Processing of structured documents

Spring 2003, Part 9
Helena Ahonen-Myka

In this part:

- XML linking language (XLink)
- Resource Description Framework (RDF)

XML Linking Language (XLink)

- W3C Recommendation June 2001
- Create and describe links between resources
- both basic unidirectional links and more complex linking structures
- XLink allows XML documents to:
  - assert linking relationships among more than two resources
  - associate metadata with a link
  - express links that reside in a location separate from the linked resources

Simple hyperlinks

- Like HTML <a>
- the hyperlink uses URIs as its locator technology
- the hyperlink is expressed at one of its two ends
- the hyperlink identifies the other end
- users can initiate traversal only from the end where the hyperlink is expressed to the other end
- the hyperlink's effect on windows, styles, etc. are determined by user agents (e.g. browsers)

XLink concepts

- Links and resources
- arcs, traversal, and behavior
- resources in relation to the physical location of a linking element

Links and resources

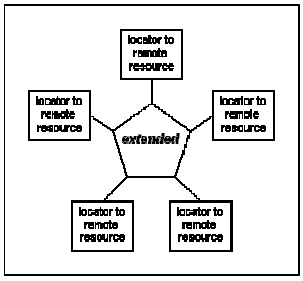
- An XLink **link** is an explicit relationship between resources or portions of resources
  - represented as an XML element (= "linking element")
- a **resource** is any addressable unit of information or service
  - files, images, documents, programs, query results

Links and resources

- A resource is addressed by a URI reference
  - it is possible to address a portion of a resource
  - e.g. if the whole resource is an XML document, a useful portion might be a particular element inside the document. Following a link to it might result, for example, in highlighting that element or scrolling to that point in the document
- when a link associates a set of resources, those resources are said to **participate** in the link
- XLink links can associate also non-XML resources

7

Link and resources



8

Arcs, traversal, and behavior

- Using or following a link for any purpose is called **traversal**
  - even though some links can associate arbitrary numbers of resources, traversal always involves a pair of resources (or portions of them)
- the source from which traversal is begun is the **starting resource** and the destination is the **ending resource**

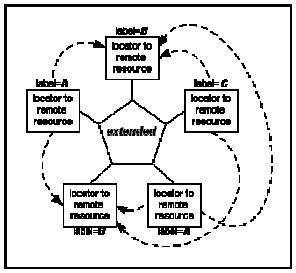
9

Arcs, traversal, and behavior

- Information about how to traverse a pair of resources, including the direction of traversal and possible application behavior information, is called an **arc**
- if two arcs in a link specify the same pair of resources, but they switch places as starting and ending resources, then the link is **multidirectional**

10

Link, resources, arcs



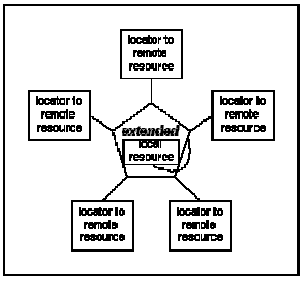
11

Resources in relation to the physical location

- A **local resource** is an XML element that participates in a link by virtue of having as its parent, or being itself, a linking element
- any resource that participates in a link by virtue of being addressed with a URI reference is considered a **remote resource**
  - even if in the same XML document as the link, or even in the linking element

12

Link (remote and local resources)



13

Resources in relation to the physical location

- An arc that has a local starting resource and a remote ending resource goes **outbound** (away from the linking element)
- if an arc's ending resource is local, but the starting resource remote, then the arc goes **inbound**
- if neither the starting nor the ending resource is local, then the arc is a **third-party** arc
- one link typically specifies only one kind of arc
  - -> outbound, inbound, third-party links

14

Resources in relation to the physical location

- To create a link that starts from a resource to which you do not have write access or which does not offer any way to embed linking constructs (e.g. multimedia content), it is necessary to use an inbound or third-party arc
- the requirements for discovery of an inbound or third-party link are greater than for outbound links
  - documents containing collections of inbound and third-party links are called link databases, or **linkbases**

15

XLink markup

- Namespace:
  - http://www.w3.org/1999/xlink
- XLink's namespace provides global attributes for use on elements that are in any arbitrary namespace
  - type, href, role, arcrole, title, show, actuate, label, from, and to
  - document creators use the XLink global attributes to make the elements in their own namespace recognizable as XLink elements

16

XLink markup

- **type** attribute
  - values: simple, extended, locator, arc, resource, or title
  - dictates the XLink-imposed constraints that such an element must follow
- allowed combinations of XLink global attributes depend on the value of type (required)
- in the following, an element with the type=xxx is referenced by 'xxx-type element'

17

Example

```
<my:crossReference
   xmlns:my="http://example.com/"
   xmlns:xlink="http://www.w3.org/1999/xlink"
   xlink:type="simple"
   xlink:href="students.xml"
   xlink:role="http://www.example.com/linkprops/studentlist"
   xlink:title="Student List"
   xlink:show="new"
   xlink:actuate="onRequest">
Current List of Students
</my:crossReference>
```

18

Two kinds of links

- Extended links
  - full XLink functionality
  - structure can be fairly complex: elements for pointing to remote resources, elements for containing local resources, elements for specifying arc traversal rules, elements for specifying human-readable resource and arc titles
- Simple links
  - shorthand syntax for a common kind of link: outbound link with exactly two participating resources
  - no internal structure

19

Extended links

- type attribute: extended
- an extended link is a link that associates an arbitrary number of resources
- the participating resources may be any combination of remote and local
- the only kind of link that is allowed to have inbound and third-party arcs
- external linking elements can be stored separately from the resources they associate (e.g. in entirely different documents)

20

Extended links

- An **extended**-type element may contain a mixture of the following elements (among other content and markup)
  - **locator**-type elements that address the remote resources participating in the link
  - **resource**-type elements that supply local resources
  - **arc**-type elements that provide traversal rules among the link's participating resources
  - **title**-type elements that provide human-readable labels for the link

21

Extended links

- An extended-type element may have the semantic attributes **role** and **title**
  - supply semantic information about the link as a whole
  - **role**: indicates a property that the entire link has (URI)
  - **title**: indicates a human-readable description of the entire link

22

Local resources

- **resource**-type elements: an entire subelement, together with all of its contents, makes up a local resource
- any content
- attributes: **role**, **title**, **label**
  - **label**: a way for an **arc**-type element to refer to this **resource**-type element in creating a traversal arc

```
<gpa xlink:type="resource"
     xlink:label="PatJonesGPA">3.5</gpa>
```

23

Remote resources

- **locator**-type elements
- a **locator**-type element may have any content
- must have the locator attribute: **href**
  - a URI reference that identifies a remote resource
- may have attributes: **role, title,** and **label**
  - **role**: a property that the resource has (in general)
  - **title**: a human-readable description of the resource
  - **label**: a way for an **arc**-type element to refer to this **locator**-type element in creating a traversal arc

24

Traversal rules

- **arc**-type elements
- an extended **l**ink may indicate rules for traversing among its participating resources by means of a series of optional **arc-**type elements
- attributes:
  - traversal attributes: **from** and **to**
  - semantic attributes: **arcrole** and **title**
  - behavior attributes: **show** and **actuate**

25

Arc attributes

- Traversal attributes
  - Define the desired traversal between pairs of resources that participate in the same link
  - resources are identified by their **label** attribute values
  - **from:** starting resources
  - **to:** ending resources
- Semantic attributes
  - the meaning of the arc's ending resource relative to its starting resource
  - arcrole: <start> "has" <the arcrole property> <end>

26

Example: link from a student to his advisor

```
<go
  xlink:from="student62"
  xlink:arcrole="http://www.example.com/linkprops/advisor"
  xlink:to="prof7"
  xlink:show="replace"
  xlink:actuate="onRequest"
  xlink:title="Dr. Jay Smith, advisor" />
```

27

Locating linkbases

- In the case of inbound and third-party links, the XLink application needs to be able to find somehow both the starting resource and the linking element
- XLink provides a way to instruct XLink applications to access potentially relevant linkbases

28

Locating linkbases

- in an arc-type element:
  arcrole="http://www.w3.org/1999/xlink/properties/linkbases"
- the handling of a **l**inkbase arc is much like the handling of a normal arc
  - except: traversal entails loading the **l**inkbase to extract its links for later use
  - keeping track of the starting resources
    - whenever a starting resource is loaded, the respective links can be presented

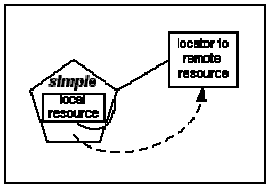
29

Simple links

- A simple link is a link that associates exactly two resources, one local and one remote, with an arc going from the former to the latter (= outbound link)
- a single simple **l**inking element combines the basic functions of an **extended**-type element, a **locator**-type element, an **arc**-type element, and a **resource**-type element

30

Simple links



31

Simple links

- <studentlink xlink:type="simple" xlink:href="...">Pat Jones</studentlink>
- the **simple**-type element itself, together with all of its content, is the local reference of the link (as if the element were a **resource**-type element)
- attributes: **type, href, role, arcrole, title, show, actuate**
  - **type**: simple
  - **href, role** and **title** as in a **locator**-type element
  - **show**, **actuate** and **arcrole** as in an arc-type element

32

Behavior attributes

- **show and actuate**
  - may be used on the **simple-** and **arc-**type elements
- **show**
  - used to communicate the desired presentation of the ending resource on traversal from the starting resource: **new, replace, embed, other, none**
- **actuate**
  - used to communicate the desired timing of traversal from the starting resource to the ending resource: **onLoad, onRequest, other, none**

33

Behavior attributes

- **show**
  - **new:** an application traversing to the ending resource should load it in a new window, frame, or other relevant presentation context
  - **replace:** an application should load the resource in the same window, frame, ... in which the starting resource was loaded
  - **embed:** an application should load the resource in place of the presentation of the starting resource

34

Behavior attributes

- **show**
  - **other:** the behavior of an application is unconstrained by the XLink specification
    - the application should look for other markup present in the link to determine the appropriate behavior
  - **none:** the behavior of an application is unconstrained by the XLink specification
    - no other markup is present to help the application determine the appropriate behavior

35

Behavior attributes

- **actuate**
  - **onLoad:** an application should traverse to the ending resource immediately on loading the starting resource
  - **onRequest:** an application should traverse from the starting resource to the ending resource only on a post-loading event triggered for the purpose of traversal
    - e.g. the user clicks on the presentation of the starting resource
  - **other** and **none**: like with **show**

36

Future of XLink?

- implementations are rare
- the emphasis of XML use (and users' interest) is elsewhere?
- the developers of XHTML takes XLink into consideration
  - ... but HTML <a> might be enough for most applications?

37

Resource Description Framework (RDF)

- Web was originally built for human consumption
  - although everything is machine-readable, the data is not (usually) machine-understandable
  - it is hard to automate anything on the web, but it is also not possible to manage the data manually
- Solution: metadata ("data about data")

38

RDF

- A framework that provides interoperability between applications
- emphasizes facilities to enable automated processing of Web resources

39

RDF: application areas

- resource discovery
- cataloging
- intelligent software agents
- content rating
- describing collections of pages
- describing intellectual property rights
- expressing privacy preferences
- profiles (of devices, connections, persons, services)

40

RDF

- W3C, November 2002:
  - RDF Concepts and Abstract Syntax
  - RDF Vocabulary Description Language 1.0: RDF Schema
  - RDF/XML Syntax Specification

41

Basic RDF model

- A model for representing named properties and property values
- three object types
  - resources
  - properties
  - statements

42

Basic RDF model

- Resources
  - all things being described are called resources
  - may be a web page, a part of a web page, a collection of pages, an object that is not directly accessible via the web (e.g. a printed book)
  - resources are always named by URIs
    - anything can have a URI

43

Basic RDF model

- Properties
  - a property is a specific aspect, characteristic, attribute, or relation used to describe a resource
  - each property has a specific meaning, defines its permitted values, the types of resources it can describe, and its relationship with other properties

44

Basic RDF model

- Statements
  - a specific resource + a named property + the value of that property for that resource is an RDF statement
  - three parts: subject, predicate, object
  - the object can be another resource or a literal

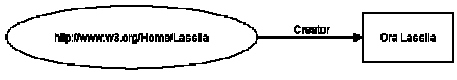
45

Examples

- "Ora Lassila is the creator of the resource http://www.w3.org/Home/Lassila."
  - = "http://www.w3.org/Home/Lassila has a creator whose value is Ora Lassila"
- This sentence has the following parts:
  - subject (resource): http://www.w3.org/...
  - predicate (property): Creator
  - object (literal): "Ora Lassila"

46

Graph



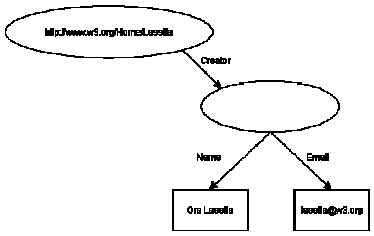
47

Example

- "The individual whose name is Ora Lassila, email <lassila@w3.org>, is the creator of http://www.w3.org/Home/Lassila".
- Now the creator is represented as another resource
  - the resource is anonymous

48

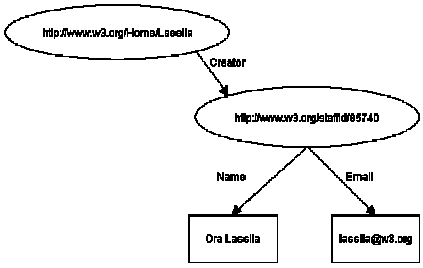
Graph



Example

- "The individual referred to by employee id 85740 is named Ora Lassila and has the email address lassila@w3.org. The resource http://www.w3.org/Home/Lassila was created by this individual."
- Assume the URIs that serve as unique keys of employees might be like
  - http://www.w3.org/staffId/85740

Graph



Basic XML syntax

- The RDF data model provides an abstract, conceptual framework for defining and using metadata
- a concrete syntax is also needed
  - creating and exchanging metadata
- XML can be used for encoding RDF

Example, full serialization syntax

```
<?xml version="1.0"?>
<rdf:RDF  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
          xmlns:s="http://example.org/schema">
    <rdf:Description rdf:about="http://www.w3.org/Home/Lassila">
        <s:Creator>Ora Lassila</s:Creator>
    </rdf:Description>
</rdf:RDF>
```


Example, full serialization syntax

```
<rdf:RDF  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
          xmlns:s="http://example.org/schema"
          xmlns:v="http://person.org/schema" >
  <rdf:Description about="http://www.w3.org/Home/Lassila">
    <s:Creator>
      <rdf:Description about="http://www.w3.org/staffId/85740">
        <v:Name>"Ora Lassila"</v:Name>
        <v:Email>"lassila@w3.org"</v:Email>
      </rdf:Description>
    </s:Creator>
  </rdf:Description>
</rdf:RDF>
```

Example: abbreviated syntax

```
<rdf:RDF  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
          xmlns:s="http://example.org/schema"
          xmlns:v="http://person.org/schema" >
  <rdf:Description about="http://www.w3.org/Home/Lassila">
    <s:Creator rdf:resource ="http://www.w3.org/staffId/85740"
      v:Name="Ora Lassila"
      v:Email="lassila@w3.org" />
  </rdf:Description>
</rdf:RDF>
```

55

RDF Schemas

- The RDF model does not define the vocabulary for properties, constraints of resources, etc.
- resource description communities require the ability to say certain things about certain kinds of resources
  - for describing bibliographic resources: "author", "title", "subject",...

56

RDF Schemas

- The declaration of these properties and their corresponding semantics are defined as an RDF Schema
- the RDF schema specification language provides a basic type system
  - it defines properties and resources such as rdfs:Class and rdfs:subClassOf that are used in specifying application-specific schemas
  - RDF schemas are also RDF documents

57

Semantic web

- Semantics of resources (data, services,...) are described using RDF (or something like that)
- new information can be inferred using the semantic metadata (reasoning)
- -> the new generation of Web

58