INFO/CS 4302
Web Information Systems

FT 2012

Week 4: Structured Data and Document Presentation Formats

(Lecture 6)

Theresa Velden

Lecture Plan

Tuesday

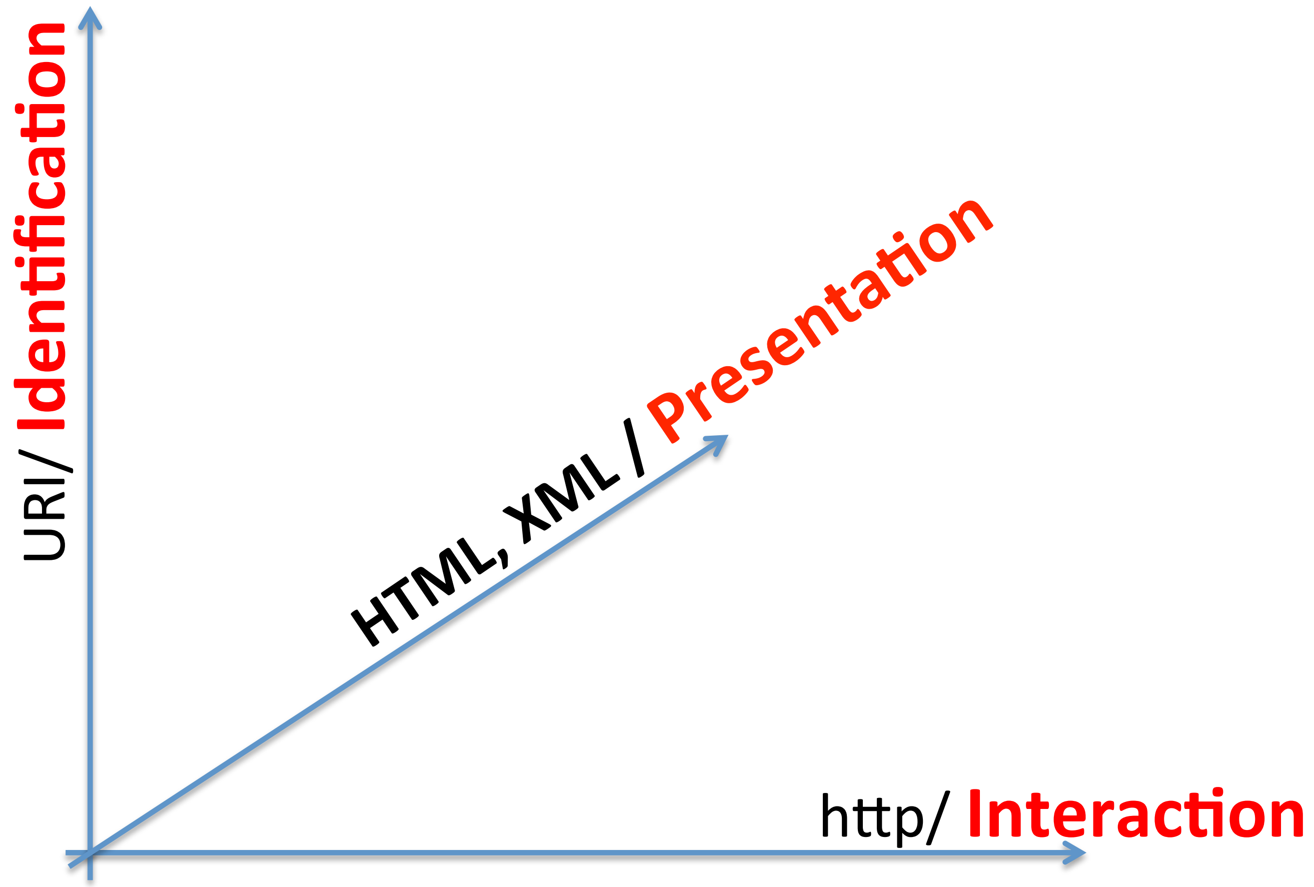
- Recap & Homework 2 Review
- BREAK [Team Formation]
- Mark-up Languages, HTML
- XML & Namespaces
- XML-DOM

Thursday Lecture

- XML Schema & RelaxNG
- XPath
- Demonstrations

RECAP

- Three Architectural Components
- Principle of Orthogonal Specification



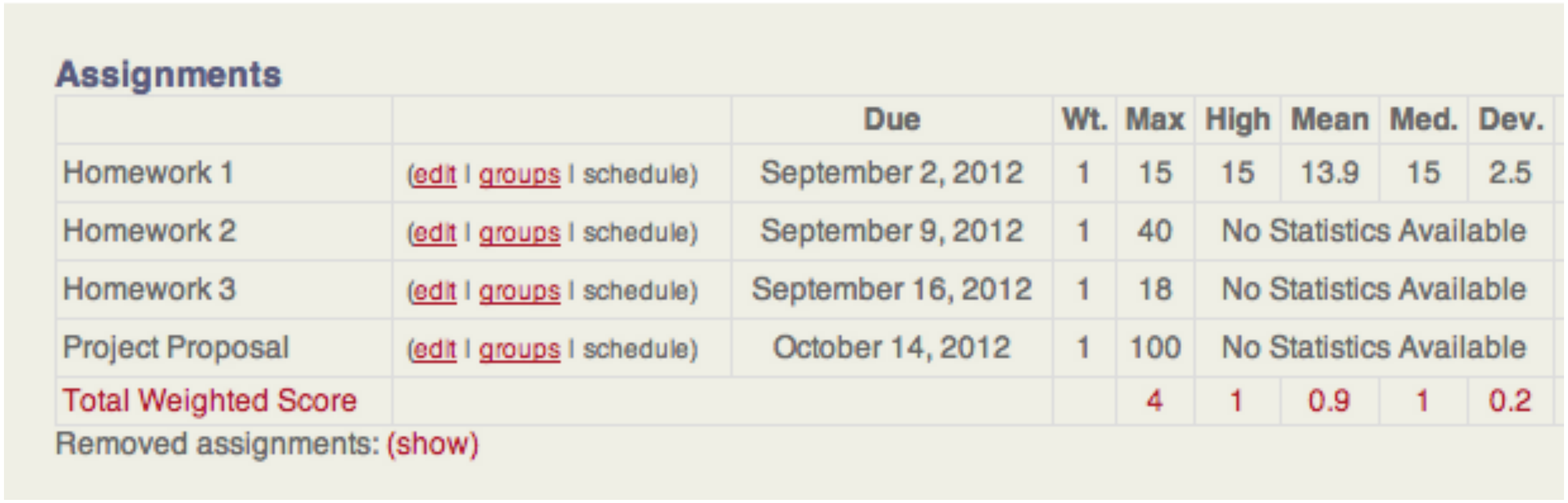
General Homework Comments

- Read questions very carefully
- Go back over lecture slides
- Don’t rely on last minute answers on piazza or by email
  - Please direct questions about the content of the course or ambiguity of homework questions to piazza so the entire course staff (instructors and TAs) can chime in - don’t send those questions by email to the instructors
- From the course website: “Individual assignments are meant to be worked on alone.”
  - Fishing for correct homework answers on piazza is not fair nor is collectively constructing the answers to specific homework questions

New Homework Schedule

- Homework about a week's topic to be released Thursday the same week
- Advantages:
  - You are familiar with the material when you start on homework
  - Better chance to follow-up unclear points during class and in office hours before homework is due
- Starting with release of hw 4 & submission of hw 3:
  - Homework assignments get released **Thursday night** (after both lectures on the respective material have been held)
    - This means hw 4 will be released 9/20
  - **Homework assignments are are due on Thursday night 11:59PM**
    - This means hw3 will be due on 9/20. This gives you extra 4 days for hw3
- Exception possible for **reading-based assignments**: sometimes more reasonable to assign ahead of a week so you are prepared to build on the readings in class discussions

Homework 1 Stats



Assignments

		Due	Wt.	Max	High	Mean	Med.	Dev.
Homework 1	(edit | groups | schedule)	September 2, 2012	1	15	15	13.9	15	2.5
Homework 2	(edit | groups | schedule)	September 9, 2012	1	40	No Statistics Available			
Homework 3	(edit | groups | schedule)	September 16, 2012	1	18	No Statistics Available			
Project Proposal	(edit | groups | schedule)	October 14, 2012	1	100	No Statistics Available			
Total Weighted Score				4	1	0.9	1	0.2

Removed assignments: (show)

Homework 2

Task 1: Identifiers (DOI, URI, DNS)

	DOI	URI	DNS
Persistence	high, cannot be reassigned; granted by IDF	low, for http URI e.g. up to domain name authority and maintainer of web server	medium, domain names can be reassigned
Scope	"a specific intellectual property entity (object), which may or may not be an Internet-accessible file" [http://www.doi.org/doi_handbook/3_Resolution.html]	"A Uniform Resource Identifier (URI) is a compact sequence of characters that identifies an abstract or physical resource." [http://tools.ietf.org/html/rfc3986]	any social entity exercising ownership over a name space to map domain names to IP addresses
Uniqueness	unique identification	unique identification (refers to only one resource)	unique identification
Governance	International Standard, ISO 26324; Managed by International DOI Foundation (IDF)	IANA scheme registry; delegation downwards to URI scheme specification and possibly further down to subordinate registries	IANA and domain name registries (also: network information center, NIC) maintain mapping tables
Actionability	yes, "Digital Identifier = network actionable identifier ("click on it and do something")" http://www.doi.org/factsheets/DOIKeyFacts.html	not inherent; accessibility cannot be assumed, only identification [RFC 3986]	yes, DNS look-up

Homework 2

Task 2: HTTP in your web browser

Dereference http://www.infosci.cornell.edu/Courses/info4302/2012fa/ and answer the following questions.

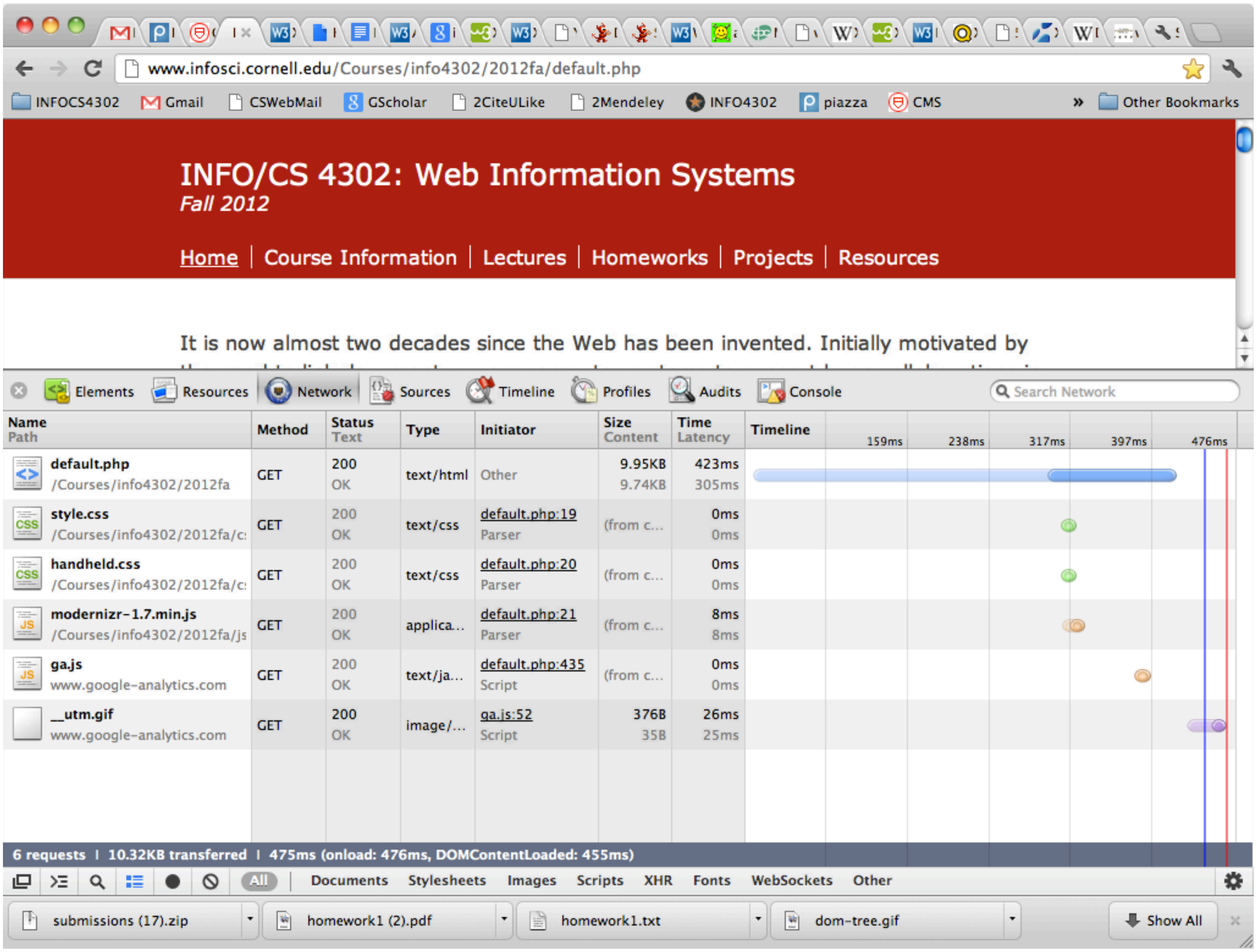
- How many web resources were requested and returned by this single HTTP request?
- Describe the sequence of events triggered by this request, how many resources were eventually requested, and what is the nature (content-type) of each resource representation?
- What is the meaning of the status code returned for each resource?
- When you hit your browser's back button and reload the page, what has changed in the HTTP transactions and why? How does this relate to the cache that you cleared at the beginning of this exercise?

Homework 2

Task 2: HTTP in your web browser

Dereference http://www.infosci.cornell.edu/Courses/info4302/2012fa/ and answer the following questions.

- How many web resources were requested and returned by this single HTTP request?



www.infosci.cornell.edu/Courses/info4302/2012fa/default.php

INFOCS4302 | Gmail | CSWebMail | GScholar | 2CiteULike | 2Mendeley | INFO4302 | piazza | CMS | » | Other Bookmarks

INFO/CS 4302: Web Information Systems
Fall 2012

Home | Course Information | Lectures | Homeworks | Projects | Resources

It is now almost two decades since the Web has been invented. Initially motivated by

Elements | Resources | Network | Sources | Timeline | Profiles | Audits | Console | Search Network

Name / Path	Method	Status / Text	Type	Initiator	Size / Content	Time / Latency	Timeline
default.php /Courses/info4302/2012fa	GET	200 OK	text/html	Other	9.95KB 9.74KB	423ms 305ms	
style.css /Courses/info4302/2012fa/c	GET	200 OK	text/css	default.php:19 Parser	(from c...	0ms 0ms	
handheld.css /Courses/info4302/2012fa/c	GET	200 OK	text/css	default.php:20 Parser	(from c...	0ms 0ms	
modernizr-1.7.min.js /Courses/info4302/2012fa/js	GET	200 OK	applica...	default.php:21 Parser	(from c...	8ms 8ms	
ga.js www.google-analytics.com	GET	200 OK	text/ja...	default.php:435 Script	(from c...	0ms 0ms	
__utm.gif www.google-analytics.com	GET	200 OK	image/...	ga.js:52 Script	376B 35B	26ms 25ms	

159ms 238ms 317ms 397ms 476ms

6 requests | 10.32KB transferred | 475ms (onload: 476ms, DOMContentLoaded: 455ms)

All | Documents | Stylesheets | Images | Scripts | XHR | Fonts | WebSockets | Other

submissions (17).zip | homework1 (2).pdf | homework1.txt | dom-tree.gif | Show All

Homework 2

Task 2: HTTP in your web browser

Dereference http://www.infosci.cornell.edu/Courses/info4302/2012fa/ and answer the following questions.

- How many web resources were requested and returned by this single HTTP request?

- **Answer:** One resource was requested, and no resource was returned. Instead one representation of content type html was returned.

Homework 2

Task 2: HTTP in your web browser

Dereference http://www.infosci.cornell.edu/Courses/info4302/2012fa/ and answer the following questions.

- Describe the sequence of events triggered by this request, how many resources were eventually requested, and what is the nature (content-type) of each resource representation?

- **Answer:** One resource was requested, one representation was returned of content type text/html and was parsed by the browser triggering requests for 5 more resources. Those had the content types
  - Text/css
  - Appplication/x-javascript
  - Text/javascript
  - Image/gif

Homework 2

Task 2: HTTP in your web browser

Dereference http://www.infosci.cornell.edu/Courses/info4302/2012fa/ and answer the following questions.

- What is the meaning of the status code returned for each resource?

- **Answer:** 200 OK is returned for all six resource get requests and means "The request has succeeded. "

Homework 2

Task 2: HTTP in your web browser

Dereference http://www.infosci.cornell.edu/Courses/info4302/2012fa/ and answer the following questions.

- When you hit your browser's back button and reload the page, what has changed in the HTTP transactions and why? How does this relate to the cache that you cleared at the beginning of this exercise?
- **Answer:** The representations of 4 resources were cached (locally stored by the browser); the http request was a conditional get request for the resource and determined that the resource has not been modified since last requested (Status code 304). Hence the local representation file was reused when loading to save time

Homework 2

Task 3: HTTP with cURL

Use curl to experiment with the following HTTP GET scenarios:

- Scenario 1: www.google.com
- Scenario 2: http://dbpedia.org/resource/Berlin
- Scenario 3: URI doi:10.1021/ci050378m

For each scenario report the following characteristics:

- the number of resources involved in the HTTP transaction.
- the number of representations and their associations with the resource.
- the role of content negotiation in the relationship between resources and representations.
- the role of redirection in the relationship between resources and representations.

Homework 2

Task 3: HTTP with cURL

- Scenario 1: access http://www.google.com to retrieve its versions in french and spanish

Answers:

```
curl -v --head --header "Accept-Language: fr" http://www.google.com
curl -v --head --header "Accept-Language: es" http://www.google.com
```

1. One resource is involved (URI=http://www.google.com)
2. Two representations of the same resource, one in French and one in Spanish.
3. Language content negotiation was involved, enacted by a http GET request that specified a preference for a specific language
4. No re-direction occurred.

Homework 2

Task 3: HTTP with cURL

Scenario 3: access to content/representation for URI doi:10.1021/ci050378m through the proxy URI http://dx.doi.org/10.1021/ci050378m (note this will only work at Cornell due to licensing restrictions). Think carefully when you answer the following question. What does each of the resources (and their respective URIs) involved in accessing a representation denote (make sure to consider the DOI, the proxy, and the final URI)?

Answers:

curl http://dx.doi.org/10.1021/ci050378m [→ HTTP/1.1 303 See Other]

curl http://pubs.acs.org/doi/abs/10.1021/ci050378m

1. In the http transaction 2 resources are involved: the resource record of the DOI at the proxy URI, and the article at the publisher website
2. Two representations: a representation of the resource record of content type text/html, and the article abstract page of content type text/html
3. Content negotiation is not involved in any obvious form, unless the HTML representations returned were the result of a server-side content negotiation determining the likely most suitable format (e.g. Based on browser version and operating system)
4. Redirection is used to make a representation of the object identified by the DOI accessible.

Homework 2

Task 3: HTTP with cURL

- Scenario 2: access to http://dbpedia.org/resource/Berlin to retrieve its versions in text/html and application/rdf+xml. Describe what the resource identified as http://dbpedia/resource/Berlin denotes. What is the "object of interest" (using the terminology of the web architecture document) that it stands for?

Answers:

```
curl -H "Accept: text/html" http://dbpedia.org/resource/Berlin
curl -I --head -H "Accept: application/rdf+xml" http://dbpedia.org/resource/Berlin
```

1. Three resources are involved, one abstract (referring to the city of Berlin), and two informational ones, one a page about Berlin with the URI http://dbpedia.org/page/Berlin, and one with structured data about Berlin with the URI http://dbpedia.org/data/Berlin
2. Two representations were returned that represent the state of the original resource 'Berlin' as well as of the data and page resource respectively.
3. The GET requests used the Accept field to express format preferences in the content negotiation with the web server. These preferences were respected and representations of the respective content types returned.
4. Redirection was used to refer to an appropriate representation of the requested resource.

Abstract and Informational Resources

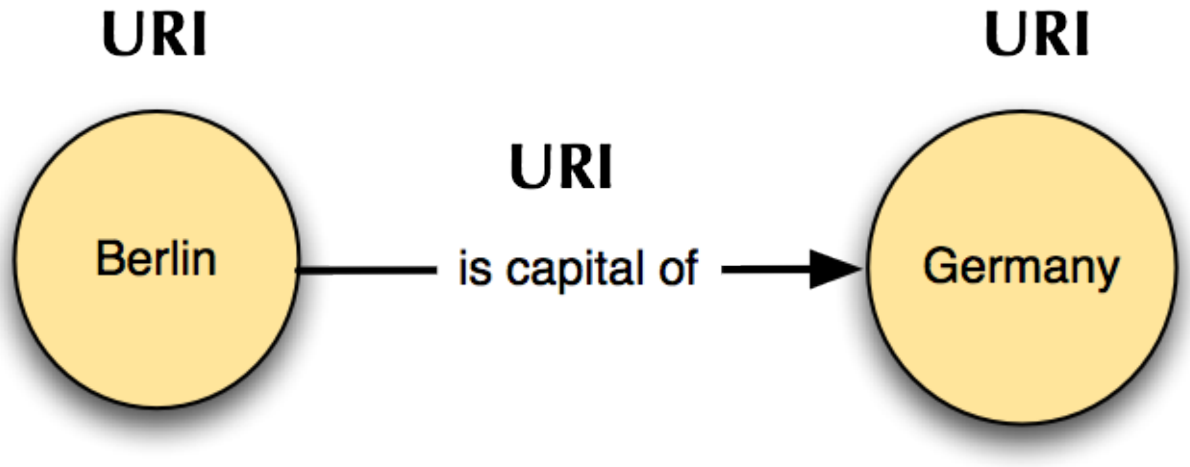
"A **resource**

- is an entity that can be identified by a URI
- is an abstract concept: we cannot see, smell, touch, examine a resource
- is not necessarily retrievable through the internet

- Abstract resources: their essence is not information
- Informational resources: their essential characteristics can be conveyed in a message → can be serialized into a bitstream

Why abstract Resources?

- “Semantic Web” or “Web of Data”
- RDF (Resource Description Framework):
  - Triple: subject, predicate, object
  - Structure and link data that describes things in the world
  - E.g.



How to incorporate URIs pointing to abstract resources has caused heated debate in the Technical Architecture Group (TAG) and was been eventually resolved by a best practice recommendation

Further Readings:

- Tim Berner’s Lee (2005) „What HTTP URIs identify”

  http://www.w3.org/DesignIssues/HTTP-URI2.html
- Bizer, C. and Heath, T. and Berners-Lee, T. (2009) Linked data-the story so far. International Journal on Semantic Web and Information Systems (IJSWIS) 5(3)

[httpRange-14] Resolved

From: Roy T. Fielding <fielding@gbiv.com>
Date: Sat, 18 Jun 2005 21:25:42 -0700
Message-Id: <3fc8037bc096da8c801ebc8c1295e09b@gbiv.com>
To: W3C TAG <www-tag@w3.org>

As everyone here knows, the TAG has spent a great deal of time discussing the httpRange-14 issue, as described at

http://www.w3.org/2001/tag/issues.html#httpRange-14

I am happy to report that we came up with a reasonable compromise solution at the recent TAG f2f meeting at MIT.

[httpRange-14] Resolved

<TAG type="RESOLVED">

That we provide advice to the community that they may mint "http" URIs for any resource provided that they follow this simple rule for the sake of removing ambiguity:

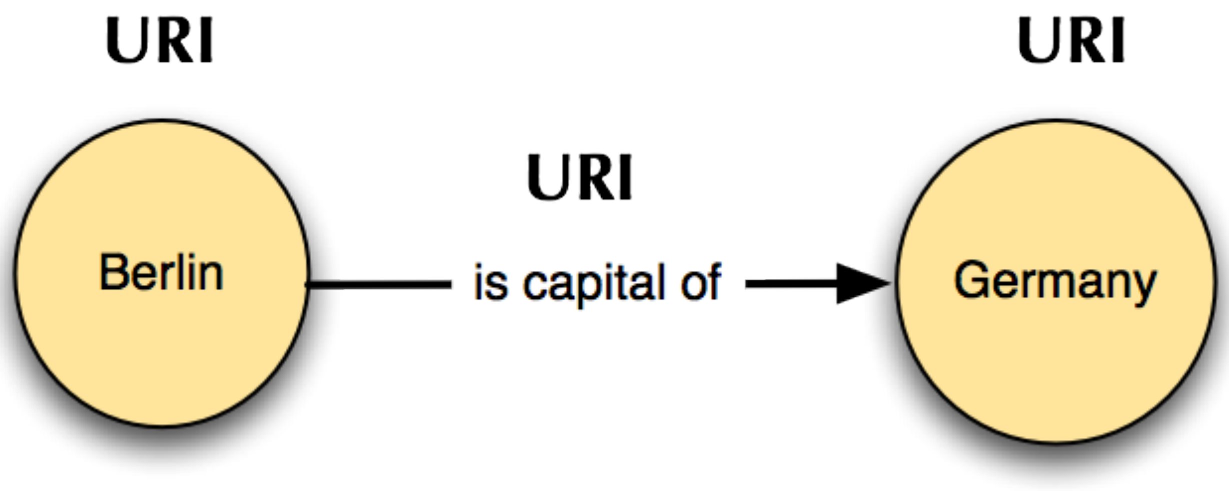
a) If an "http" resource responds to a GET request with a 2xx response, then the resource identified by that URI is an information resource;

b) If an "http" resource responds to a GET request with a 303 (See Other) response, then the resource identified by that URI could be any resource;

c) If an "http" resource responds to a GET request with a 4xx (error) response, then the nature of the resource is unknown.

</TAG>

How link description to an abstract resource?



Current Best Practice: Return on http GET request ‘303 See Also” Code and URI of description document in Location header field

Homework 2

Task 3: HTTP with cURL

- ‘For the xml+rdf request this link was given: <http://mementoarchive.lanl.gov/dbpedia/timegate/http://dbpedia.org/resource/Berlin>; rel="timegate”’

Homework 2

Task 3: HTTP with cURL

```
$ curl -v --header "Accept: application/rdf+xml" http://dbpedia.org/resource/Berlin

> GET /resource/Berlin HTTP/1.1
> User-Agent: curl/7.19.7 (universal-apple-darwin10.0) libcurl/7.19.7 OpenSSL/0.9.8r zlib/1.2.3
> Host: dbpedia.org
> Accept: application/rdf+xml
>
< HTTP/1.1 303 See Other
< Date: Tue, 11 Sep 2012 00:28:38 GMT
< Content-Type: application/rdf+xml; qs=0.95
< Connection: keep-alive
< Server: Virtuoso/06.04.3132 (Linux) x86_64-generic-linux-glibc25-64  VDB
< Accept-Ranges: bytes
< TCN: choice
< Vary: negotiate,accept
< Content-Location: /data/Berlin.xml
< Link: <http://mementoarchive.lanl.gov/dbpedia/timegate/http://dbpedia.org/resource/
Berlin>; rel="timegate"
< Location: http://dbpedia.org/data/Berlin.xml
< Content-Length: 0
```

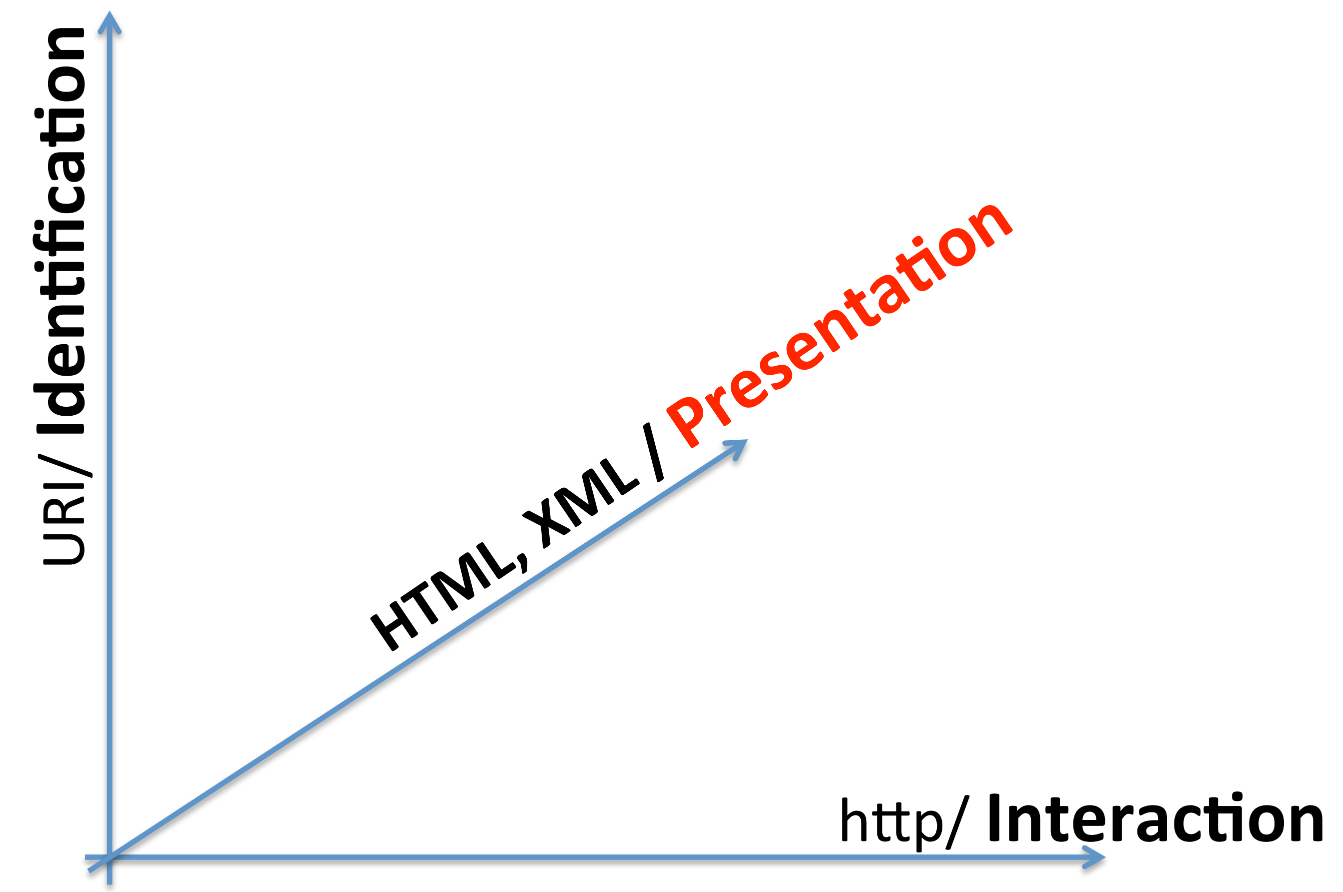
Persistence of Web Resources

Herbert Van de Sompel

MEMENTO PROJECT

- Using a new variant of content negotiation to retrieve older versions of the representations of a web resource
  - **Date time negotiation**
- HTTP-based Memento framework
  - HTTP MEMO Working Draft https://datatracker.ietf.org/doc/draft-vandesompel-memento/
- Interlink current resources with resources that encapsulate their past
- Facilitates obtaining representations of prior states of a resource (held in web archives or versioning systems of content management systems)
- Awarded the Digital Preservation Award 2010 by The Institute for Conservation and the Digital Preservation Coalition (DPC)

Three Architectural Components



BREAK (& TEAM FORMATION)

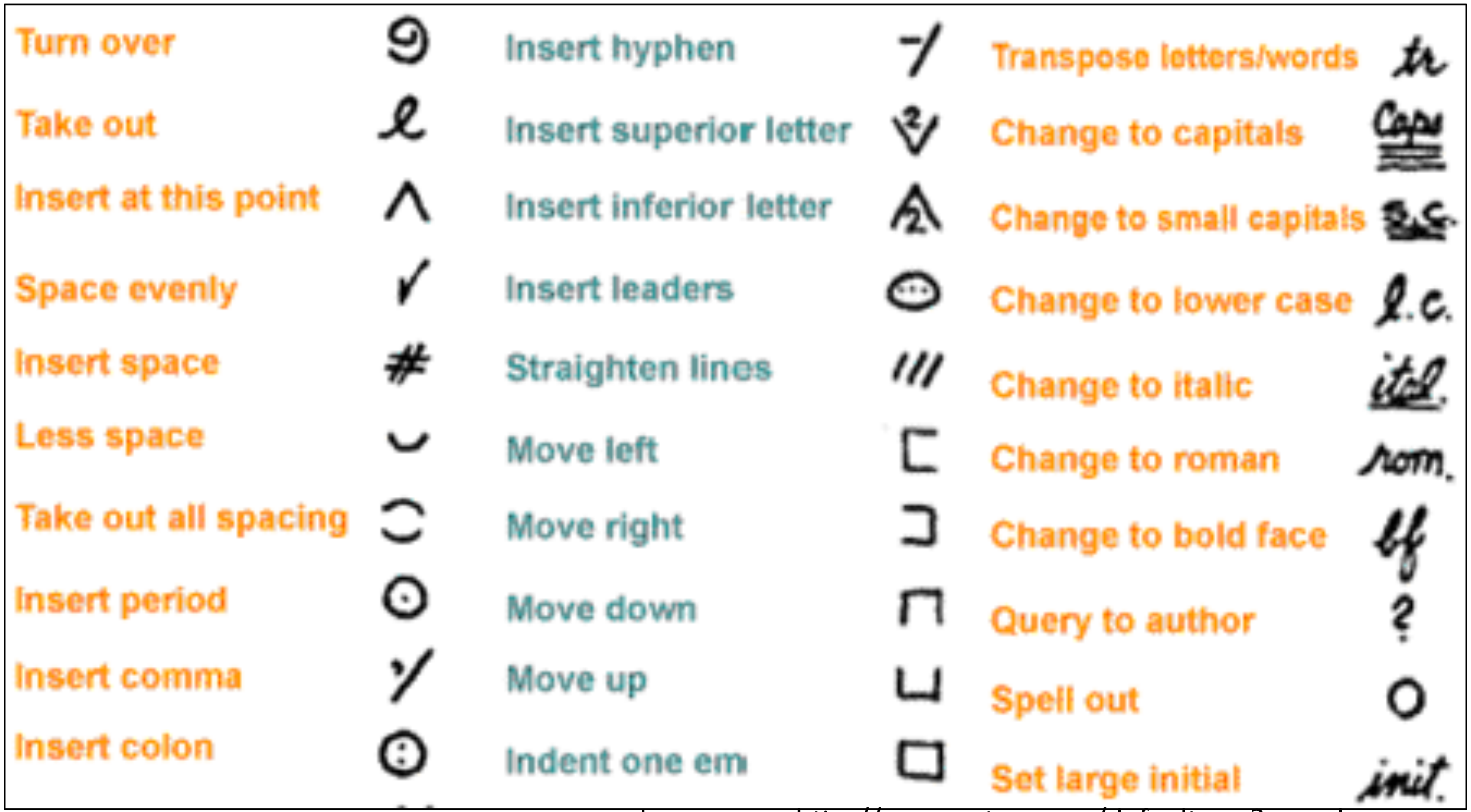
PRESENTATION

Presentation of Web Content

- Recent trend: from the document web to a web of data
- from structured representations of documents to structured representations of data
    - in human readable and machine readable form
- Document Mark-up Languages
    - text plus metadata about the text
    - basic Idea: to separate structure (and format) from content of a text

Origin of Markup Languages

Editor ‘mark-up’: instructions to a typesetter about formatting of a text



Turn over	9	Insert hyphen	-/	Transpose letters/words	tr
Take out	ℓ	Insert superior letter	∨	Change to capitals	Caps
Insert at this point	∧	Insert inferior letter	∧	Change to small capitals	S.C.
Space evenly	√	Insert leaders	⊙	Change to lower case	l.c.
Insert space	#	Straighten lines	///	Change to italic	ital
Less space	◡	Move left	[	Change to roman	rom.
Take out all spacing	⁐	Move right	]	Change to bold face	bf
Insert period	⊙	Move down	⊓	Query to author	?
Insert comma	,/	Move up	⊔	Spell out	○
Insert colon	⊙	Indent one em	□	Set large initial	init.

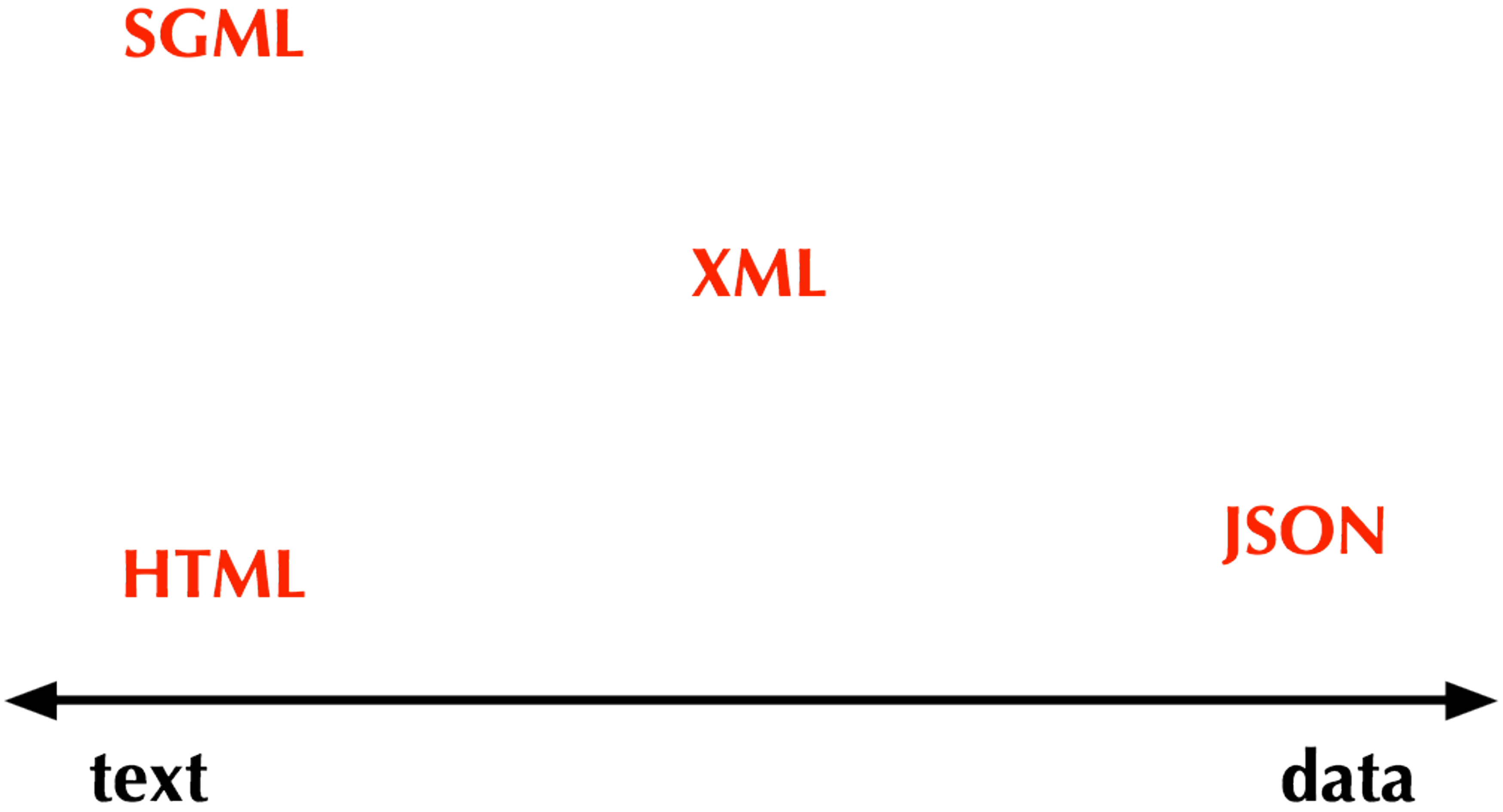
Img source: http://www.prt.wa.gov/default.asp?p=rc_how_proof

Mark-Up Languages

- **SGML** (Standard Generalized Markup Language; 1986 - approved as ISO international standard 8879)
  - Widely used: Defense, Aerospace, Semiconductor and Publishing industries
  - Very powerful and broad; lack of stable tool support
  - 'Sounds Good Maybe Later'
- **HTML** ('Killer-App' of the Web)
  - Invented by Tim Berners Lee
  - HTML IETF in 1994, 1995 HTML 2.0 was published as IETF RFC 1866
  - Fixed vocabulary '(tag set')
- **XML**
  - development started in 1996 under auspices of W3C World Wide Web consortium
  - subset of SGML suitable for delivery of content over the web
- **JSON / YAML**
  - Data serialization language (not document centric)
  - Lessons learned from XML

See: Digital History: A Guide to Gathering, Preserving, and Presenting the Past on the Web http://chnm.gmu.edu/digitalhistory/

Characteristics



HTML

Hypertext Mark-up Language

- Core web technology, derived from SGML but much, much simpler
- Simple, fixed tag set
- Introduces anchor tag for hyperlinks
- Robust since tolerant
  - <p>paragraph1 <p>paragraph2

  same as: <p>paragraph1</p> <p>paragraph2</p>
- Based on 7-bit ascii
- Expresses structure and formatting information
  - <title>Structure Information</title>
  - <b>Formatting information</b>

HTML Limitations

- Limited, fixed tag set
  - How encode domain specific content (Chemistry, Math,…)?
- Mixes structure and formatting

XML

Extensible Mark-up Language

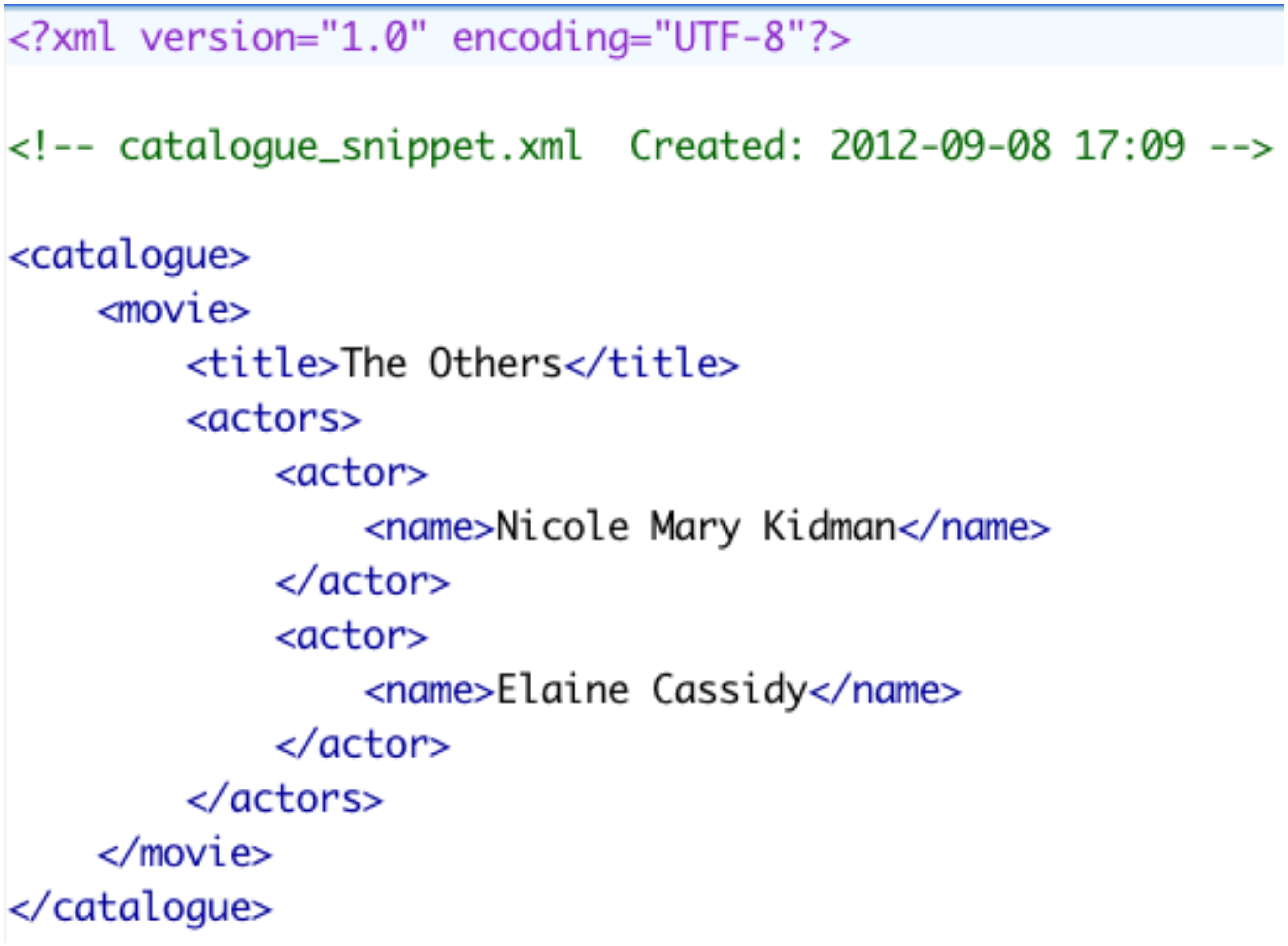
- Subset of SGML for improved ease of implementation
- Meta-Language: allows defining mark-up languages
  - No pre-defined tag set
  - Purpose specific tags and document model is defined by a DTD or schema document
- Unicode character set
- W3C Recommendation (1998)

XML Suite of Standards

- **XML** Syntax (e.g. ‘closed tags: <para></para>)
  - **‘well-formed’** XML: syntactically correct
- **XML Namespaces**
  - global semantic partitions of tag semantics (elements and attributes)
- **XML Schema**
  - Specifies allowed elements, their attributes, frequency, parent-child relationships etc.
  - **‘valid’** XML: ‘semantically correct’ = conforms to a schema
- **Xpath**
  - Addressing specific information items in an XML document
  - XPath 2.0 became a Recommendation on 23 January 2007.
- **XSLT**
  - language for transformation of XML documents
  - E.g. as a style-sheet: XML + XSLT → HTML for human consumption

- XQuery – generalized query language for xml base databases
- Xpointer – syntax for stating address information in a link to an xml document
- Xlink – specifying behaviors, types and semantics of links

XML Example: Nested Elements

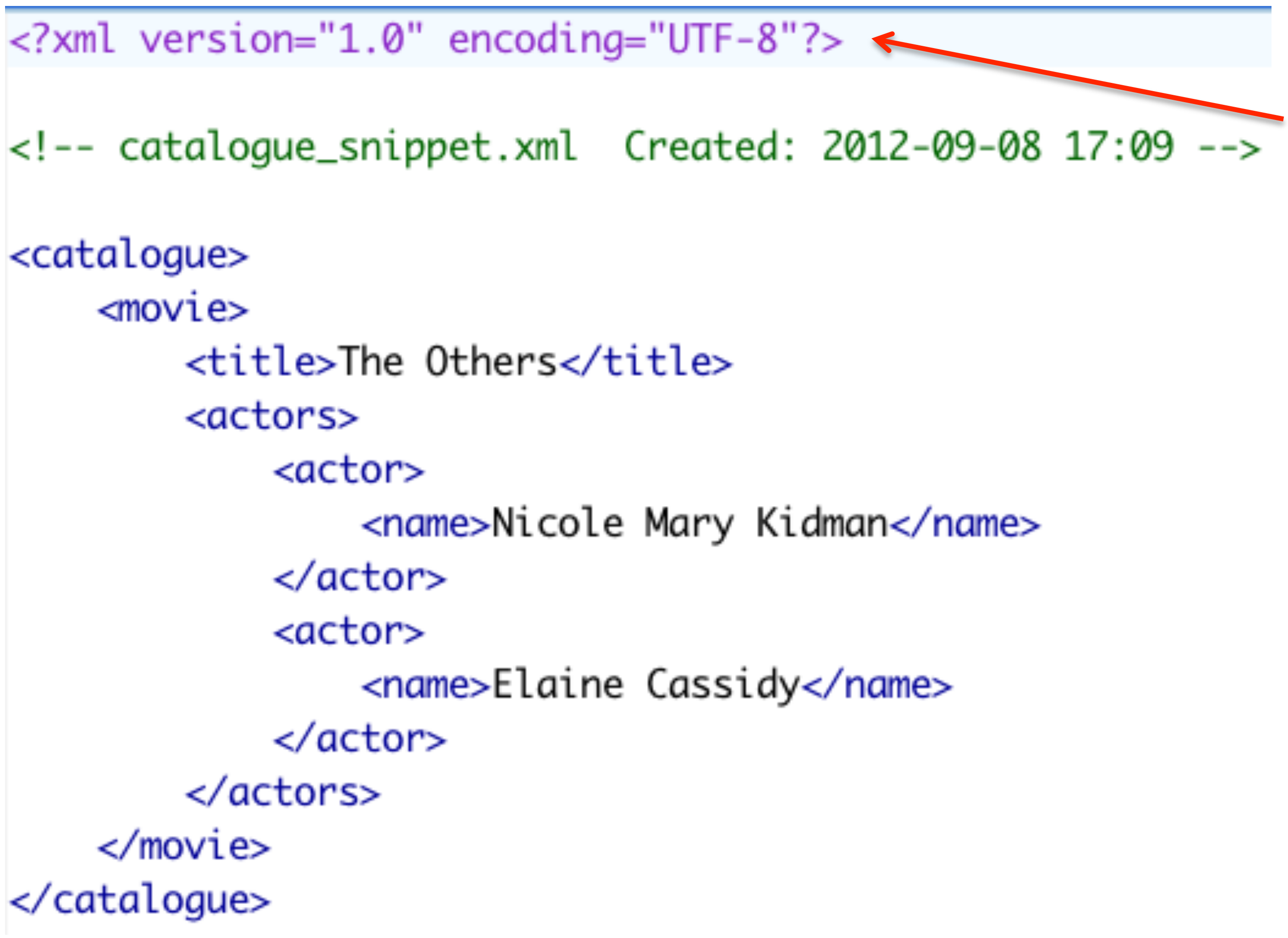


```xml
<?xml version="1.0" encoding="UTF-8"?>

<!-- catalogue_snippet.xml  Created: 2012-09-08 17:09 -->

<catalogue>
    <movie>
        <title>The Others</title>
        <actors>
            <actor>
                <name>Nicole Mary Kidman</name>
            </actor>
            <actor>
                <name>Elaine Cassidy</name>
            </actor>
        </actors>
    </movie>
</catalogue>
```

XML Example: Nested Elements



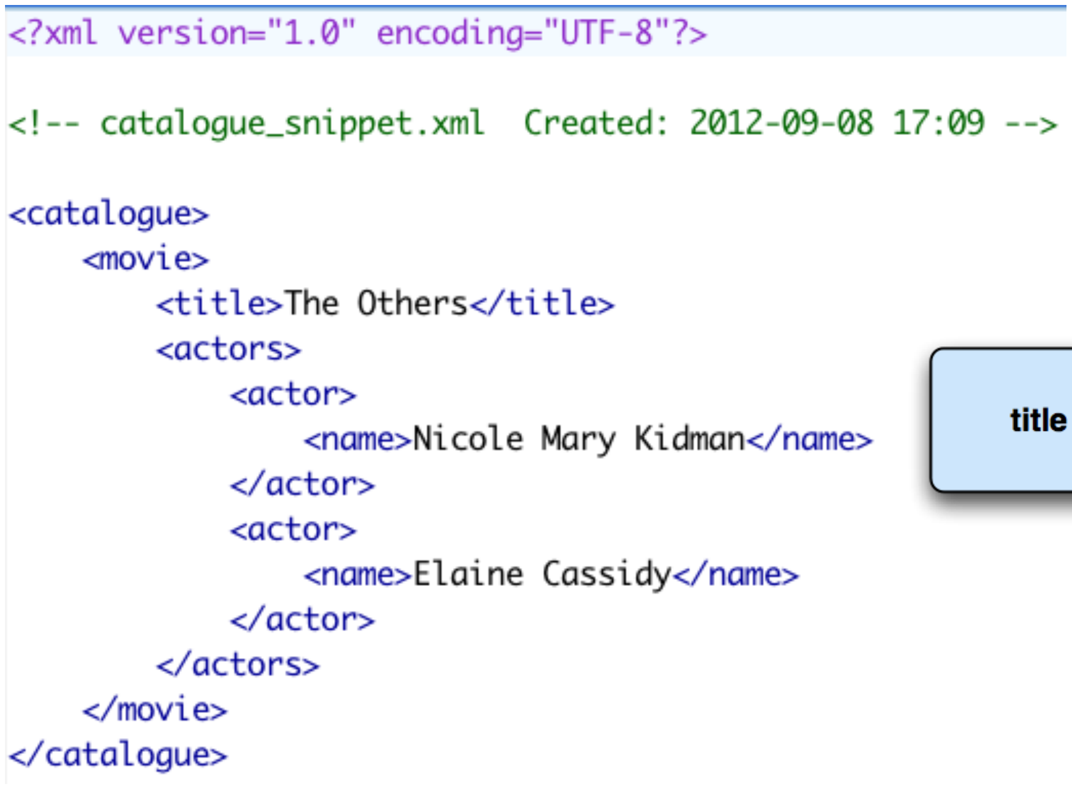
```xml
<?xml version="1.0" encoding="UTF-8"?>

<!-- catalogue_snippet.xml  Created: 2012-09-08 17:09 -->

<catalogue>
    <movie>
        <title>The Others</title>
        <actors>
            <actor>
                <name>Nicole Mary Kidman</name>
            </actor>
            <actor>
                <name>Elaine Cassidy</name>
            </actor>
        </actors>
    </movie>
</catalogue>
```

Optional XML declaration (version of xml specification, encoding)

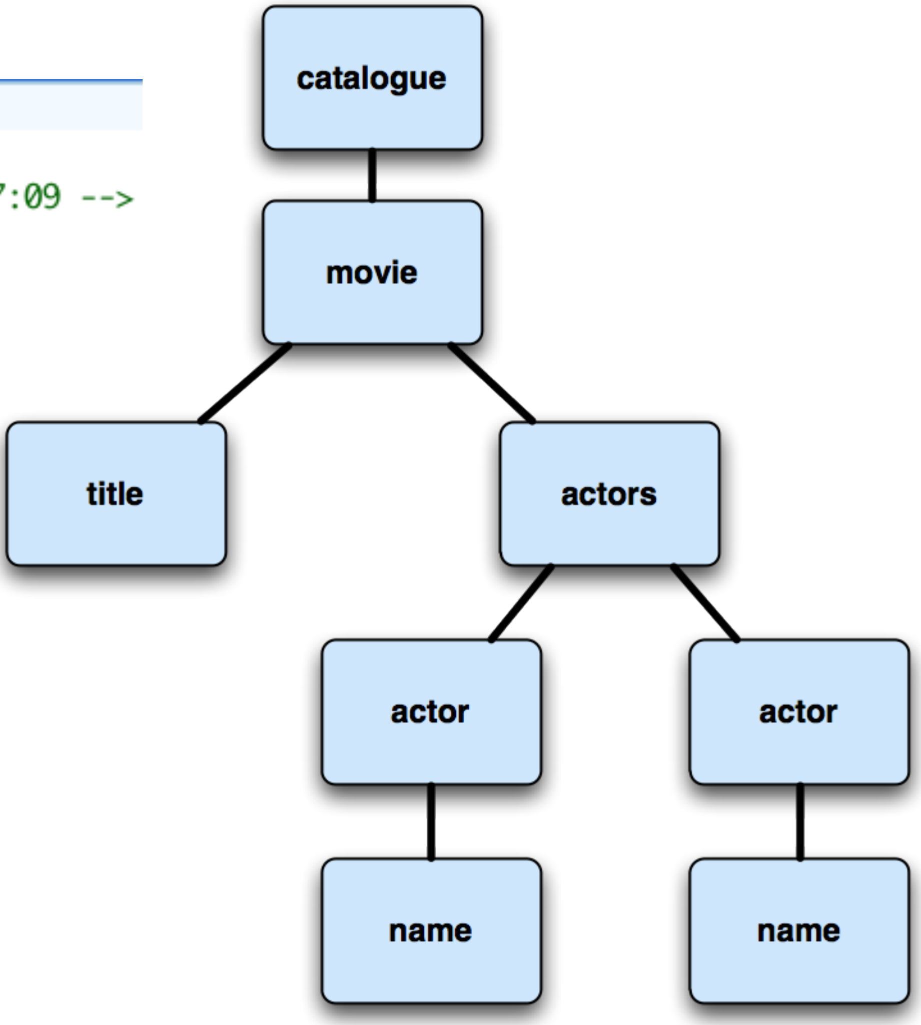
The XML Tree



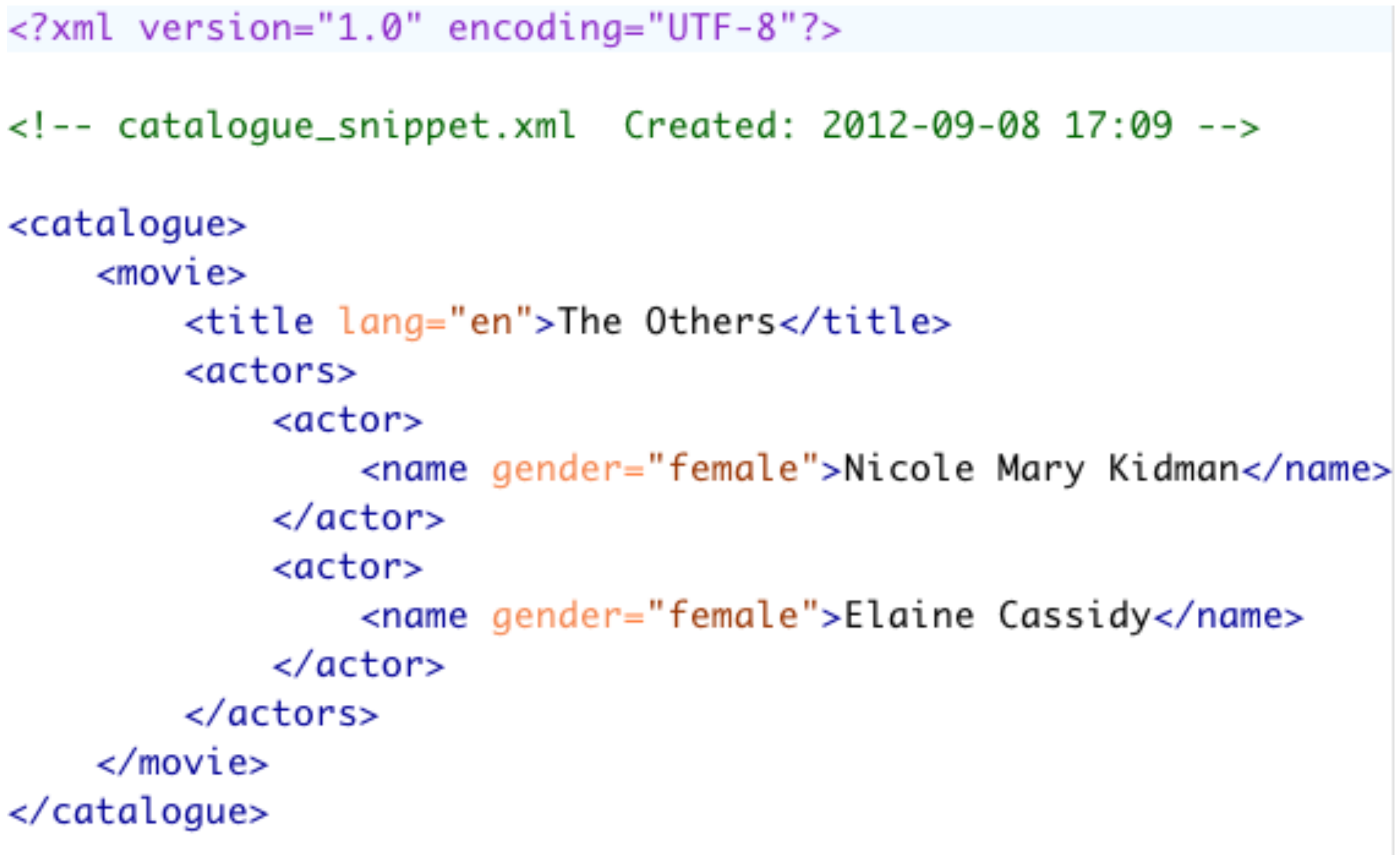
```xml
<?xml version="1.0" encoding="UTF-8"?>

<!-- catalogue_snippet.xml  Created: 2012-09-08 17:09 -->

<catalogue>
    <movie>
        <title>The Others</title>
        <actors>
            <actor>
                <name>Nicole Mary Kidman</name>
            </actor>
            <actor>
                <name>Elaine Cassidy</name>
            </actor>
        </actors>
    </movie>
</catalogue>
```



XML Example 2: Element Attributes

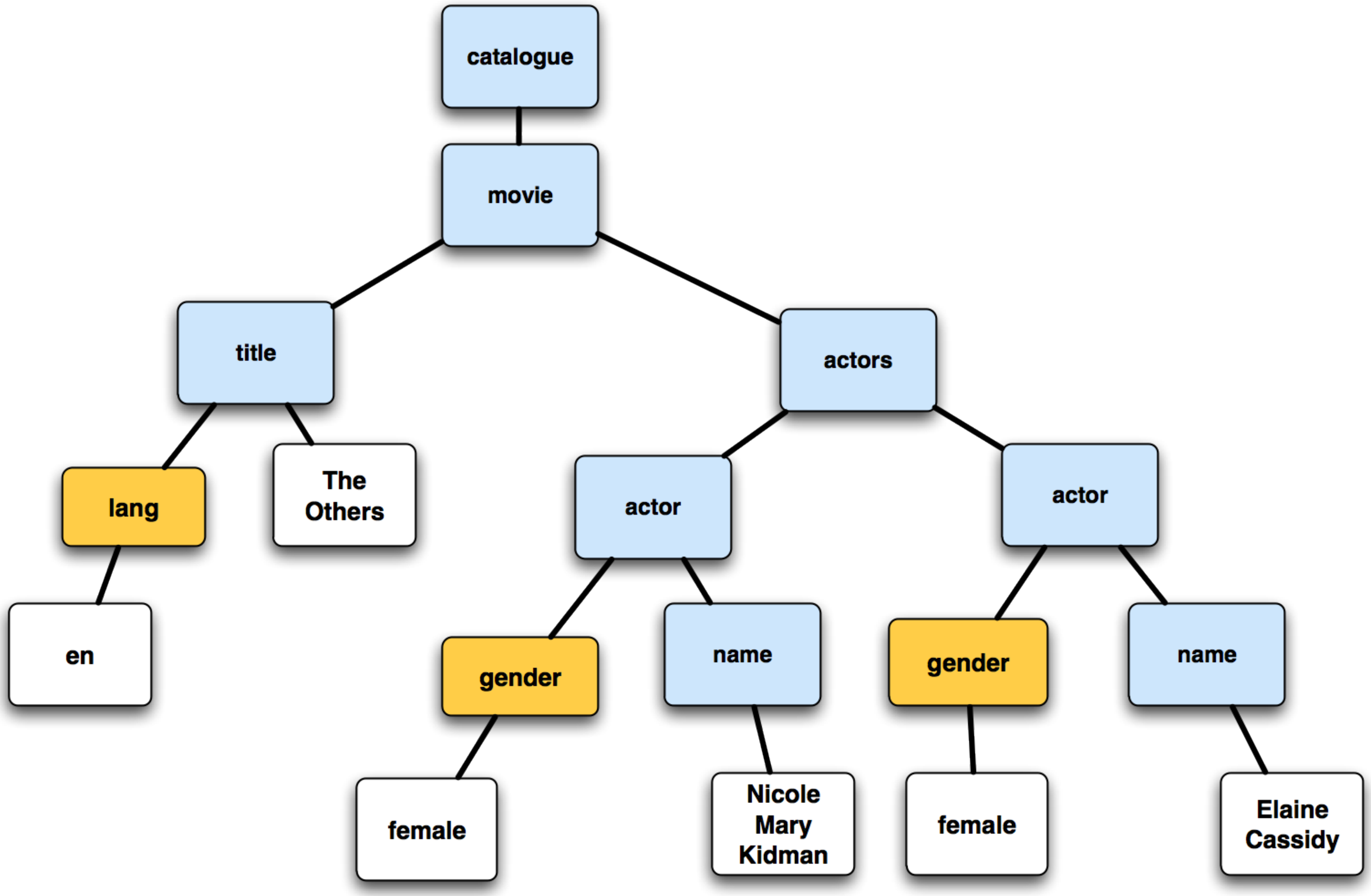


```xml
<?xml version="1.0" encoding="UTF-8"?>

<!-- catalogue_snippet.xml  Created: 2012-09-08 17:09 -->

<catalogue>
    <movie>
        <title lang="en">The Others</title>
        <actors>
            <actor>
                <name gender="female">Nicole Mary Kidman</name>
            </actor>
            <actor>
                <name gender="female">Elaine Cassidy</name>
            </actor>
        </actors>
    </movie>
</catalogue>
```

The XML Tree


catalogue
movie
title
actors
lang
The Others
actor
actor
en
gender
name
gender
name
female
Nicole Mary Kidman
female
Elaine Cassidy

XML: Element Attributes

- Attributes: name-value pairs that can be assigned to elements
- Attribute specifications must be made within start tag of an element
- When to use elements, when attributes to represent information?
  - Up to the designer; consider:
    - An element can only have one attribute with the same name
    - An attribute cannot be further structured
    - Attributes suitable for most identifiers and references, eg. id, href…

Well-formed XML Documents

- An xml document is well-formed there exists a single, unique tree structure to represent the document

Building Blocks of a well-formed XML document

- One or more **elements**
  - Empty element (terminal node in a tree)

    <movie/> or <movie></movie>
  - Non-empty element
    - Simple (CDATA) value = only one child, a ‘text node’

      <actor>Nicole Mary Kidman</actor>
    - Complex value = root of an arbitrary sub-tree

      <actor><name>Nicole Mary Kidman</name><movie>The Others</movie></actor>
- Requires one single **root element**
- One or more **attributes** per element

  <title lang=“en”>The Others</title>

Building Blocks of a well-formed XML document

- Every xml document SHOULD have a declaration
- Every opening tag must have a closing tag
- Tags can not overlap (must be well-nested)
- XML documents can have only one root element
- Attribute values must be in quotation marks (single or double) and only one value per attribute.

Well-formed XML documents: syntactic requirements

- Comments and processing instructions must not appear within tags
- Reserved characters should be encoded, e.g. &lt; instead of <
- Elements must obey XML naming conventions – case sensitive, start with letter or underscore
- Whitespace is meaningful – no &nbsp;

Motivation: Interoperability

Vocabulary – Namespaces
Syntax – XML
Grammar – Ontologies (e.g. OWL)
Protocols – HTTP

Môsieur J. [version 7.0.1]'s photostream

Namespaces

- How the web works:
  - Individually created documents linked by ambiguous references
- How improve into making it a global database of knowledge?
  - Key: allow for distributed knowledge creation and lazy integration
- Problems:
  - Collisions (of how things are named)
  - Joins (how to link related content)
- Namespaces:
  - Build on URI notion
  - Uniquely qualify intra-document name collisions
  - Provide technology for cooperation

[Carl Lagoze, INFO 4302, Fall 2011]