Jul – Nov 2022
IT458

Structured Information Retrieval

IR systems v/s Relational databases (RDB)

- *IR systems* - retrieve information from unstructured document collections.

- *RDB systems* - used for querying structured relational data
  - Consists of sets of records that have values for predefined attributes such as employee number, title and salary.

Why Structured IR?

- Text documents often contain structural information
- can be exploited at several stages of the IR process
  - Indexing stage
  - Querying stage
  - Retrieval stage
  - Result presentation stage

Structured Retrieval

- Basic premise:
  - queries are structured or unstructured;
  - documents are structured (or contain structured data that can be used).
- Suitable for –
  - data sources containing text that is best modeled as structured documents rather than relational data.
- Applications -
  - Digital libraries, patent databases, research publications, blogs…
  - tagged text with entities like persons, identifiers and locations (named entity tagging)

Structured Retrieval

- Categorized according to three main aspects
  - explicit vs. implicit structure of the docs
  - static vs. dynamic structure
  - single vs. multiple hierarchical structure

Structured Retrieval

explicit vs. implicit structure

- *Explicit structure:* if the documents are composed of sections, chapters, titles, etc.
  - structure is usually provided through the use of a markup language
  - For example,

    section CONTAINS "nitk beach"

    - Will return all sections that contain the sentence "nitk beach"

Structured Retrieval

explicit vs. implicit structure

- *Implicit structure:* if document structure is not explicitly distinguished from its text content
  - Documents are modeled as sequences of tokens without distinguishing a word token from a markup token

Structured Retrieval

explicit vs. implicit structure

- *Implicit structure:* if document structure is not explicitly distinguished from its text content
  - Documents are modeled as sequences of tokens without distinguishing a word token from a markup token

- A structural element is therefore constructed at querying time
  - Example:

    ("<section>" FOLLOWING "</section>") CONTAINS "nitk beach"
    - The section element only exists at querying time

Structured Retrieval

static vs. dynamic structure

- *Static structure:*

```
<?xml version="1.0" encoding="UTF-8"?>
<breakfast_menu>

<food>
<name>Belgian Waffles</name>
<price>$5.95</price>
<description>Two of our famous Belgian Waffles with plenty of real maple syrup</description>
<calories>650</calories>
</food>

<food>
<name>Strawberry Belgian Waffles</name>
<price>$7.95</price>
<description>Light Belgian waffles covered with strawberries and whipped cream</description>
<calories>900</calories>
</food>

<food>
<name>Berry-Berry Belgian Waffles</name>
<price>$8.95</price>
```

Structured Retrieval

static vs. dynamic structure

- *Dynamic structure:*

```
<?xml version="1.0" encoding="UTF-8"?>
<breakfast_menu>

<food>
<name>Belgian Waffles</name>
<price>$5.95</price>
<description>Two of our famous Belgian Waffles with plenty of real maple syrup</description>
<calories>650</calories>
</food>

<food>
<name>Strawberry Belgian Waffles</name>
<price>$7.95</price>
<description>Light Belgian waffles covered with strawberries and whipped cream</description>
<calories>900</calories>
</food>

<food>
<name>Berry-Berry Belgian Waffles</name>
<price>$8.95</price>
```

Structured Retrieval

static vs. dynamic structure

- *Dynamic structure:*

```
<?xml version="1.0" encoding="UTF-8"?>
<breakfast_menu>

<food>
<name>Belgian Waffles</name>
<price>$5.95</price>
<description>Two of our famous Belgian W
<calories>650</calories>
</food>

<food>
<name>Strawberry Belgian Waffles</name>
<price>$7.95</price>
<description>Light Belgian waffles cover
<calories>900</calories>
</food>

<food>
<name>Berry-Berry Belgian Waffles</name>
<price>$8.95</price>
```

Example XSLT Stylesheet:

```
<?xml version="1.0" encoding="UTF-8"?>
<html xsl:version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<body style="font-family:Arial;font-size:12pt;background-color:#EEEEEE">
<xsl:for-each select="breakfast_menu/food">
  <div style="background-color:teal;color:white;padding:4px">
    <span style="font-weight:bold"><xsl:value-of select="name"/> - </span>
    <xsl:value-of select="price"/>
    </div>
  <div style="margin-left:20px;margin-bottom:1em;font-size:10pt">
    <p>
    <xsl:value-of select="description"/>
    <span style="font-style:italic"> (<xsl:value-of select="calories"/> calories per serving)</span>
    </p>
  </div>
</xsl:for-each>
</body>
</html>
```

Structured Retrieval

static vs. dynamic structure

- *Dynamic structure:*

Example XSLT Stylesheet:

```
<?xml version="1.0" encoding="UT
<html xsl:version="1.0" xmlns:xs
<body style="font-family:Arial;f
<xsl:for-each select="breakfast_
  <div style="background-color:t
    <span style="font-weight:bol
    <xsl:value-of select="price"
    </div>
  <div style="margin-left:20px;m
    <p>
    <xsl:value-of select="descri
    <span style="font-style:ital
    </p>
  </div>
</xsl:for-each>
</body>
</html>
```

Belgian Waffles - $5.95

Two of our famous Belgian Waffles with plenty of real maple syrup *(650 calories per serving)*

Strawberry Belgian Waffles - $7.95

Light Belgian waffles covered with strawberries and whipped cream *(900 calories per serving)*

Berry-Berry Belgian Waffles - $8.95

Light Belgian waffles covered with an assortment of fresh berries and whipped cream *(900 cal*
serving)

French Toast - $4.50

Thick slices made from our homemade sourdough bread *(600 calories per serving)*

Homestyle Breakfast - $6.95

Two eggs, bacon or sausage, toast, and our ever-popular hash browns *(950 calories per servi*

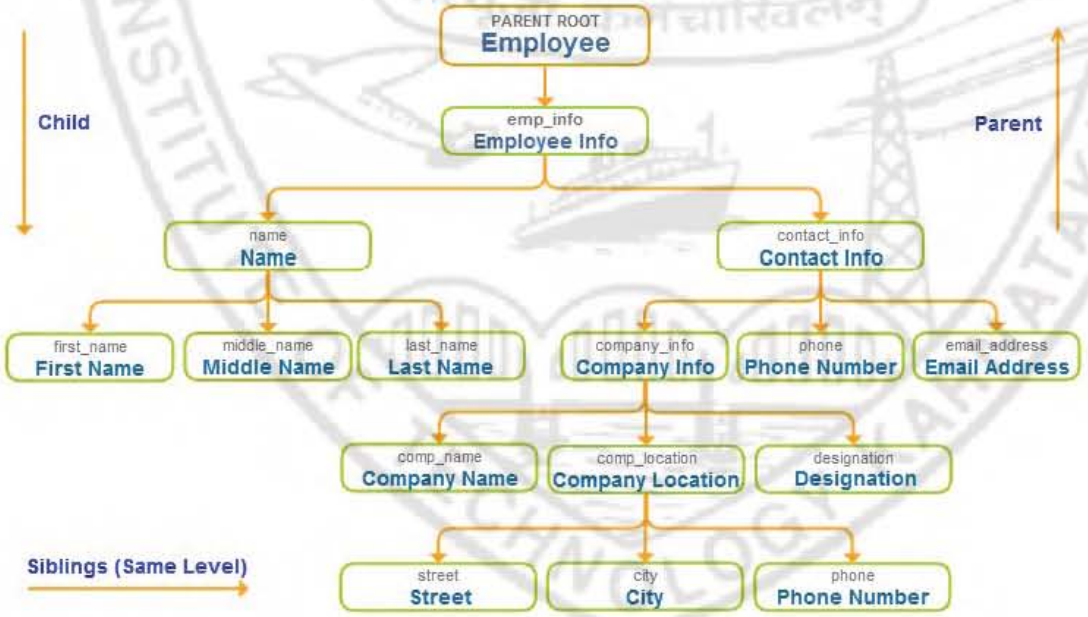
Structured Retrieval

single vs. multiple hierarchies

- Text retrieval models using implicit structure assume a single hierarchy.

  e.g. ("<section>" FOLLOWING "</section>") CONTAINS "nitk beach"

- Approaches based on explicit structure assume that multiple structural hierarchies are present on the same document.



Structured IR Models

- XML is popularly adopted as the markup language for structured documents.

- INEX (INitiative for the Evaluation of XML Retrieval)
  - Provides test collections for focused retrieval of content and structure.
  - Provides a forum for the evaluation and comparison of XML retrieval approaches.

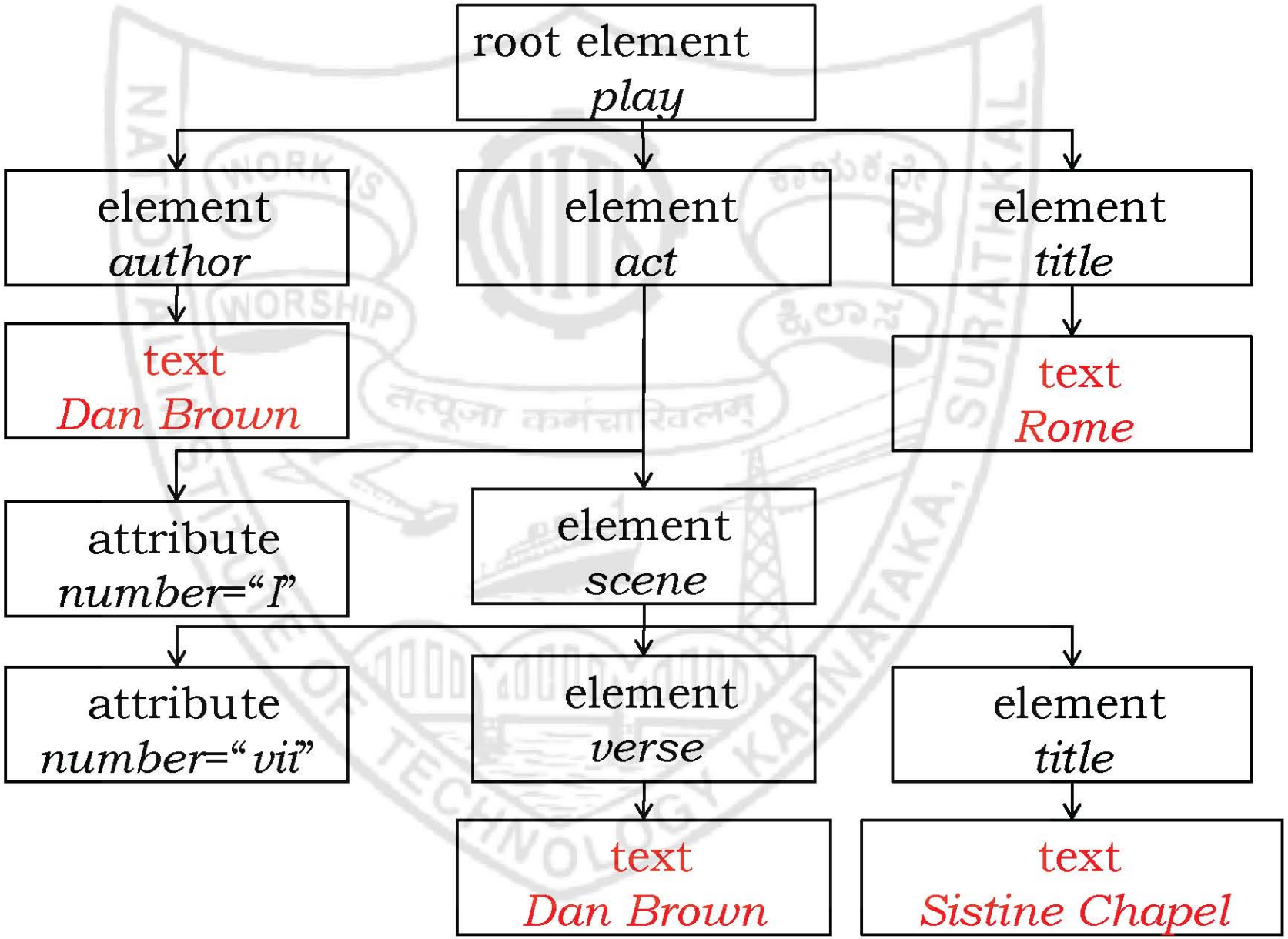
XML document

- Ordered, labeled tree
- Each node of the tree is an XML element, written with an opening and closing XML tag
- 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>Dan Brown</author>
<title>Rome</title>
<act number="I">
<scene  number=""vii">
<title>Sistine Chapel</title>
<verse>Will I with wine
…</verse>
</scene>
</act>
</play>
```

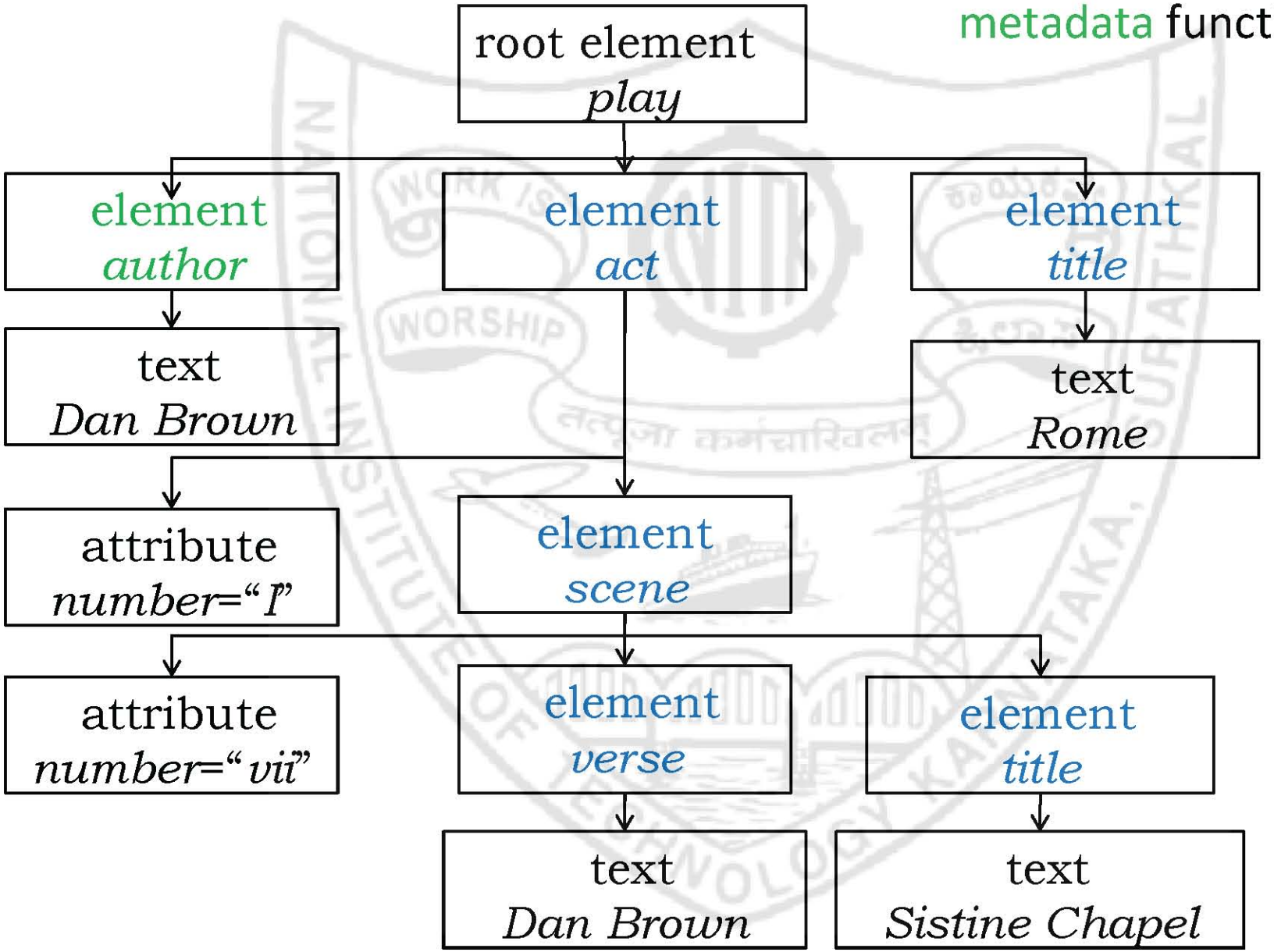
XML Document

Leaf nodes consist of text



XML Document

The internal nodes encode document structure or metadata functions



Structured IR Models (contd.)

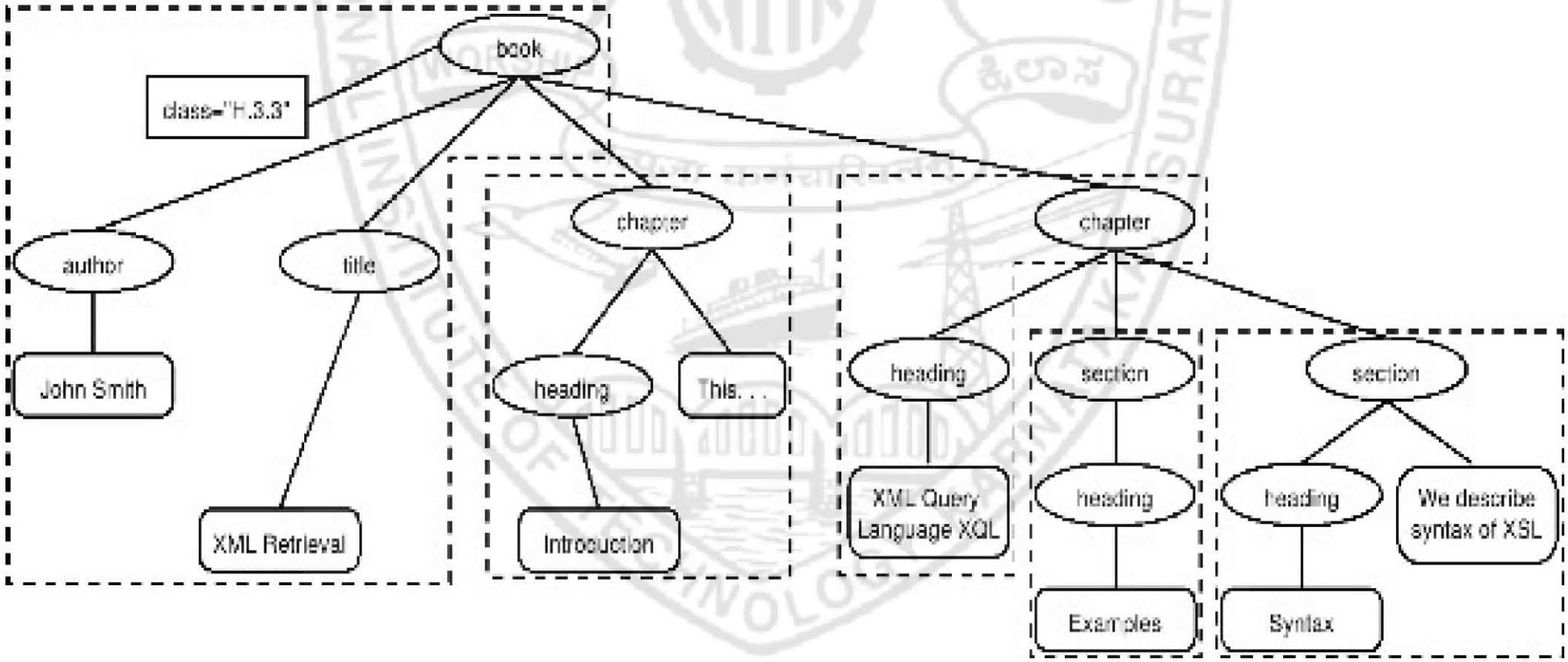
- Defines a retrieval strategy that –
  - returns the smallest unit that contains the information sought.
  - called *document unit* or *indexing unit*.
    - the central notion for indexing and ranking in structured IR:

Structured IR - Document representation

- Different approaches to defining the indexing unit
  - non-overlapping pseudo-documents
  - top down
  - bottom up
  - Hybrid
  - …

Structured IR - Document representation (contd.)

- **Approach: Group nodes into non-overlapping pseudo-documents.**
  - *Indexing units*: e.g. books, chapters, section, but without overlap.



Structured IR - Document representation (contd.)

- **Top down Approach**
  - Start with one of the latest elements as the indexing unit.
    - e.g. the *book* element in a collection of books
  - *Issue*: often fails to return best subelement
    - E.g the relevance of a whole book is often not a good predictor of the relevance of small subelements within it.

Structured IR - Document representation (contd.)

- **Bottom up Approach**
  - Instead of retrieving large units and identifying subelements *(i.e., top down)*, consider all leaves, select the relevant ones as per predefined criteria and then extend them to larger units during postprocessing.
  - *Issue:* Similar problem as top down -- the relevance of a leaf element is often not a good predictor of the relevance of elements it is contained in.

Structured IR Model : Query language

- Types --
  - Tag –based queries
  - Path-based queries
  - Clause-based queries

Structured IR Model : Query language

- Tag –based queries
  - use annotated words that specify a structural constraint.
  - E.g. section: nitk beach
    → "retrieve sections about nitk beach"

Structured IR Model : Query language

- Path-based queries:
  - Uses Xpath/NEXI* syntax to match document structure wrt query.
  - E.g. //document[section[about(.,Surathkal)]//section[about(.,beach)]

*Narrowed Extended XPath I (NEXI)

Structured IR Model : Query language

- Clause-based querying
  - use nested clauses to express information needs, very similar to SQL.
    - Can use XQuery syntax for querying
  - E.g. for $x in /document/section
    where $x/title=NITK
    return $x/section

Some Ranking mechanisms

- ***Context Resemblance* function CR –**

  - measure of the similarity of a path c_q in a query and a path c_q in a document:

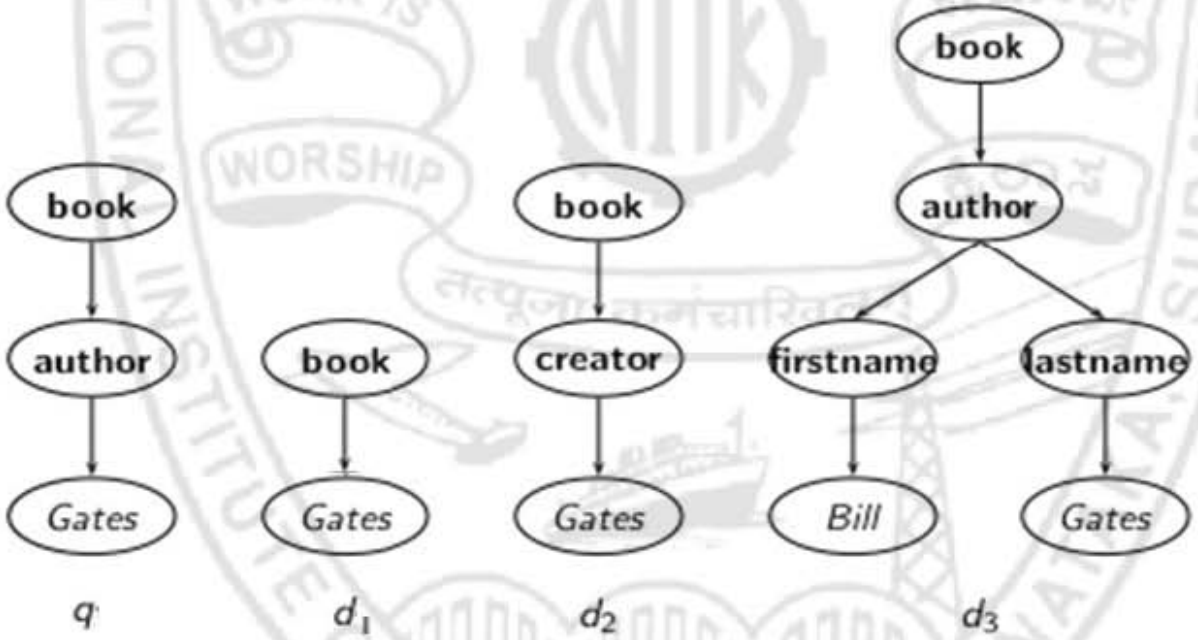
$$\mathrm{CR}(c_q, c_d) = \begin{cases} \frac{1+|c_q|}{1+|c_d|} & \text{if } c_q \text{ matches } c_d \\ 0 & \text{if } c_q \text{ does not match } c_d \end{cases}$$

- $|c_q|$ and $|c_d|$ are the number of nodes in the query path and document path, respectively.

Ranking (example)

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

where, $|c_q|$ and $|c_d|$ are the number of nodes in the query path and document path, resp.



$\mathrm{CR}(c_q, c_{d1}) = 0$

$\mathrm{CR}(c_q, c_{d2}) = 0$

$\mathrm{CR}(c_q, c_{d3}) = 1+2/1+3 = 0.75.$

Further reading...

- Fuhr, N., Gövert, N., Kazai, G., & Lalmas, M. (2002, August). INEX: INitiative for the Evaluation of XML retrieval. In *Proceedings of the SIGIR 2002 Workshop on XML and Information Retrieval* (Vol. 2006, pp. 1-9).

- Sigurbjörnsson, Börkur, Jaap Kamps, and Maarten de Rijke. "An element-based approach to XML retrieval." *INEX 2003 Workshop Proceedings*. 2004.

- Le Maitre, Jacques. "Indexing and querying content and structure of xml documents" *Proceedings of the IADIS International Conference WWW/Internet*. .