Mapping XLink to HTML in EAD version 2002 (article in progress)

by Mark Carlson, Special Collections Computer Support Analyst, University of Washington

Unidirectional links : the HTML hyperlink and <img> tags

We are all familiar with what XLink terms unidirectional or "outbound" links. These are the typical <a href="...">click here</a> hyperlinks that we are used to seeing on the web today. These are unidirectional links from the source document to the remote resource. In the case of the <img> tag in HTML, this is also a unidirectional link, except that the remote resource is embedded in the source document when the page loads. Outbound links such as these are termed "simple-type links" in XLink. They involve two (or three as we will see later on) resources, a local resource (such as the text you click on to initiate the link) and the remote resource (the value of the "href" attribute). As an illustration, note the following example:

<p>To view a copy of the Declaration of Independance, click <a href="document.html" target="_blank">here</a>.</p>

This would generate this sort of display in a typical web browser:

To view a copy of the Declaration of Independance, click here.

In an EAD document, to generate the same type of link, you would use either one of the appropriate "simple-type" linking elements from the chart below, or, as per the XLink specification, you could use an "extended-type" link to create simple-type link functionality (XLink 1.0 section 5.2).

Simple	Comparable element(s) in HTML
<extptr> <dao>	<img> <embed> <object> <applet>
<extref> <bibref> <archref> <title>	<a href="link.html">click here</a>
<ptr>	<a href="#target" />
<ref>	<a name="target" />

In an EAD document, this could be represented in this way:

<p>To view a copy of the Declaration of Independance, click <extref href="document.html" role="text/html" show="new" actuate="onrequest">here</extref>.</p>
If you choose to use an "extended-type" linking element such as <linkgrp> or <daogrp>, you must, per the XLink specification, have at least two resources (see XLink 1.0 section 2.2) plus an "arc-type" linking element (<arc>). To create an equivalent unidirectional link as above with an extended-type link element, you could formulate it this way:

<p>To view a copy of the Declaration of Independance, click <linkgrp><resource label="start">here</resource><extptrloc label="document" href="document.html" role="text/html"/><arc from="start" to="document" show="new" actuate="onrequest"/></linkgrp>.</p>

Extended type linking elements <linkgrp> and <daogrp> and their child elements:

Resource	Locator	Arc
<resource>	<extptrloc>	<arc>
	<extrefloc>	
	<ptrloc>	
	<refloc>	
	<daoloc>	

As you may notice, it is much more complicated to properly create an extended-type link in EAD. The <resource> element can contain any information, but often serves as a way for the user to initiate the link. This can either be a textual link as above, or, if the element is left blank, some type of content (such as an icon) can be generated by whatever program or process interprets the link (XLink 1.0 section 5.1.1). A locator-type element (in this case, <extptrloc>) indicates the location of the resource being traversed to via the "href" attribute. Each of these "resources" contains a "label" attribute which is used by the <arc> element. The final required <arc> element indicates information about "how" and "when" the link should be traversed and also indicates the pairs of resources involved in the link via the "from" and "to" attributes. The importance of the <arc> element should not be overlooked. Unlike the simple-type linking elements, with an extended link, the locator element (<extptrloc>) "...does not constitute a link just because it has a locator (href) attribute..." per the XLink specification (XLink 1.0 section 5.1.2). Without the corresponding <arc> element, there is nothing in the markup to create an association between the various resources within this extended link, how to initiate the link, nor how or when to display the link. Thus, a markup like the following does not imply anything about (1) how to initiate the link such as whether it is to be embedded or displays new window, for example, or (2) when to display it such as when the document loads or when the user requests it. Only an <arc> element can supply this information.

Incomplete usage:

```
<daogrp>
<daoloc href="document.html" role="text/html"/>
</daogrp>
```

This type of markup does not tell a style sheet (or another program or process) how or when to display this link.

An extended-type link can come in handy in the cases where you want to embed an image in a document and then use that image as a link to a larger version of the same image. In EAD, this could be represented like this:

```
<linkgrp>
<resource label="start"/>
<extptrloc label="small" href="http://www.myserver.com/smallimage.jpg" role="image/jpeg"/>
<extptrloc label="large" href="http://www.myserver.com/largeimage.jpg" role="image/jpeg"/>
<arc from="start" to="small" show="embed" actuate="onload"/>
<arc from="small" to="large" show="new" actuate="onrequest"/>
</linkgrp>
```

This markup could be used to indicate (via the first <arc> element) that the first image is to be embedded when the document loads and that the second would open in a new window when requested by the user.

This would be equivalent to using the <img> tag in HTML. Since the ending resource (the image in this example) is set to be embedded by the <arc> element, then no content is generated that requires user-intervention to initiate the link, so the <resource> element is empty.

Mark Carlson
2004-07-20
Rev. 2004-09-23

My home page