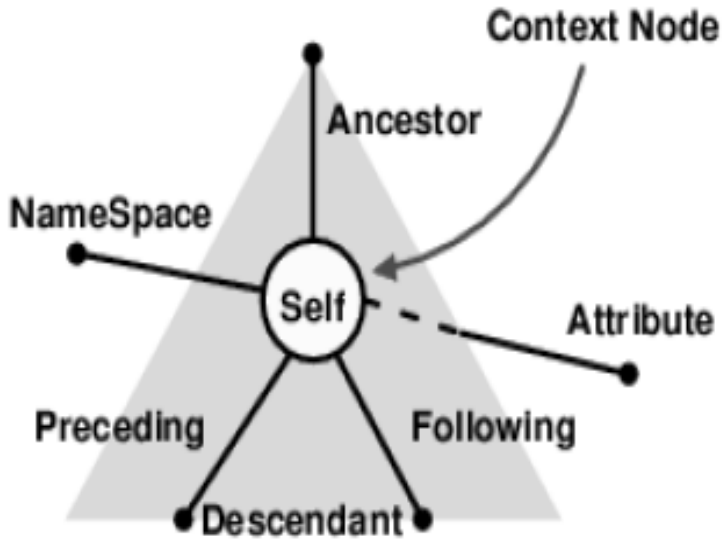
XQuery (XML Query Language)
for selecting elements that are arguments of further operations (value joins, restructuring, aggregation)

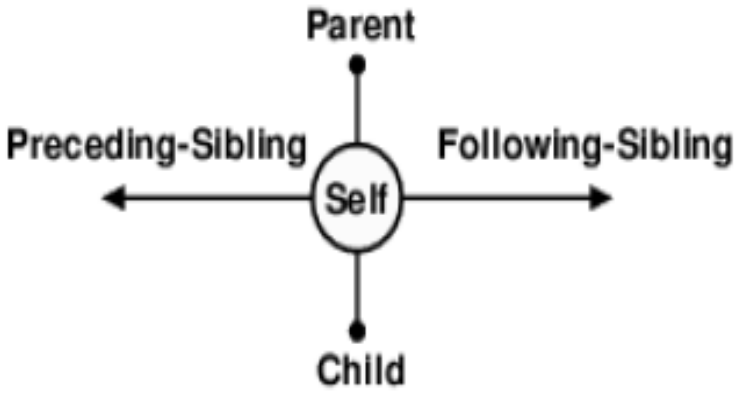
Path Expressions

- search for single elements:
  `heading`
- parent-child:
  `chapter/heading`
- ancestor-descendant:
  `chapter//heading`
- document root:
  `/book/*`
- filter wrt. structure:
  `//chapter[heading]`
- filter wrt content:
  `/document[@class="H.3.3" and author="John Smith"]`

generalization of locator operators







Model: Ordered set of nodes with attributes

IR and relational databases

IR systems are often contrasted with relational databases (RDB).

- Traditionally, IR systems retrieve information from *unstructured text ("raw" text without markup).*
- RDB systems are used for querying *relational data: sets of* records that have values for predefined attributes such as employee number, title and salary.

IR and relational databases

	RDB search	unstructured IR
objects	records	unstructured docs
main data structure	table	inverted index
model	relational model	vector space & others
queries	SQL	free text queries

Why Use XML?

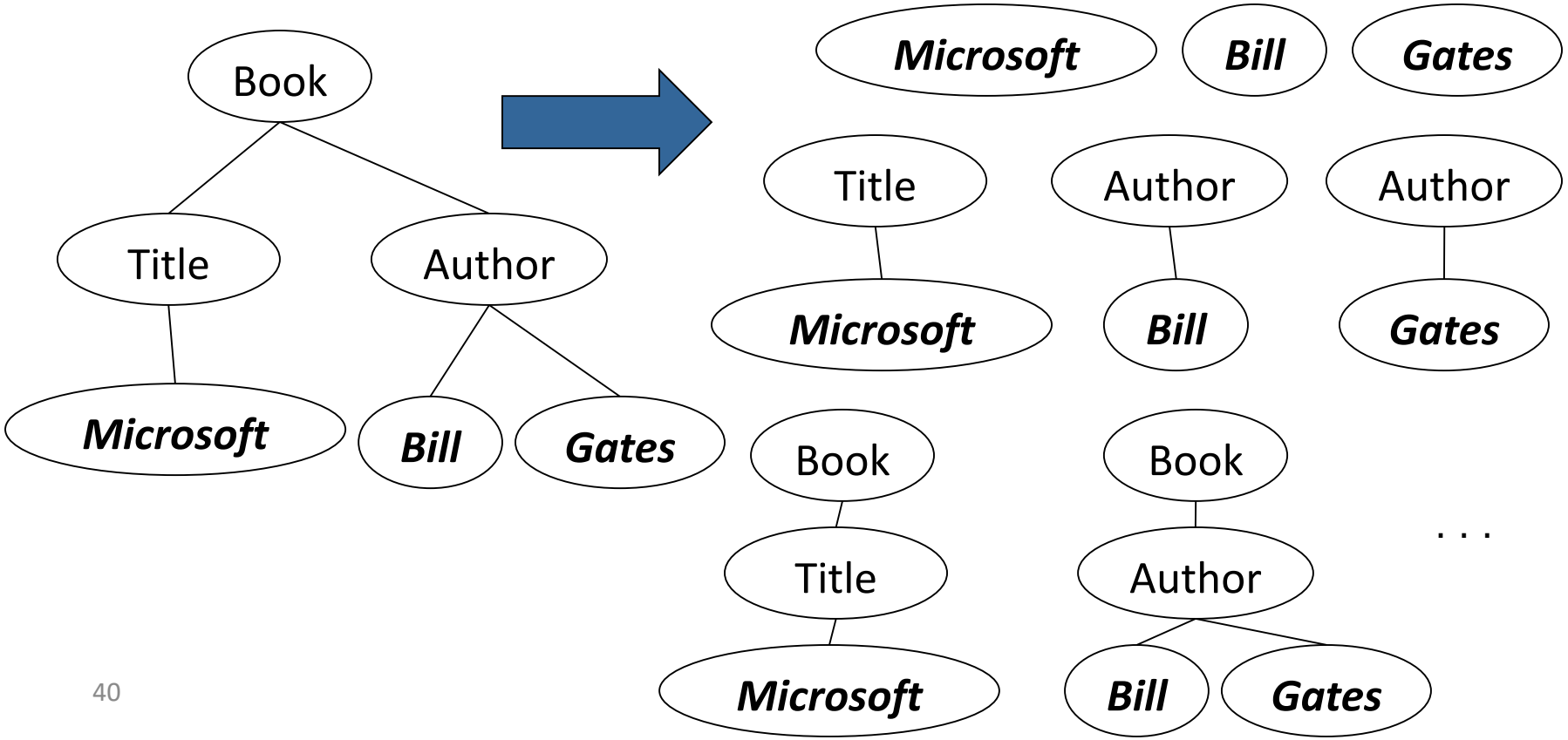
- Represent semi-structured data
  - data that are structured, but don't fit relational model
- XML is more flexible than DBs
- XML is more structured than simple IR
- You get a massive infrastructure for free

Outline

1. Vector space model for XML IR

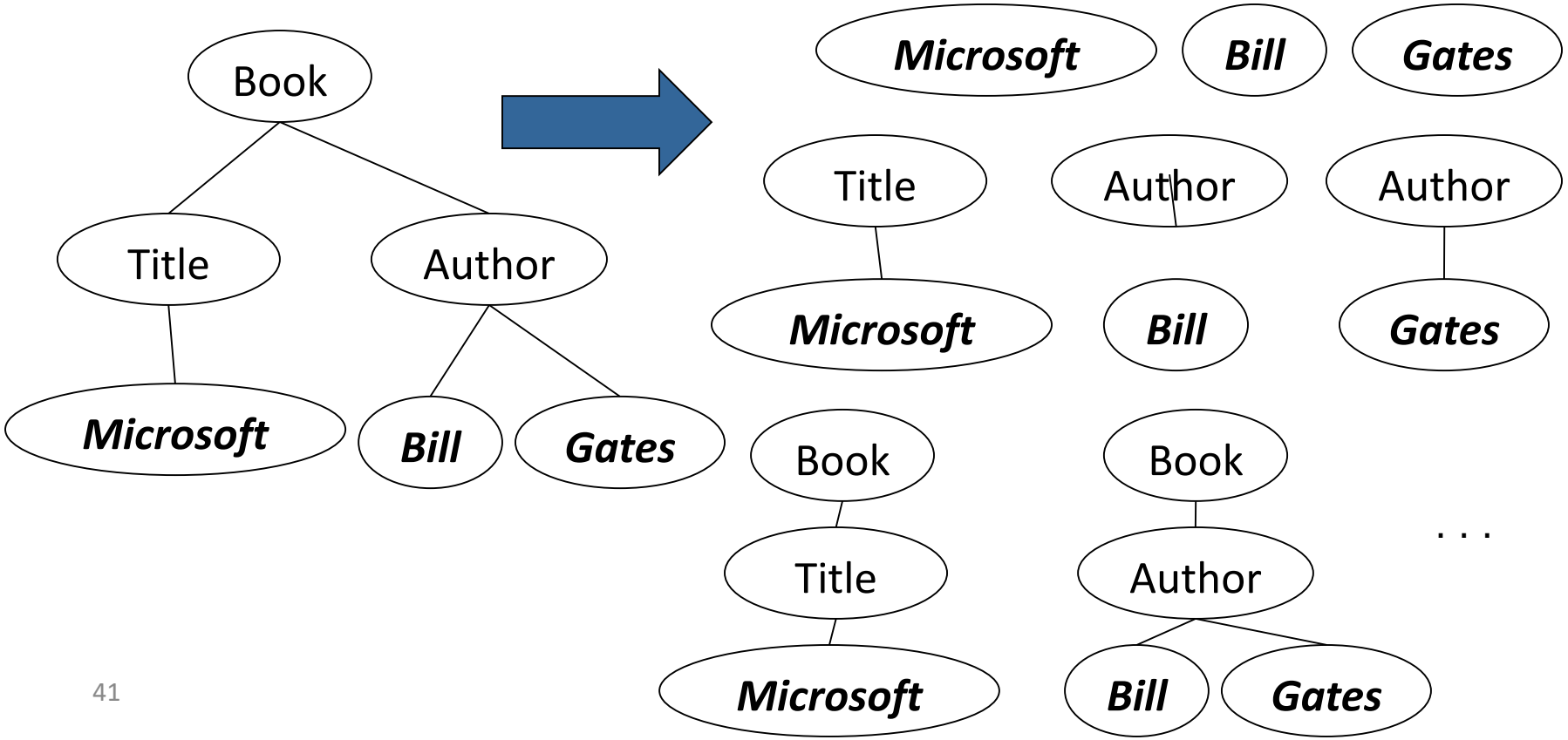
Main idea: lexicalized subtrees

Aim: to have each dimension of the vector space encode a word together with its position within the XML tree.
How: Map XML documents to lexicalized subtrees.



Main idea: lexicalized subtrees

1. Take each text node (leaf) and break it into multiple nodes, one for each word. E.g. split *Bill Gates* into *Bill* and *Gates*
2. Define the dimensions of the vector space to be lexicalized subtrees of documents – subtrees that contain at least one vocabulary term.



Lexicalized subtrees

We can now represent queries and documents as vectors in this space of lexicalized subtrees and compute matches between them,
e.g. using the vector space formalism.

Vector space formalism in unstructured VS. structured IR

The main difference is that the dimensions of vector space in unstructured retrieval are vocabulary terms whereas they are lexicalized subtrees in XML retrieval.

Structural term

There is a tradeoff between the dimensionality of the space and the accuracy of query results.

- If we restrict dimensions to vocabulary terms, then we have a standard vector space retrieval system that will retrieve many documents that do not match the structure of the query (e.g., *Gates* in the title as opposed to the author element).
- If we create a separate dimension for each lexicalized subtree occurring in the collection, the dimensionality of the space becomes too large.

Compromise: index all paths that end in a single vocabulary term, in other words all XML-context term pairs. We call such an XML-context term pair a structural term and denote it by <*c*, *t*>*:* a pair of XML-context *c* and vocabulary term *t*.

Context resemblance

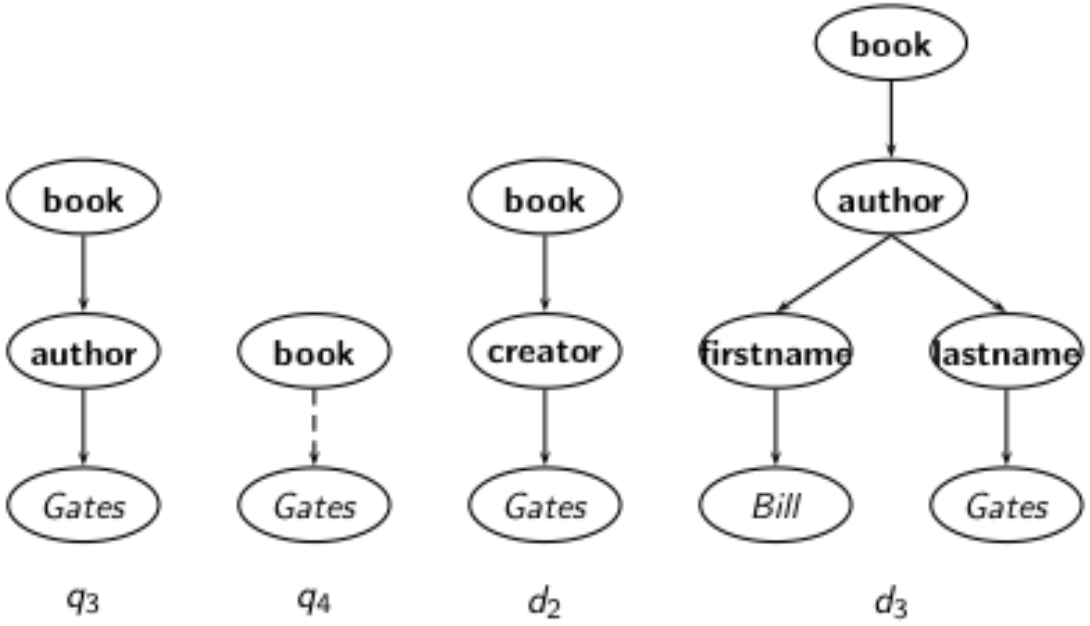
A simple measure of the similarity of a path c_q in a query and a path c_q in a document is the following *context resemblance* function C_R:

$$C_R(c_q, c_d) = \begin{cases} \frac{1+|c_q|}{1+|c_d|} & \text{if } c_q \text{ matches } c_d \\ 0 & \text{if } c_q \text{ does not match } c_d \end{cases}$$

$|c_q|$ a esp.

c_q matches c_d iff we can transform c_q into c_d by inserting additional nodes.

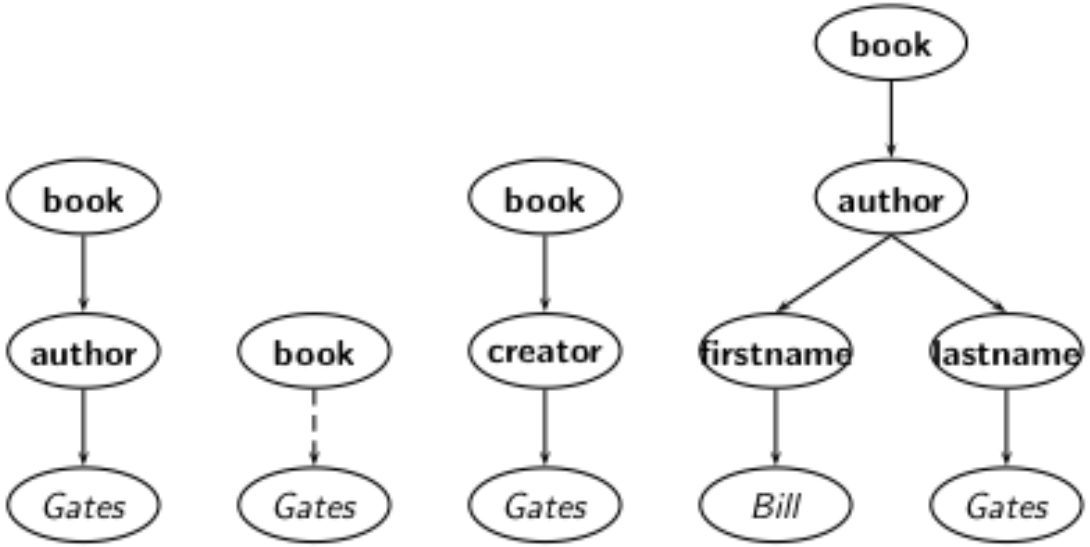
Context resemblance example



$$\mathrm{CR}(c_q, c_d) = \begin{cases} \frac{1+|c_q|}{1+|c_d|} & \text{if } c_q \text{ matches } c_d \\ 0 & \text{if } c_q \text{ does not match } c_d \end{cases}$$

$\mathrm{CR}(c_q, c_d) = 3/4 = 0.75$. The value of $\mathrm{CR}(c_q, c_d)$ is 1.0 if q and d are identical.

Context resemblance example



$$\mathrm{CR}(c_q, c_d) = \begin{cases} \frac{1+|c_q|}{1+|c_d|} & \text{if } c_q \text{ matches } c_d \\ 0 & \text{if } c_q \text{ does not match } c_d \end{cases}$$

$\mathrm{CR}(c_q, c_d)$ = ? $\mathrm{CR}(c_q, c_d)$ = 3/5 = 0.6.