

**Dublin City University
School of Computing**

CA4009: Search Technologies

Section 2: Hypertext, Metadata and XML

Gareth Jones
September 2016

Introduction

In Section 1, we introduced the use of links between content to link related materials, and the use of annotation to describe content to support searching for them.

This section introduces the subject of hypertext which establishes many of the conceptual principles of content linking, metadata which provides a means to label content, and XML which provides formal mechanisms to capture content descriptions and their relationships

An understanding of these concepts forms an important component of a number of topics later in this module.

Introduction

This section of the module introduces:

- Definition of hypertext and its relationship to search applications.
- Metadata as a means of content description for search.
- XML as a technology to support implementation of content annotation and its use in search applications.

Definitions of Hypertext

A hypertext contains content elements marked up with links to other content elements.

Hypertext elements can contain links within the information pointing to other information based on the content of the source and the target elements.

Perhaps the most obvious reasons for a link is that the target in some way describes the source of the link.

Links may be placed in various ways. They may connect the source of the link:

- to elsewhere in the same content element.
- to another content element in the same location.
- to another content element at a different location.

Web pages provide examples of all these potential link types.

Definitions of Hypertext

A software application designed to view a hypertext data is referred to as a *hypertext browser*

These browsers may provide various views of the data and encourage reading in non-linear manner.

The World Wide Web is essentially a hypertext or more fully a hypermedia.

Hypermedia combines text, images, audio, video.

The web does not define hypertext, and it is really an implementation of a subset of hypertext concepts.

Sequential Text

Traditional print media generally focuses on sequential presentation of information.

A typical example is a magazine which is a collection of distinct articles.

- the articles can be read in any order.
- most people will read magazines in unique orders.

Sequential Text

While readers may have a unique experience in the order in which they consume the content of a magazine, the spatial layout of the magazine is clear.

- Generally a magazine has a “contents page” to aid “jumping” among articles.
- Advertisements clearly designed to grab our attention.
- Sidebars containing additional information.
- Picture captions.
- Article splitting e.g. “go to page 23 for the rest of this article”

Definitions of Hypertext

Hypertext enables:

- computer-supported non-linear viewing of information
 - where a reader can view information in any order they desire by “jumping” into the hypertext wherever they like,
 - further jumping (navigation) from link to link until they are satisfied.
- a way to organise information
 - not using the normal organisation structures such as hierarchical, e.g. chapters, sections, and subsections.

Definitions of Hypertext

Hypertext is a way of supporting associative linking of information, without the constraints of linear (or hierarchical) organisation.

One way of viewing this is as having unlimited footnotes and cross references.

There is no physical constraint on the number or size of these links, so as many of these in as much detail as needed can be embedded as desired in a hypertext,

These can be used as in a traditional document by constructing footnotes and references to add understanding, or treated as a hypertext element through which the reader navigates.

You can't read a hypertext in a traditional linear manner, but you follow the

links in a browsing fashion to find information.

Where to begin in a Hypertext

A true hypertext has no start point and no end point. Consider for a moment ... where does *wikipedia* begin? where does the web begin? ... where do they end?

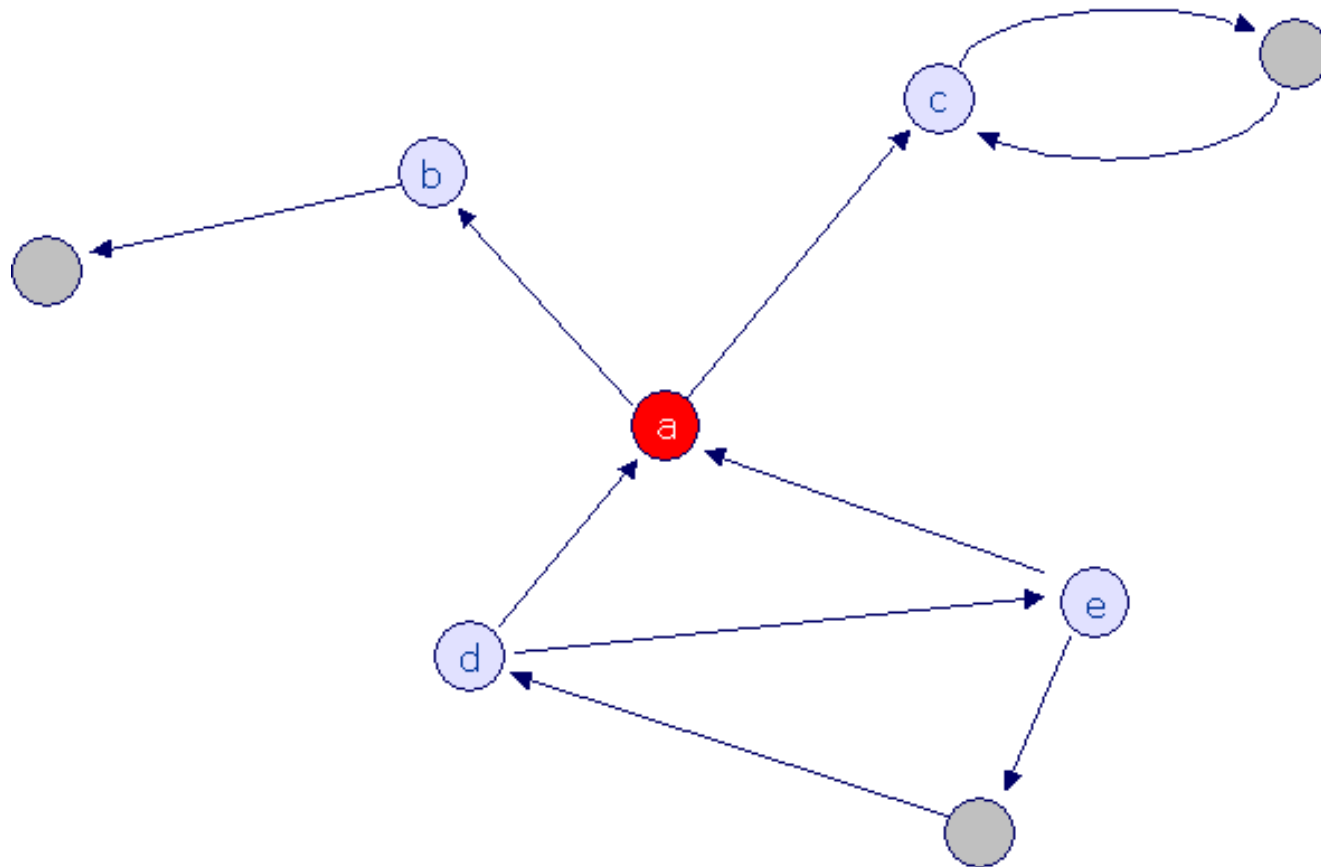
Where should you begin reading the hypertext?

One way to view the role of a search engine in a hypertext is to provide a list of potential entry points into a hypertext.

The search engine provides a number of suggestions of where to jump into the hypertext to start reading it?

Jumping into a Hyperspace

The suggested entry points are the notes that the search engine judges to be most likely to provide content relevant to the user's information need.



Search as a dynamic index

One way to view the output of a search engine is as providing a dynamic index on the subject (or subjects) of the search request.

The dynamically created list provides potential jump in points for the hypermedia indexing by the search engine.

Components of Hypertext

Hypertext contains:

- **Nodes**

- are chunks of information.
- the basic unit of information in a hypertext system; this unit may be decomposed into smaller units.

An HTML web page is a node.

Standard sequential text may be displayed within a node.

- **Links**

- connect two (or more) nodes.
- best example is a web link in HTML.

Components of Hypertext

Hypertext contains:

- **Anchors**

- are persistent selections in documents/nodes.
- they can be:
 - * highlighted words,
 - * phrases,
 - * strings in text.
 - * icons,
 - * tgraphic/image (“hotspots”).
- they *represent* the source (or destination) of an information link. (This about this idea, this is an important point.)

Links in a Hypertext

- What should they look like?
- How are they to be embedded into a document?
- There is also uncertainty as to whether links should be:
 - from a specific anchor to a complete node (like HTML).
 - from a specific anchor to another specific anchor (like HTML)
 - one-to-one or one-to-many, called multi-links (in XML).
 - unidirectional or bi-directional (bi in XML)
 - regional, to a region of a page, allowed in XML

anchors in a hypertext

- anchors represent the source (or destination) of a hyperlink.

How should they look?

- They should always be instantly identifiable.
 - but should they always look the same within a hypertext?
- What if there are distinct types of targets?
 - should these be displayed differently?
 - this will introduce the notion of *typed* links.
- Can the link sources indicate the type of target node to a user before the link is followed?

Information in Anchortext

- The text in an anchor (referred to as “Anchortext”) can provide a link to another node which gives a detailed indication of the content of the link item,
e.g. a link to a review of the movie Avatar.

In this case the anchor text of the source document containing the link is “review of the movie Avatar” ... this gives us a reasonable idea of the content of the target node...

- but if the link to the same page just says “link” to the same review, or another just says “Avatar review here”.

What information do we (or an automatic analysis program) have about the target node now?

Exploiting Anchortext

As well as indicating to a user the contents of the document/item pointed to by a link, anchortext can be used for other purposes in information retrieval:

- Indicating the contents of non-text items, e.g. the contents of an image picture of The Sunflowers, probably points to a picture of sunflowers, and the anchortext can be used to label the image.
- It can be used as a representation of a document that has not been crawled by a web spider. The document at an `http` address that has not been indexed, can be represented by the anchortext that points to it.
- Contents of a document which has been indexed can be augmented by the anchortext of links which point to it.

Problems with Hypertext

- When should a link be inserted?
- Difficulty of reading.
- Difficulty of authoring.
- Disorientation:
 - Where am I?
 - How did I get here?
 - How do I get back?
 - How do I get to that page on X?

Solutions to Disorientation

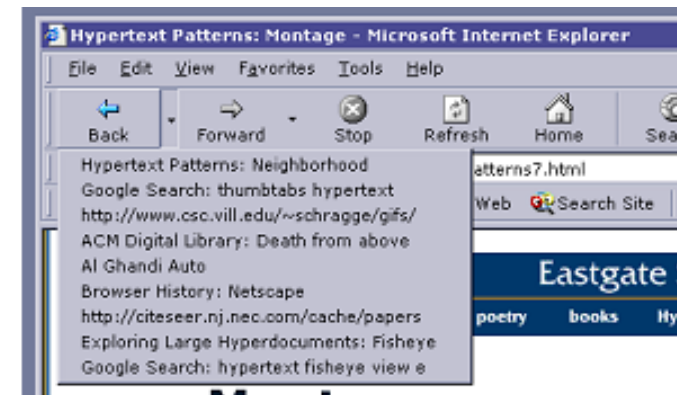
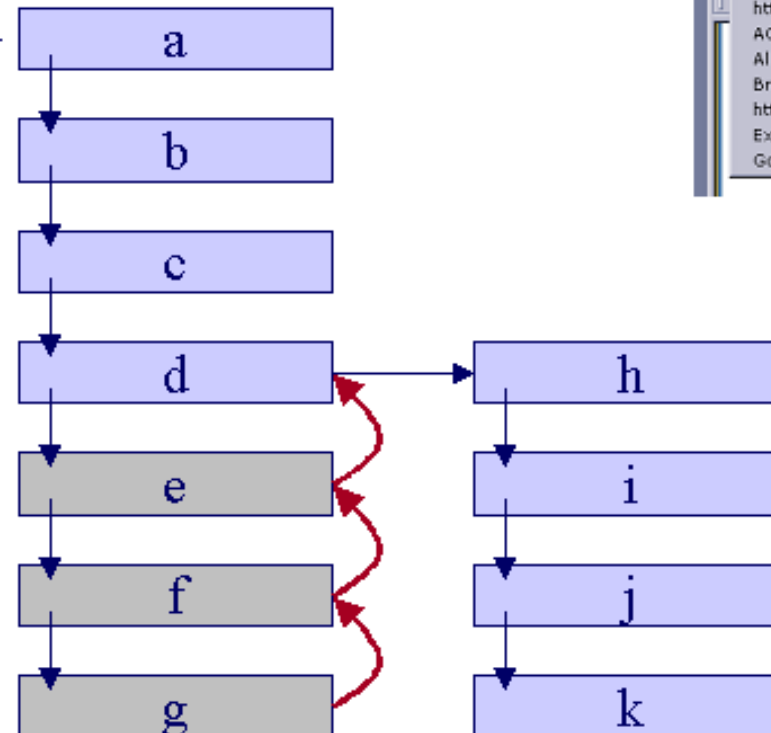
- Graphical browsers
 - multi-windows - enable traversal or movement between node and orientation.
- “Go Back” command:
 - requires a stack listing where to go back to.
- History lists:
 - go back to the page which contains ...
 - list is limited in length – perhaps be able to indicate important documents for longer storage – leads to the idea of bookmarks.
- Bookmarks:
 - mark current place, go for a wander, click on bookmark to return.

Solutions to Disorientation

- Search Facilities:
 - to provide a 'good starting point' to explore a hypertext structure.
 - find your way back to a particular node again.
- Fisheye views:
 - local data presented in detail, remote data is abstract.
- Margin Notes:
 - personalised notes which should be iconised and individualised.
- Breadcrumbs, coffee stains:
 - to indicate that the reader was here recently, similar to history.

Go Back command

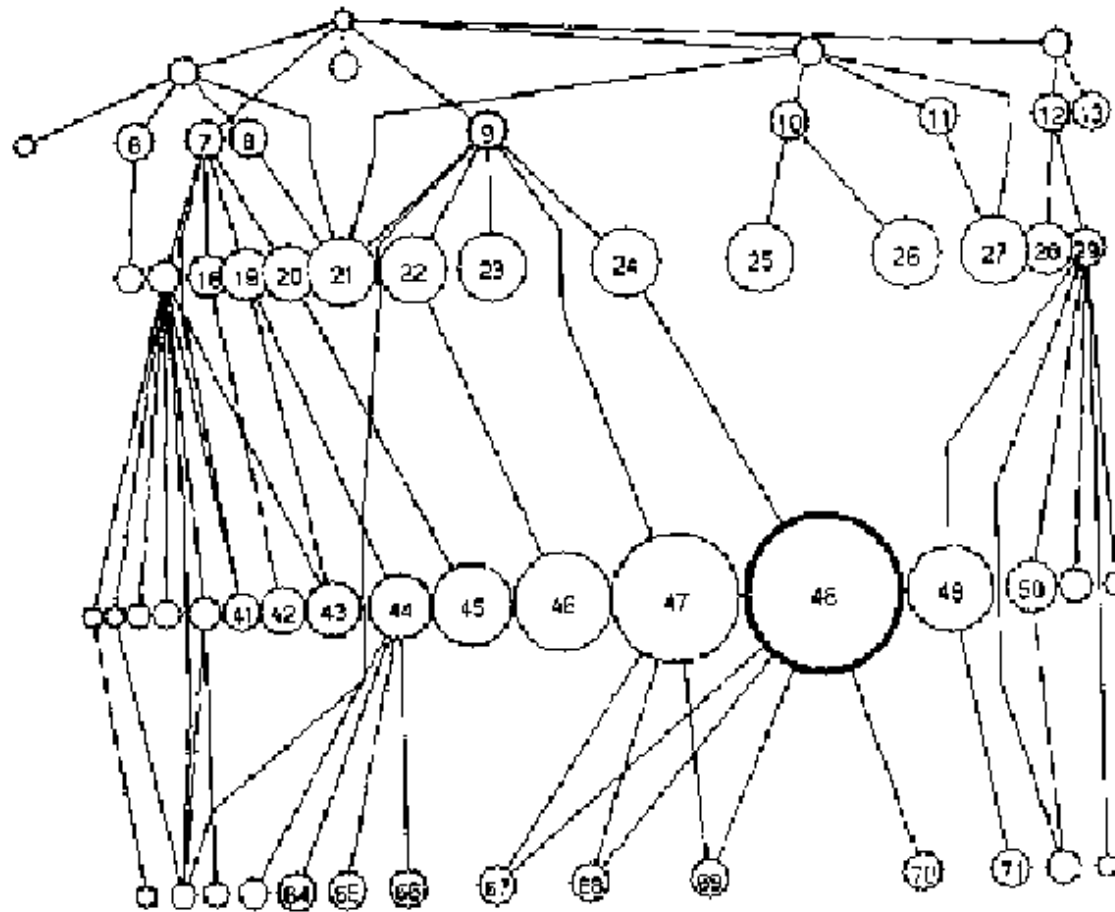
Node at which
The hypertext
Session
began →



Montage

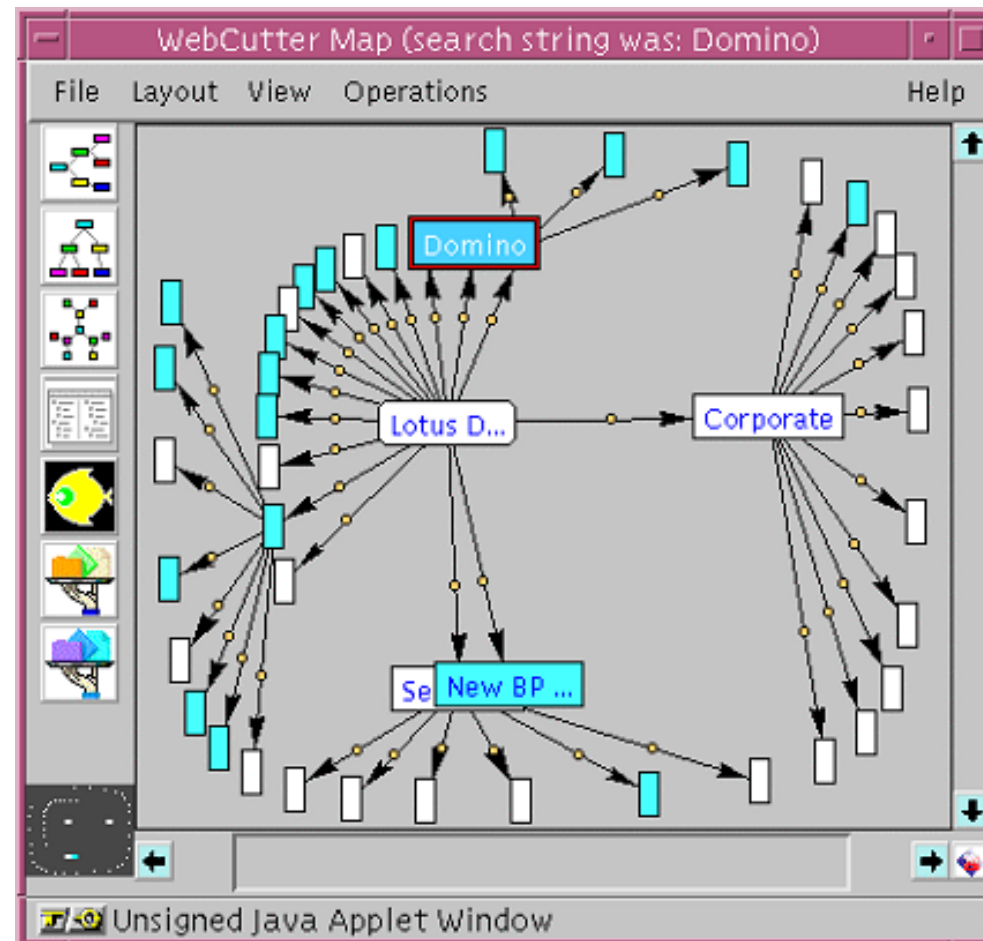
In **Montage**, several distinct windows appear simultaneously, reinforcing each other while retaining their separate identities. Montage is frequently effected through superimposition, which establishes connections across the boundaries of nodes and links. Montage is prominent in the hypertexts of George P. Landow [47, 51], each

Fisheye Views



Fisheye Views

WebCutter example (web link)



The Golden Rules of Hypertext Generation

When creating a hypertext it is advisable to adopt the following rules:

- organise the source data into fragments that can be linked together.
- ensure that the fragments relate to one another.
- ensure that the the user needs only a small fraction of the fragments at any one time.

Hypertext and the WWW

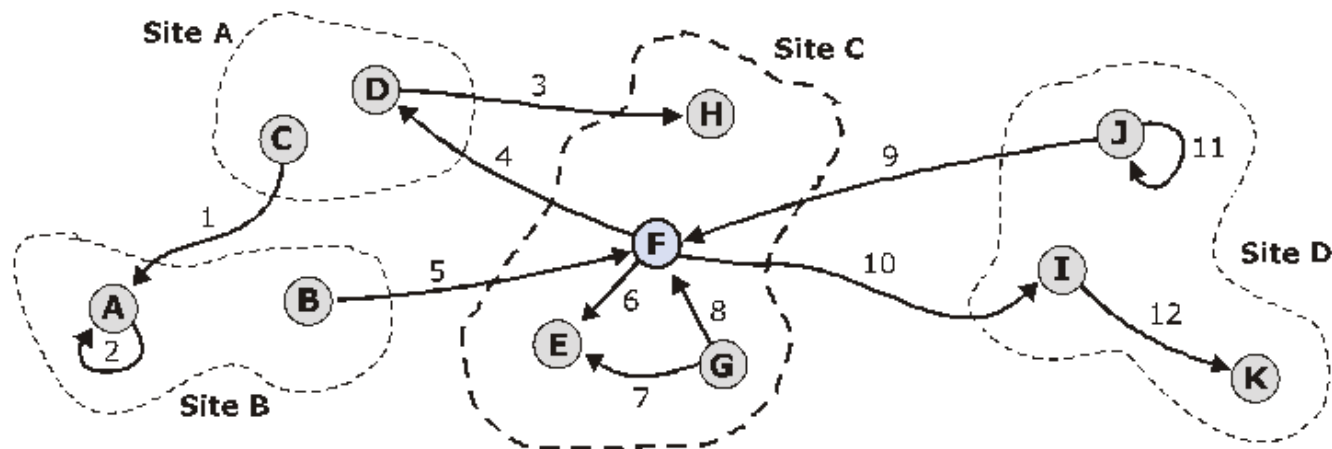
Nodes (HTML web pages) are not of the same size.

Links in HTML are:

- Unidirectional.
- One to one
- May point at:
 - entire HTML pages.
 - files of various media image, audio, video, etc.
 - sections of a document using the (# fragment identifier)
 - mail addresses.

HTML Link Types

in-link to doc F:	5,8,9
out-link from doc F:	4,6,10
self-links:	2,11
on-site links:	6,8,12
off-site links:	1,2,3,4,9,10
on-site in-links to doc F:	?
off-site out-links of doc F:	?



Links in the WWW

The web is very large - billions of documents. If we rely only on the user query and the contents of the documents for search, why is this likely to cause problems when trying to find relevant documents based on a standard user query?

What do the links between documents on the web tell us?

How might we use this information to provide more effective web search?

What issues or problems could this strategy create?

Definitions of Metadata

As we saw in Section 1, metadata provides a way of describing the contents of items which can be used to support search.

Metadata is machine readable information about media content.

The media may be audio, image, video, or text(!).

“an element of metadata describes an information resource, or helps provide access to any information resource.”

Metadata may be included in a file with the item being described, or it may be stored separately and refer to the item.

Metadata can itself refer to other metadata.

Real World scenarios using Metadata

- Library book request by keywords, title, author, ISBN, subject, etc.
- Video file request by title, director, actors, genres, etc.
- Golden Pages business directory lookup company name, product section, address, etc.
- many, many other examples, ...

Can be created automatically or manually. Can use a controlled or fixed vocabulary, standardised features (e.g. dates), free text (can introduce mistakes, typos, etc.).

Metadata can be used to support search or for filtering (e.g. by date).

SGML

Metadata for an information element needs to be stored in a form that makes its purpose (i.e. what it describes) clear, and make the individual information elements easily accessible.

- SGML: Standardised Generalised Mark-up Language: an international standard that describes a generalized markup scheme for representing document structure and content in a system and platform independent way.
- A language that provides facilities for *defining* mark-up languages.
 - it is not a language, it is a meta-language for defining mark-up languages.
- Tags are stored in a DTD (Document Type Definition) file.

SGML

- Specialised software must be produced in order to visualise SGML.
- SGML is used for working with and archiving electronic texts.
- SGML is the basis for defining markup languages, including HTML.
 - HTML is defined by an SGML DTD.

XML

- XML: eXtensible Markup Language: A simplified subset of SGML.
- A protocol for designing mark-up languages.
- A family of technologies that can do everything from formatting documents to aiding the filtering of data and transferring of data between applications.
- A philosophy for information handling that seeks maximum usefulness and flexibility for data by refining it to its purest and most structured form.
- A defined markup language is used to contain and manage the information.

XML facilitates

- The storage and organisation of information in a form tailored to the individual needs of each user.
- Rules for syntax checking, link checking, correctness validation.
- The easy reading and understanding by humans and easy parsing by programs.
- By combining XML with stylesheets formatted documents be created for display which never have to include any formatting information in the document itself.
- XML is based on a Unicode character set which enables it to be used with numerous languages.
- It an open standard not restricted to a predefined keyword set.

XML vs. HTML

XML is not HTML.

XML is **not** a replacement for HTML.

XML and HTML were designed with different goals:

- XML was designed to *describe* data and to focus on what data is.
- HTML was designed to *display* data and to focus on how data looks.

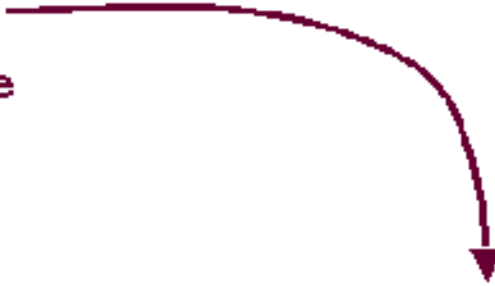
XML markup adds information to a document to make enhance its usefulness.

HTML is about displaying information, XML is about describing information.

To display XML encoded data a *stylesheet* can be used to turn tags into formatting instructions.

Example HTML

```
...  
<center>  
    ... this is HTML and here is<br>some  
    <bold>  
        bold text  
    </bold>  
    to illustrate  
</center>  
the point...
```



... this is HTML and here is
some **bold text** to illustrate
the point...

Example XML

```
<record>
  <date><d>24</d><m>10</m><y>2003</y></date>
  <person>A</person>
</record>
<record>
  <date><d>15</d><m>01</m><y>2002</y></date>
  <person>B</person>
</record>
```

?

3+4*5 In XML →

```
<add>
  <int>3</int>
  <mult>
    <int>4</int><int>5</int>
  </mult>
</add>
```

Example XML

XML is extensible. This means that new tags can be defined for specific domains or data.

```
<?xml version="1.0"?>
```

← Document Prolog

```
<course>
  <title>
    <modcode>CA437</modcode>
    :
    <modtitle>Multimedia Information Retrieval</modtitle>
  </title>
  <author>
    <firstname>Gareth</firstname>
    <surname>Jones</surname>
  </author>
  <year_runs>
    <year>2005</year>
    -
    <year>2006</year>
  </year_runs>
</course>
```


Contrasting HTML and XML

In HTML some elements do not have to have a closing tag.

The following code is legal in HTML:

```
<p>This is a paragraph
```

```
<p>This is another paragraph
```

In XML all elements must have a closing tag like this:

```
<p>This is a paragraph</p>
```

```
<p>This is another paragraph</p>
```

Contrasting HTML and XML

XML tags are case sensitive, HTML tags are not.

In XML the tag `<Letter>` is different from the tag `<letter>`.

Opening and closing tags must therefore be written with the same case:

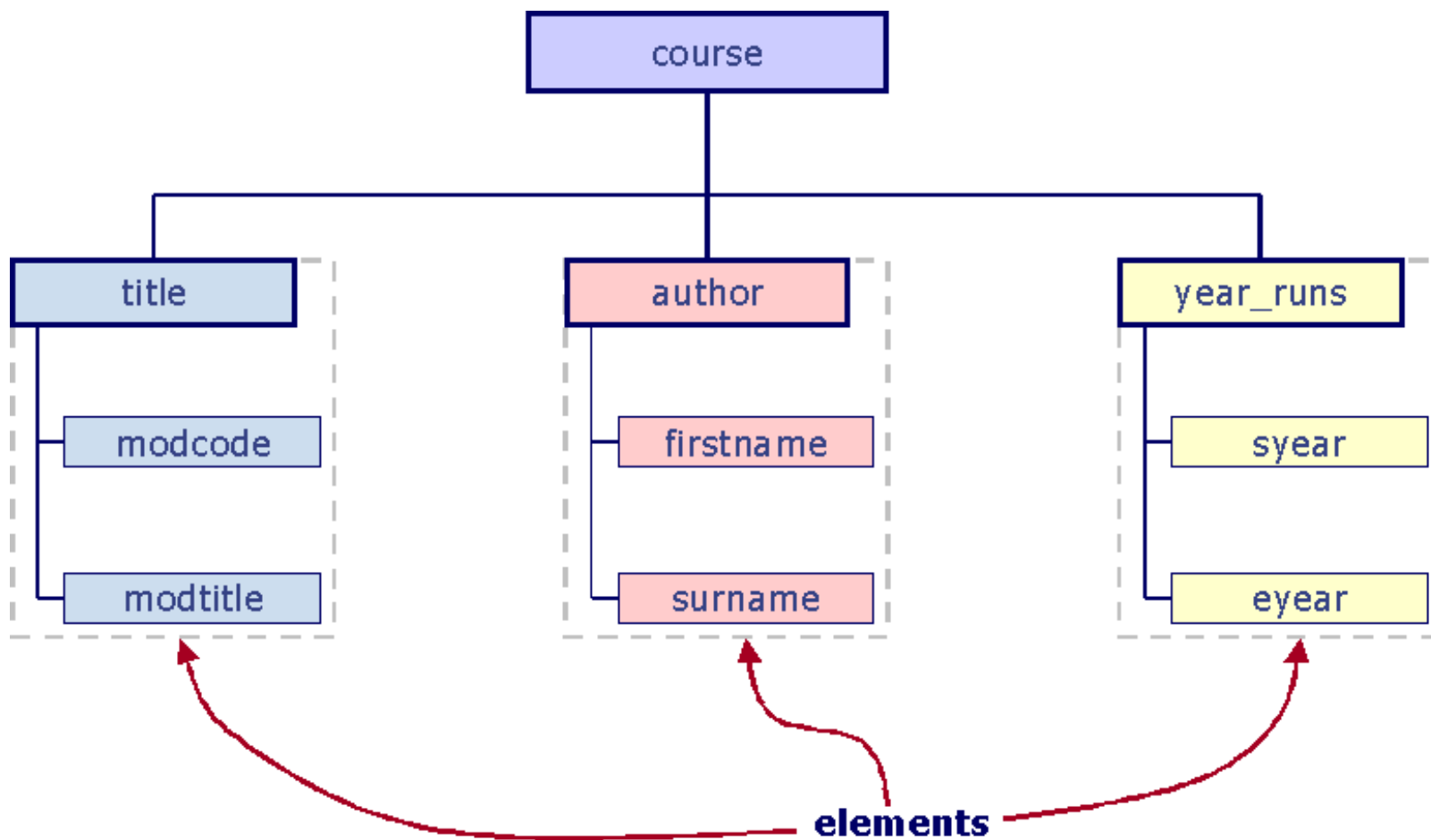
```
<Message>This is incorrect</message>
```

```
<message>This is correct</message>
```

Well Formed vs. Valid Document

- A “well formed” XML document (freeform XML) conforms to:
 - All elements must be properly nested.
 - All attributes must be quoted.
 - All elements with empty content must be self identifying.
(`<p>Name: <input name="name" /> </p>`)
 - Case sensitivity.
 - No use of characters with reserved meanings.
- A “valid” XML document (document modelling) conforms to:
 - Rules as above.
 - A valid DTD. A document instance is compared to a document model in a process called *validation*.

The XML Doc



Course DTD

```
<!-- course.dtd -->
<!DOCTYPE course
[
    <!ELEMENT course (title,author,year_runs)>
    <!ELEMENT title (modcode,modtitle)>
    <!ELEMENT modcode (#PCDATA)>
    <!ELEMENT modtitle (#PCDATA)>
    <!ELEMENT author (firstname,surname)>
    <!ELEMENT firstname (#PCDATA)>
    <!ELEMENT surname (#PCDATA)>
    <!ELEMENT year_runs (syear, #PCDATA ,eyear)>
    <!ELEMENT syear (#PCDATA)>
    <!ELEMENT eyear (#PCDATA)>
]>
```

Displaying XML

In HTML tags have default formatting interpretations.

In XML tags have no defined formatting characteristics. In order to display it, separate rules must be used to interpret XML marked up information.

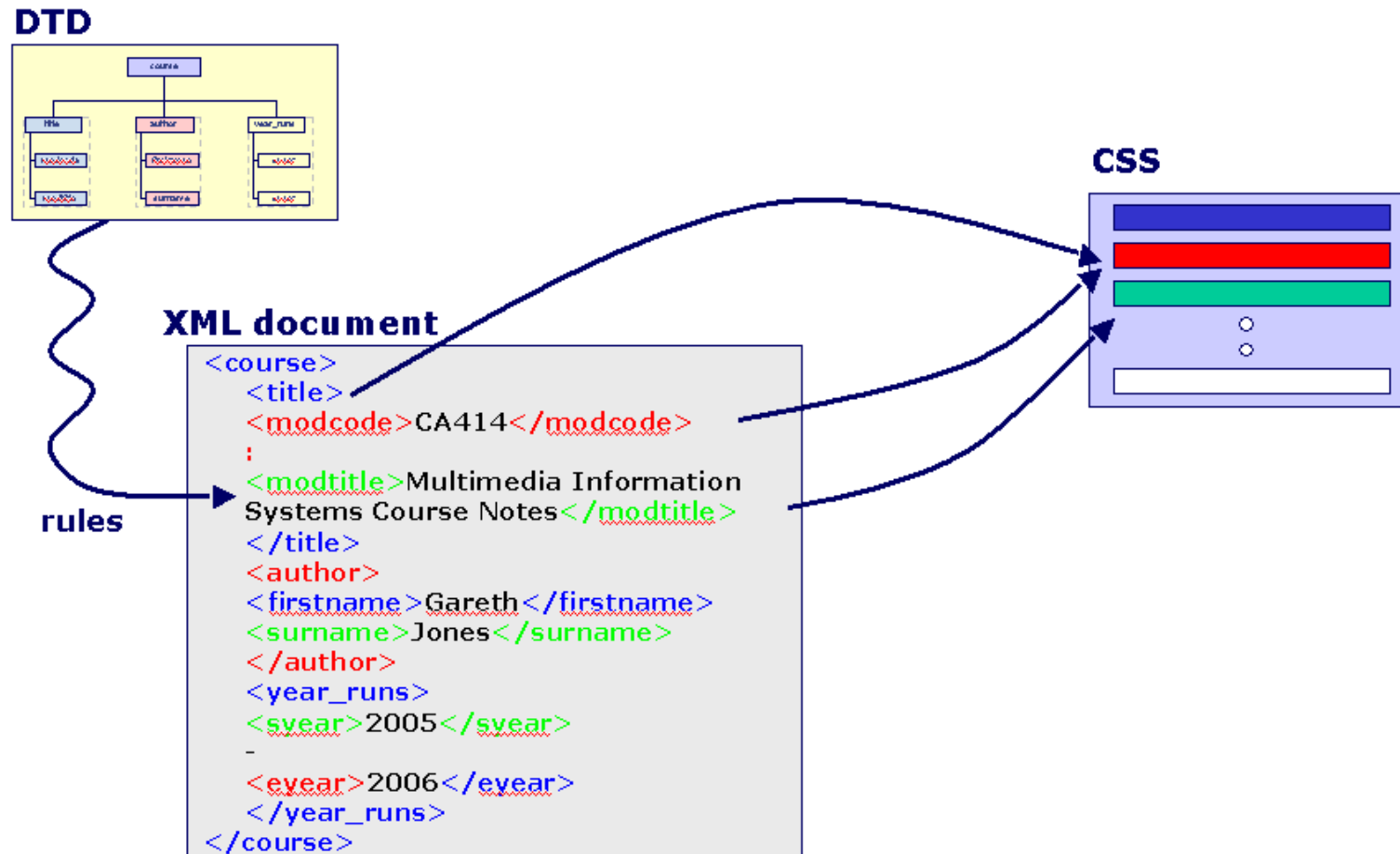
Two popular ways to do this are: CSS (Cascading Style Sheets) and XSLT.

Cascading style sheets add formatting information in layers which overrule previous layers.

- The display rules are defined in a style sheet.

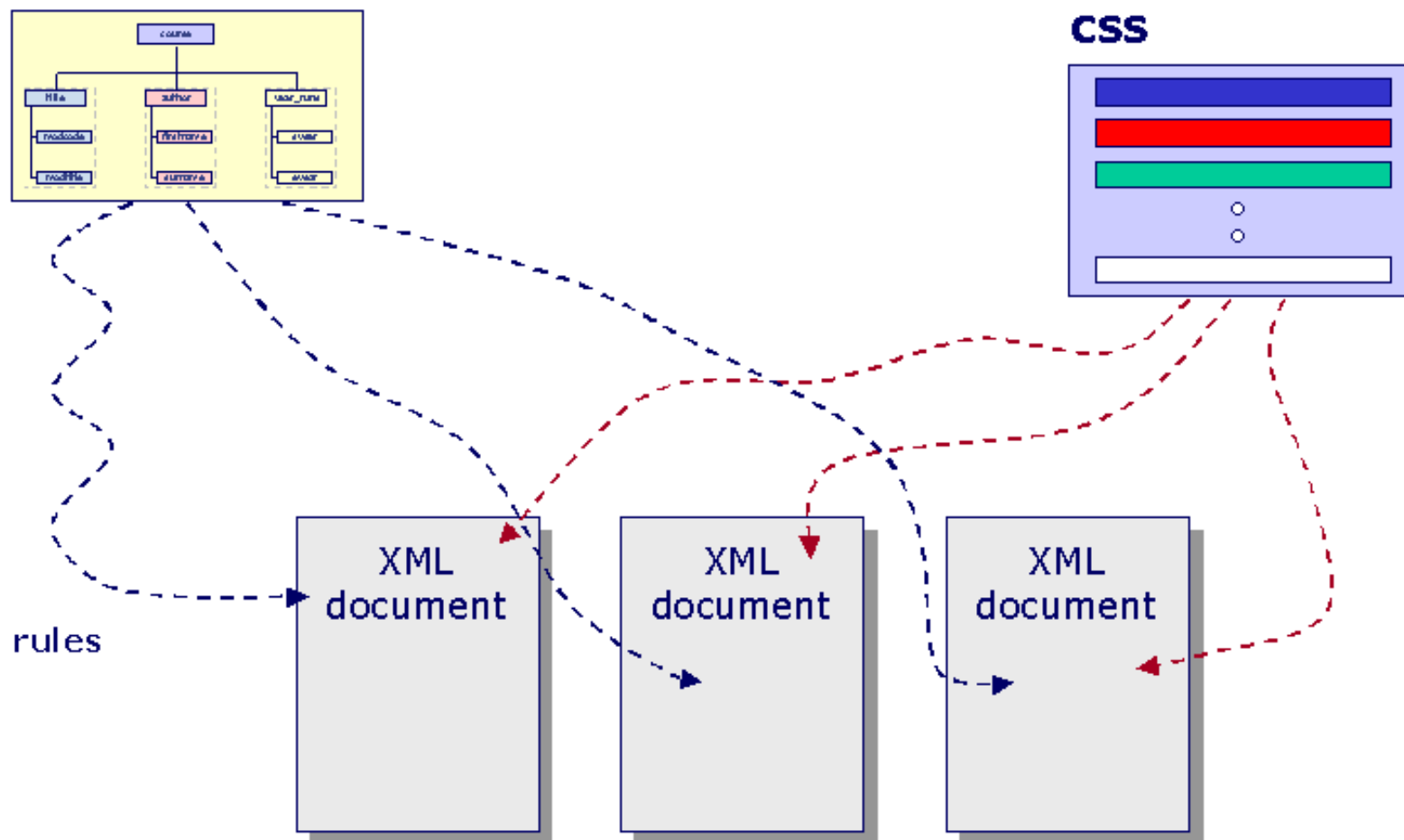
Cascading style sheets can be applied to HTML or directly to XML.

Displaying XML



Displaying XML

Many documents, one DTD and Stylesheet.



Using a CSS

```
<?xml version="1.0"?>
<?xml=stylesheet href="notes.css" type = "text/css" ?>
<course>
  <title>
    <modcode>CA414</modcode>
    :
    <modtitle>Multimedia Information Systems Course Notes</modtitle>
  </title>
  <author>
    <firstname>Gareth</firstname>
    <surname>Jones</surname>
  </author>
  <year_runs>
    <year>2005</year>
    -
    <year>2006</year>
  </year_runs>
</course>
```

What is in a CSS?

```
course      {font-family: Arial; font-size: 12pt;}
title       {font-style: italic;}
modcode     {color: #009900;}
modtitle    {color: #0099FF;}
author      {font-style: italic; font-weight: bold;}
firstname   {color: #009900;}
lastname    {color: #0099FF;}
...
year_runs   {font-weight: bold;}
syear       {color: #009900; text_align: right;}
eyear       {color: #0099FF; text_align: right;}
```

XML in Search

XML DTDs can be defined for objects to entered into a search engine.

For example, a photo or other image, an XML markup scheme can be defined which captures the attributes of the image.

- time of capture: day, month, year, minutes, hours, seconds, ...
- which can be interpreted into: day of week (Monday,...), month (January, ...), season (autumn, ...), weekend, .., etc.
- GPS location: mapped to named locations via gazzateer.
- image quality and other features
- automatic content analysis (colours, shapes, named places, named individuals, etc.); manual labels for the same thing

XML in Search

Structured image metadata of this type can be used in various ways by a search engine:

- Search against individual elements, e.g. search for images taken only by a certain make or model of camera, or by using the description of the images entered when they were uploaded to an archive.
- Combine all metadata into a single unstructured file, index like a standard free text document, and then search using a standard information retrieval system.