

XML Linking Concepts

Linking concepts, XML linking, and beyond

Ulrike Schäfer, INFOTAKT information design, Würzburg, Germany (Web: <http://www.infotakt.de>, E-Mail: ulrike.schaefer@infotakt.de)

Abstract

One of the basic purposes we need links for is to get better, more efficient access to information. To achieve this purpose, we almost never need only links but a combination of different techniques. And we definitely never need only techniques but a good, user-oriented concept.

XML and related standards provide linking techniques that fulfill fundamental technical requirements for building powerful link networks. Since XML is designed as a web language and therefore shall be supported by a wide range of tools and applications, the technical preconditions will become much better than they were in the past.

Apart from this fact, the real challenge still remains: Building good, user-oriented concepts. This can't be done without a planned, well-organized mixture of abstraction and concrete testing during the design and implementation phase and afterwards.

What do we need links for?

Since relations between objects are a fundamental phenomenon, we need links in many fields. One of the basic purposes is to access information in a more specific way than given in linear texts: By providing not only texts but *hypertexts*. These are two classical situations:

Searching for a specific piece of information

To find the piece of information we are looking for, sometimes a search will do. But quite often we can't go so straight forward because we haven't enough background information to formulate our question detailed enough. In this case we rather combine different techniques. Navigation and retrieval are basic techniques used in many situations and different combinations to search for a specific piece of information. Web searches are very familiar examples for that approach: Quite often the result given by a search engine is only the beginning of the journey: From there we use the links given in the resulting sites and navigate through the pieces of information until we find what we are looking for.

Learning something about a topic

When we want to learn something about a - more or less - specific topic, good navigational aid is a basic requirement. In some situations we need to be guided very closely, like in online tutorials or training courses: Lesson 1, followed by lesson 2, etc. In other situations we want to navigate through a given information pool in a more flexible way, depending on the focus of our interest, like in online help systems, technical documentation, intranets, or the web. In this case we need more freedom to go wherever we want, but also more hints which ways to go: When I am navigating through information about some projects and I want to know more about the project managers, I may not be interested in their private homepages but in the past and current projects they managed. I need hints of what to expect behind the target of a link so that I don't have to follow them all in order to find what I am interested in.

But who is "we"?

In the example above, it depends on the users' profiles which information - and which relations between the pieces of information: which links - are relevant to them and how they should be presented to them: An advocate will be interested in other links between legal information than a private user.

Which information modules and which relations between them should be presented to the users? It depends

on the users' focus of interest, their backgrounds, their skills. And since one of the basic purposes of electronic media publishing is to reach new customer groups, the users' profiles get more and more heterogeneous.

And what is "the information pool"?

In printed media there are links like "see also..." that may point from one printed work to another. If we talk about electronic media and information pools, the borders between different pools don't have to be so fixed: The links between them could - and should - make them look like one information pool to the user. When I am interested in information about Leonard Bernstein, it would be best to have a link network that connects encyclopedias, works about conductors and composers (and even teachers), and biographies seamlessly.

Consequences: Which linking techniques do we need?

Which technical requirements result from these examples, and how does XML provide them?

Combining different techniques

We almost never need only links but a combination of different techniques: Linking, full text and context retrieval, metadata that describes resources and user profiles, document or text variants, and others. These techniques must be combined in a way that offers the users better, more specified access to the sort of information they need.

So when it comes to links, also other XML techniques have to be taken into account, like

- RDF, a framework for describing resources by metadata,
- XML-QL, a language for expressing context queries within XML documents,

and one of the basic requirements is that these languages can be combined accordingly. To provide this purpose, it is best to have a common standard that all these techniques are based on (like it is the case with XML languages), and to have them sharing common techniques for doing the same task, like

- XPath / XPointer for addressing resources and parts of them,

which are used by all XML languages that are dealing with addresses (XLink, XSLT, XML Query Language, XML Schema Language, and others).

The meaning of a link

Often the context shows me what to expect behind a link target:

```
Project manager: Linda Parkland (current projects, private homepage)
```

But in some cases - in particular when there are many kinds of links with heterogeneous meanings - it would be much more efficient to choose a filter, like: Please show me only the business-related links, hide the others. Apart from that fact, different kinds of links should be processed and presented to the user in different ways.

The link between two pieces of information has a meaning of its own. This meaning must become machine-readable as well. In XLink we can distinguish between different link *roles* to do this.

Flexible navigation

But where is the entry to the information pool? It depends on the user and the situation. Another user may start from the contact information of Linda Parkland and may be interested in the projects she is currently managing. To provide this flexibility more elegantly than creating another link from the contact information to the past and current projects, XLink's multidirectional links can be used: Thus a link is not something that points from a source to a target but a *relation* between two (or more) resources.

Links between different data pools

When we want to provide an *information* pool, it must be possible to cross the borders of different *data* pools seamlessly, i.e. to build links between our resources, like articles of an encyclopedia, and resources that are owned by others, like biographies, and not only to address an entire pool but a specific piece of information in it. This is what XPath / XPointer provide for: They enable us to address read-only resources without changing them.

Different link networks

Since there is not only one user with one focus of interest, there is not only one kind of relations between information pieces, and thus not only the one and only "right" way to build a link network in an information pool. As a consequence, it is necessary to build different link networks. That can be done efficiently by using XLink's extended links, which can be stored outside the linked data. This way we are able to build different link collections and to choose the according one depending on the user's profile.

And beyond

These are fundamental techniques needed in a wide range of information management scenarios: Web and intranet publishing, technical documentation, legal publishing, reference works, online help systems, and others. There are some other techniques and methods that can be combined with or based on XML techniques:

XLink and XPointer/XPath are derived from HyTime in order to provide techniques for fundamental, common linking requirements. HyTime includes additional facilities. But since HyTime and XLink are not incompatible, it is possible to migrate from XLink to HyTime when more sophisticated linking methods are needed.

Beyond these basic techniques, Topic Maps are designed to provide a concept for building knowledge networks and thus to offer a standardized method to organize information on a higher level.

Conclusion

XML provides a lot of technical and conceptual preconditions for building link networks according to our needs. To highlight the most important ones:

- Links themselves have a semantic that can be described machine-readable.
- Links are not pointers from a source to a target but a relation between resources and thus provide a technique for multidirectional linking.
- Links can be stored separated from the linked data. This technique can be used to build more than one link network for a given information pool.
- XML provides not only for linking techniques but also for other techniques that in most situations will be used combined with linking.

Last not least: The real challenge

Choosing the right linking techniques is a challenge, but that's only the beginning: We have not solved any user's problem by introducing XLink or HyTime techniques. The real challenge is to develop powerful and flexible, user-oriented applications to access information efficiently. Here are a few basic statements concerning this task:

- There is a simple, often mentioned, basic advice: We should not be impressed - and thus overwhelmed - by techniques. We should start with exploring what our requirements are, or, better to say, what the requirements of our customers are. If we know them, some techniques become less important to us, some become fundamental, some insufficient. There are no better criteria for that than our own, specific requirements.
- "I haven't seen any commercial TNM [Topic Map], but I'll bet, this will look like a Gordian knot. You take it and unless you hide 90% of the information you don't know anything." Rafal Ksiezuk [3]. The same can be said about link networks in general: User-oriented applications must not consist of mere link

networks. These link networks must be customized according to the users' interests, the kind of information they need to access, and the way in which they need to access it.

- This task can't be done in a mere abstract approach. Since the customer groups and using scenarios are getting more and more heterogeneous, we can't forecast them in theory. It is not a mathematical task that has only one right answer; there are many bad solutions, some better ones, and a few optimal ones. To find a good one, the entire design phase must be a well-organized mixture of abstraction and concrete testing - and again abstraction, and again testing.

It is like always: XML is an enabler, not a problem solver. XML is as good or as bad as the way we use it.

References

- Martina Hemrich, Ulrike Schäfer: XML-based linking concepts. Online Information 1999.
- Eliot Kimber: Practical hypermedia. An introduction to HyTime. 1996/98 (<http://www.drmacro.com>)
- Rafal Ksiezyk: Trying not to get lost with a topic map. XML Europe 1999 (<http://www.topicmaps.de/content/resources/xmleur99/rak.htm>).
- World Wide Web Consortium: Technical reports:
 - XLink (XML Linking Language: <http://www.w3.org/TR/xlink/>)
 - XPath (XML Path Language: <http://www.w3.org/TR/xpath>)
 - XPointer (XML Pointer Language: <http://www.w3.org/TR/xptr>)
 - XML-QL (A Query Language for XML: <http://www.w3.org/TR/NOTE-xml-ql/>)
 - RDF (Resource Description Framework: <http://www.w3.org/TR/REC-rdf-syntax/>, <http://www.w3.org/TR/rdf-schema/>)