# Open Hypermedia as User Controlled Meta Data for the Web

**Kaj Grønbæk, Lennert Sloth, Niels Olof Bouvin**

Department of Computer Science, Aarhus University,

Åbogade 34, DK-8200 Århus N, Denmark.

Email: {kgronbak,les,n.o.bouvin}@daimi.au.dk

## Abstract

This paper introduces an approach to utilise open hypermedia structures such as links, annotations, collections and guided tours as meta data for Web resources. The paper introduces an XML based data format, called Open Hypermedia Interchange Format - OHIF, for such hypermedia structures. OHIF resembles XLink with respect to its representation of out-of-line links, but it goes beyond XLink with a more rich set of structuring mechanisms, including e.g. composites. Moreover OHIF includes an addressing mechanisms (LocSpecs) that goes beyond XPointer and URL in its ability to locate non-XML data segments. By means of the Webvise system, OHIF structures can be authored, imposed on Web pages, and finally linked on the Web as any ordinary Web resource. Following a link to an OHIF file automatically invokes a Webvise download of the meta data structures and the annotated Web content will be displayed in the browser. Moreover, the Webvise system provides support for users to create, manipulate, and share the OHIF structures together with custom made web pages and MS Office 2000 documents on WebDAV servers. These Webvise facilities goes beyond ealier open hypermedia systems in that it now allows fully distributed open hypermedia linking between Web pages and WebDAV aware desktop applications. The paper describes the OHIF format and demonstrates how the Webvise system handles OHIF. Finally, it argues for better support for handling user controlled meta data, e.g. support for linking in non-XML data, integration of external linking in the Web infrastructure, and collaboration support for external structures and meta-data.

## Keywords

Open Hypermedia, XML, Meta Data, Semantic Web, WebDAV, XLink.

## 1. Introduction

Recently the notion of meta data has received a lot of attention in relation to the WWW [1], through the introduction of standards such as XML [4] and RDF [26]. Meta data is seen as a potential means to create the "Semantic Web" as called for by Tim Berners-Lee in his WWW8 keynote. Services such as Alexa and the

Netscape "What's related?" predates RDF. These meta data services are mainly based on author supplied or automatic generated meta data. This paper goes beyond these services and introduces a mechanism also for users or groups of users to control and generate their own meta data and structures for specific purposes.

In the Digital Library domain, Marshall [20] has argued that user generated annotations and structures are useful meta data for other users of a digital library collection who need to pursue research in the same territory as other colleagues have covered earlier. The Webvise system and the OHIF format discussed in this paper are aimed at supporting this kind of meta data creation by end users. The principles and the approach is inherited from Open Hypermedia System research.

Open Hypermedia Systems research [8, 14, 23, 24, 25], has been working since the early nineties on providing general support for user controlled annotations and structuring which can be kept separate to the documents containing the information content. Recently open hypermedia systems have been developed with support for annotating and structuring Web content [3, 10, 12, 16, 18, 21].

This paper describes how the Webvise [12] open hypermedia system has been extended to act as a tool for users to generate and control meta data in an XML format which can be distributed on the Web similar to the documents containing the base information. The meta data format is an XML encoding of a data model agreed upon in the Open Hypermedia Systems Working Group [25]. This in turn comes from earlier work on developing interchange formats for general hypermedia structures. Interchange formats have been a research and standardisation focus for many individual types of media, such as text and video. Within the area of hypermedia, work on interchange mainly focus on the hypermedia structures and not the contents per se. The Dexter model [13] made a proposal for a SGML-family interchange format for hypermedia structures. This proposal was used e.g. to interchange hypertexts between two quite different hypermedia systems KMS and Intermedia [17]. Later the HyTime standard [6] was introduced as a general format to cover hypermedia structures as well as synchronisation mechanisms for time based content. However, these formats have not been supported by general tools on the Web yet and they have thus not been put into widespread use. The upcoming Xlink language, however, represents a simpler and easier implementable version of HyTime-like linking mechanisms.

The Open Hypermedia Interchange Format (OHIF) that we propose in this paper is taking advantage of XML and it is thus simpler and easier to build tools for than HyTime. Currently we have two prototypes Webvise [12] and Arakne [2] that can read and generate the format. Currently the Webvise prototype can be downloaded for trial from **http://www.cit.dk/coconut** under "Prototypes".

Finally, we have addressed the issue of supporting collaboration on meta data and open hypermedia structures by building WebDAV [30] support into our Open

Hypermedia client applications. Webvise is thus able to open and save meta data files to WebDAV servers, it takes and releases locks such that collaborating users are prevented from lost updates on the meta data and content files that are shared on the WebDAV server.

The rest of the paper is organised as follows. Section 2 provides a brief overview of current work on supporting meta data for the Web. Section 3 introduces open hypermedia structures and the OHIF format as meta data for the Web. Section 4 describes the kind of support for user controlled meta data that is provided by the Webvise system. Section 5 describes the WebDAV based collaboration support for meta data provided by Webvise. Section 6 concludes the paper.

## 2. Current meta data support on the Web

The amount of raw data on the Web has reached a point, where it has become difficult to find pages precisely relevant to a given topic. As it stands, the Web can be compared to a giant library with no index, no catalogue, and no librarian. Sophisticated search engines abound, but these face many problems:

- **The sheer size of the Web makes it very difficult to maintain a complete and up to date index**
- **Keyword spamming is quite common (i.e. the practice of manipulating a Web page's ranking in a query result by e.g. inserting superfluous keywords to fool search engines)**
- **Most searches result in (many) pages unrelated to the topic, the user is interested in**

A natural consequence has been the proliferation of Web sites that categorise other Web sites (e.g. www.yahoo.com, www.about.com). These Web sites essentially provide information about information, or meta data. This is however mainly human readable information, difficult for computers to handle and analyse.

Meta data can take many forms. It can be information about the author of some pages, a classification of the information found on the page, the relationship between these pages and other pages, a ranking of the pages in some index, or relations between indices. A crucial element of meaningful meta data is a clear and widely adhered to standard. An example of a widely used classification standard found in libraries is the Dewey Decimal Code. A system for bibliographic information often cited in conjunction with RDF is the Dublin Core (**http://purl.oclc.org/dc**).

### 2.1 The Resource Description Framework (RDF) and the emergence of meta data on the Web

RDF [26] is a World Wide Web Consortium Recommendation for the description of meta data relating to data found on the Web (or more precisely anything that can be addressed with a URI). Meta data adhering to the RDF standard is usually encoded in XML, and consists generally of a series of statements. Statements are triplets of a resource (addressed with a URI), a property type (an attribute of the resource, such as 'author'), and a value (such as 'John Doe'). Statements can be resources or values for other statements. By using XML Name Spaces it is possible to mix different meta data standards in a statement.

While XML is but one representation of RDF (RDF can also be represented as graphs or 3-tuples), meta data is usually distributed as XML files. To support inter-operation between related meta data formats, name spaces as well as schemas are utilised. Protocols such as querying/searching are currently not supported, though listed as possible future activities by the RDF interest group. RDF is still a new technology and has not yet become widespread in use. One question that remains is this: who will author the meta data? Meta data is of enormous importance in the library world, where there is a well-established culture regarding the creation of meta data, and where meta data (such as an index card) can be relied on. Not so on the Web, where the <meta> header tag introduced in HTML has become widely misused through keyword spamming, rendering the information highly unreliable. This is most unfortunate, but a reality of the Web. Meta data is as such no silver bullet for the Web's current information overload. A likely development will be the development of Web sites concentrating on specific subject domains, within which meta data can be meaningfully defined and relied on by users, thus establishing a community of use.

### 2.2 RDF and Open Hypermedia

RDF and (open) hypermedia are both ways of describing relations between entities. Both are externally stored outside the documents, they reflect upon, and both provide a reader with an 'added value' or a structuring mechanism. A major distinction between RDF and open hypermedia is the addressing scheme used. RDF resources are identified by a URI. The advantage is that URI is very well supported on the Web. The disadvantage is that it severely limits what can be addressed and how. URI is a coarse grained addressing scheme limited to whole documents or predefined (named) regions within. Through the use of LocSpecs [7] in open hypermedia, arbitrary selections can be addressed. This makes it easier to create a relationship (e.g. a link) between two or more statements in different documents of diverse media types. Conceivably RDF could be combined with addressing schemes such as XPointer to provide a finer granularity.

RDF and open hypermedia can work well in conjunction with each other, as they address different needs. RDF is very well-suited for making general statements about documents and their relationship to other documents. Open hypermedia on the other hand is a general structuring mechanism, and can be used to express explicit relationships between multiple statements in documents of arbitrary types.

### 2.3 XLink, XPointer, and XPath

Linking to and from XML documents is to be supported through the use of XLink [32] (used to describe links), XPointer [33] (used to identify regions in XML documents to link to and from), and XPath [34] (used by XPointer to identify regions in XML documents using a hierarchical scheme). As of this writing (29/2/2000) XPath 1.0 is Recommended by the W3C, XLink is pending Last Call for comments, and XPointer has passed the Last Call for comments.

XLink is a general tool to describe navigational hypermedia, and to allow expressions of navigational hypermedia to be inserted into XML documents. While the main application of XLink is expected to be linking *in* XML documents, the standard itself is not limited to address solely XML locations, provided that appropriate locators have been defined. XLink can support the linking currently found on the Web (e.g. unidirectional 1-ary untyped links), as well as bi-directional n-ary typed links. Links can be stored externally ('out-of-line') of the documents, they address, or they can be in-line (as with HTML documents). Traversal of a link may result in replacing the document currently viewed (as is the standard behaviour in the context of HTML), or by inserting the target for the link in the viewed document. The traversal may be initiated by the user (e.g. by clicking on a link), or at the time of document retrieval.

XPointer is used to identify regions of interest in XML documents. XPointer allows for selection based on ids, hierarchical structure (from XPath), or an arbitrary user selection (e.g. selecting a string in the rendered XML document). This is a quite sophisticated addressing scheme that should cover most uses. XPointer can address arbitrary XML documents, but the explicit support for XPointer can also be added to a DTD. A given region may be identified using several locators, which improves reliability, as one locator might fail after a document has been edited.

XLink is designed for navigational hypermedia, which has long been the classic hypermedia application. While linking is certainly still very important in hypermedia, new structuring mechanisms have been introduced which cannot easily (or at all) be described in terms of links. This goes for composites, guided tours, spatial, or taxonomic hypermedia [24].

XLink/XPointer is more closely related to open hypermedia than is RDF. On the whole these standards address similar goals and employ similar means to achieve this. One major difference is the integration of third party applications, that is the core of open hypermedia [22]. Open hypermedia is explicit about this support, and much effort has been done in the open hypermedia community to integrate third party applications with open hypermedia systems. While XLink is general in the sense that locators can be (and have been) created to allow linking into non XML documents, the specification is vague on the behaviour of the applications that support these document types. Not all applications can be expected to be conforming to XLink, and how XLink works with such applications remains unclear.

### 2.4 Alexa/What’s Related

A widely used service on the Web is the one provided by Alexa. Alexa is available as a stand alone tool as well as integrated with several Web browsers. By clicking a 'What’s Related' button (the exact invocation depends on the Web browser) users are presented with Web sites related to the one currently shown. The result is generated on basis of the collective browsing habits of people using Alexa, which is stored centrally. Whereas the model behind open hypermedia, RDF, and XLink/XPointer is based on humans asserting relationships between entities, Alexa functions solely by relying on the relevance between consecutively visited Web sites. In practice the authors have found this to work quite well, especially for high profile Web sites. With less popular Web sites the success rate seems to be lower, which should be expected, as the 'error rate' (users consecutively visiting Web sites not related to each other) would be more significant. As a tool Alexa is well suited to give suggestions to sites that might be of interest. However as the relationship between pages is wholly statistical rather than semantic, it is of little help, when searching for an answer to a specific question. Furthermore the individual user has little control over which pages are presented. Alexa works well in conjunction with the above described technologies as a tool for gathering Web sites relevant for the task at hand; Web sites that must later be sorted by a human. In this manner Alexa is similar in use to ordinary search engines. By the nature of how it works, Alexa is restricted to relate whole Web pages to each other, which is also a marked difference between it and open hypermedia/RDF/XLink/XPointer.

### 2.5 Flyswat

A recent development is the Flyswat service (www.flyswat.com). The Flyswat service (through the use of an ActiveX component) highlights keywords on Web pages, so that a click on the keywords can lead to a definition of the word, the Web site associated with the term, or a page where the item may be purchased. The effectiveness of Flyswat greatly depends on whether a given subject is covered by the Flyswat company or any of the associated partners. There does not seem to be any possibility of submitting links or keywords to the service, outside of becoming an associated partner with Flyswat. Thus, the system does not provide for *user* controlled meta data or hypermedia functionality.

## 3. Open Hypermedia Structures as Meta Data

As mentioned in the introduction, we consider relationships and annotations made by users of Web resources as potential useful meta data for other collaborating users or for the general public. An example of user annotations as meta data are the reviews made by readers of books from Amazon.com, and an example of automatically generated meta data relationships are the information that "buyers of this book also bought ...". Amazon.Com is an example of a site that provides this specific service for its visitors, based on opening up their site for adding and computing such meta data relationships.

Another example of the need for user controlled meta data is in the areas of Digital Libraries [19, 20], where readers of a particular resource may benefit from the fact that another reader has read the material and commented on it with marginalia, similar to what is possible in paper books.

Instead of having every Web site implementing their own forms and databases for user generated meta data, we are proposing a general open hypermedia mechanism that can provide support for such meta data for arbitrary Web sites, thus reducing the cost of developing Web sites that enables dynamic exchange of information among its users.

In the following sections we will describe a meta data format and tools to help users (collaboratively) to generate and control meta data, including annotations and relationships.

### 3.1 The basic open hypermedia data model

In the OHS community a standard data model for open hypermedia structures has been proposed [9]. An outline of the currently agreed upon OHSWG navigational data model is depicted in Figure 1.

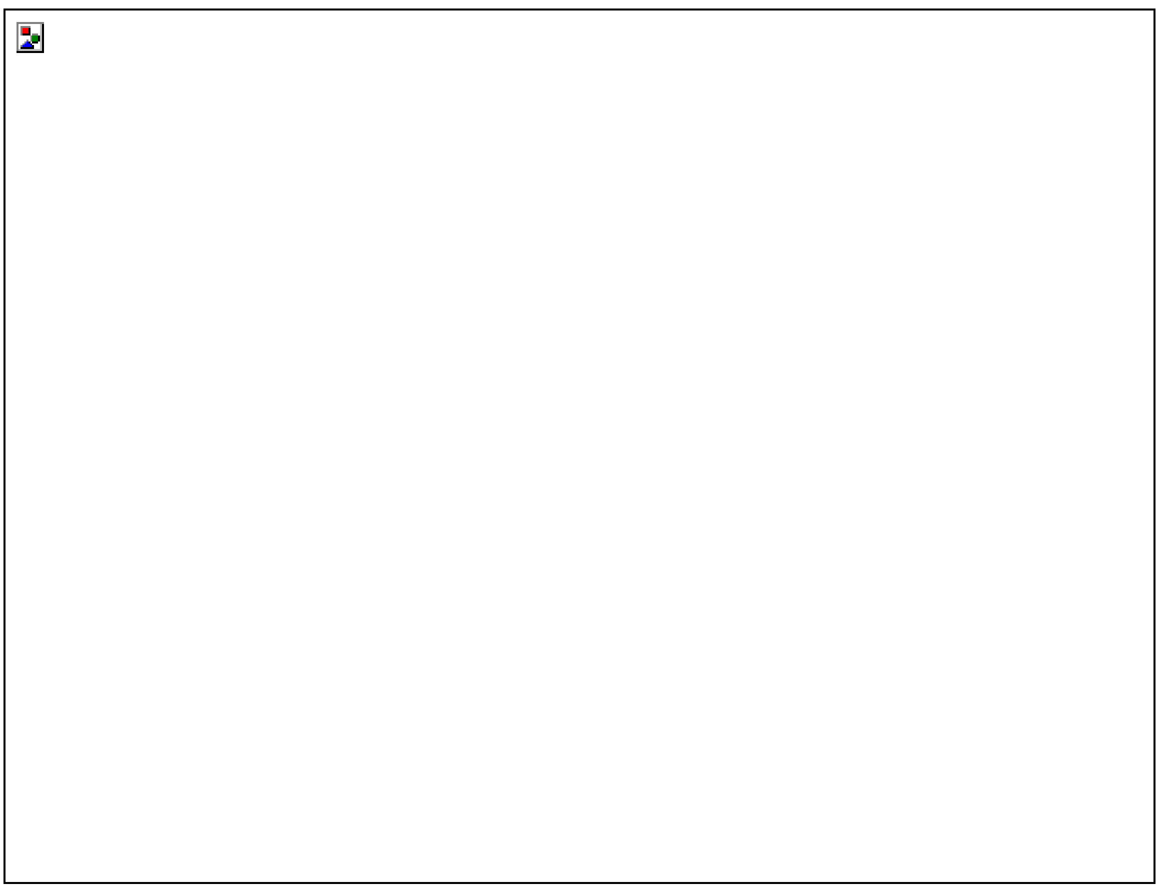


***Figure 1: Outline of OHSWG navigational data model***

From this data model, an XML DTD (See **http://www.daimi.au.dk/~les/ohif/ohif.dtd** ) is derived which in turn allows OHS systems to use standard XML parsers to load the structures into OHSs, which are compliant with the standardised data model. The derived data format is called OHIF, and excerpts of the XML are shown below. Each OHIF file represents a 'context' object of the data model. A context is an (indexed) collection of other hypermedia objects. An example of a node is shown in Table 1.

```
<NODE id="ariel.xserver.2" type="NODE" name="DOM">
  <DESCRIPTIONSET>
    <DESCRIPTION name="default">
      <VALUE>DOM Specifications</VALUE>
    </DESCRIPTION>
  </DESCRIPTIONSET>
  <CHARACTERISTICSSET></CHARACTERISTICSSET>
  <COMPUTATIONID></COMPUTATIONID>
  <CONTENTSPEC>
    <URL>http://www.w3.org/TR/REC-DOM-level-1/</URL>
    <CONTENTS></CONTENTS>
    <MIMETYPE>Text/HTML</MIMETYPE>
  </CONTENTSPEC>
</NODE>
```

***Table 1: Example of an OHIF 'node' element***

OHIF links are general many-to-many relationships as supported first in the Dexter model [13] and HyTime [6]. The OHIF link is thus a collection of endpoints as shown in Table 2.

```
<LINK id=ariel.xserver.6 type="LINK" name="DOM relationships">
  <DESCRIPTIONSET>
    <DESCRIPTION name="default">
      <VALUE>DOM specifications</VALUE>
  </DESCRIPTIONSET>
  <CHARACTERISTICSSET></CHARACTERISTICSSET>
  <COMPUTATIONID></COMPUTATIONID>
  <ENDPOINTIDSET>
    <ID>xsite.xserver.10</ID>
    <ID>xsite.xserver.14</ID>
    <ID>xsite.xserver.20</ID>
  </ENDPOINTIDSET>
</LINK>
```

***Table 2: An example of an OHIF 'link' element***

The OHIF link element resembles links in the upcoming XLink specification [32], and when the XLink standard is stable we may consider simply using the XLink language and name spaces for specifying the linking part of the OHIF format.

However, we need support for other structuring mechanisms such as composites [13] like collections and guided tours to get the power needed to express more elaborate hypermedia structures. With respect to locating and addressing we need to go beyond the URL and XPointer specs and get support for LocSpecs to locate fragments of arbitrary non-XML based media as proposed in [7].

Endpoints are responsible for addressing the anchor which is responsible for keeping the LocSpec for a specific location inside a node's content. But endpoints also hold a presentation specification (PSpec) which allow a link to determine how the endpoint node should be presented when arrived at via this link. The PSpec may e.g. impose a specific stylesheet on the document as a side effect of following the link. Endpoints also hold a direction attribute, with information about whether the endpoint is considered a source, a destination or both.

```
<ENDPOINT id="ariel.xserver.9" type="ENDPOINT" name="Professor"
          linkid="xsite.xserver.3" anchorid="xsite.xserver.10"
          direction="BIDIRECTIONAL">
  <PSPECIDSET>
    <ID>xsite.xserver.7</ID>
  </PSPECIDSET>
</ENDPOINT>
```

***Table 3: An example of an endpoint element. This endpoint is referenced by a link with id xsite.xserver.3, its presentation specification is stored a pspec with id xsite.xserver.7 and the location specification is stored an anchor with id xsite.xserver.10.***

The anchor element inlines the LocSpec to locate the actual part of a node's content which constitute a source or a destination for the link.

```
<ANCHOR id="xsite.xserver.6" type="ANCHOR" name="Information" >
  <CHARACTERISTICSSET>
    <CHARAC name="userNote">
      <VALUESET>
        <VALUE>This mobile phone is the ...</VALUE>
      </VALUESET>
    </CHARAC>
  </CHARACTERISTICSSET>
  <PSPECIDSET>
    <ID>xsite.xserver.5</ID>
  </PSPECIDSET>
  <SIMPLELOC>
    <SELECTION>Information</SELECTION>
    <SELECTIONCONTEXT>Information about the Coconut Project
     </SELECTIONCONTEXT>
  </SIMPLELOC>
</ANCHOR>
```

***Table 4: An example of an 'anchor' element. The anchor contains location information in a 'locspec' element. This anchor also contains the id of the shown in table 5. The pspec designates this anchor to be presented as an annotation anchor.***

```
<PSPEC id="xsite.xserver.5" type="PSPEC">
  <CHARACTERISTICSSET>
    <CHARAC name="userNoteKind">
      <VALUESET>
        <VALUE>Popup</VALUE>
      </VALUESET>
    </CHARAC>
  </CHARACTERISTICSSET>
</PSPEC>
```

***Table 5: An example of a 'pspec' element. This is a pspec for an annotation anchor.***

The anchor and pspec shown above implement annotations, similar to ThirdVoice (www.thirdvoice.com) and ComMentor [27] in a specific location inside a Web document [12].

The above elements are included in the basic OHSWG data model for open hypermedia. However, the users of our prototypes are requiring several structuring mechanisms that are not yet covered in the data model. But the OHSWG data model has on purpose been specified as an extensible model, where new structures may be added without disturbing the standardised handling of the basic elements. In the next section we will show a few examples on such extensions, which have already been implemented on top of the basic model.

### 3.2 Example of an Extension to the Core OHSWG Data model

Our open hypermedia systems, Webvise and Arakne, both support guided tours similar to the mechanisms described by [16, 29]. This mechanism is not included in the basic open hypermedia data model, but since the data model is extensible, we can add this specialised data model element without interfering with the general and common data model elements. Extending the XML based DTD with this new data element will thus just cause the systems not supporting the data element to skip it in the XML parse procedure.

Table 6 show an example of such an OHIF extension namely a guided tour, which is a composite representing a graph of nodes.

```
<GUIDEDTOUR id="xsite.xserver"
            type="GUIDEDTOUR"
            name="Mobile computing">
  <COMPREFIDSET>
     <ID>xsite.xserver.8</ID>
     <ID>xsite.xserver.12</ID>
     <ID>xsite.xserver.6</ID>
     <ID>xsite.xserver.10</ID>
     <ID>xsite.xserver.4</ID>
  </COMPREFIDSET>
  <EDGEIDSET>
     <ID>xsite.xserver.14</ID>
     <ID>xsite.xserver.13</ID>
  </EDGEIDSET>
</GUIDEDTOUR>
```

```
<EDGE id="xsite.xserver.14"
      type="EDGE"
      originid="xsite.xserver.10"
      destinationid="xsite.xserver.12">
</EDGE>
```

***Table 6: An example of a small guided tour with five vertices and two edges, as well as an example of an 'edge' element. Besides being a HMObject an edge contains the ids of the two vertices it is connecting.***

Finally, a first attempt has been made on providing semantic types for open hypermedia links which in turn can bring us closer to what Tim Berners-Lee called the "Semantic Web" in his WWW8 keynote. The type mechanism is inspired by the hypermedia type system introduced by Trigg in his TextNet system [28]. When introducing types on links and other hypermedia objects, the need for a full-blown hierarchical type system is revealed, and it thus requires an XML representation in it self. Such a type system and an XML representation has been proposed in [15], and it is illustrated in Table 7. It has, however, not been implemented in OHIF yet.

```
<LINKTYPES>
  <LINKTYPE typeid="0" typename="Link Type">
    <TYPEATTRIBUTES>
      <ATTRNAME>noOfUsages</ATTRNAME>
      <ATTRVALUE>0</ATTRVALUE>
```

```
      </TYPEATTRIBUTES>
      <SUPERTYPES></SUPERTYPES>
    </LINKTYPE>
    <LINKTYPE typeid="1" typename="Quality of Source">
      <TYPEATTRIBUTES>
        <ATTRNAME>Why</ATTRNAME>
        <ATTRVALUE></ATTRVALUE>
      </TYPEATTRIBUTES>
      <SUPERTYPES>
        <TYPEID>0</TYPEID>
      </SUPERTYPES>
    </LINKTYPE>
  </LINKTYPES>
```

***Table 7: Representing a hierarchical link type system as proposed by Hansen et al. (1999) [15]***

Having introduced the OHIF meta data model and its XML format, we will in the next section illustrate how it can be utilized to provide support for users to generate, control, and manipulate meta data for arbitrary Web resources.

## 4. Support for open hypermedia meta data

Webvise is an application that augments Microsoft Internet Explorer with open hypermedia services. It provides structures such as contexts, links, annotations and guided tours that are stored and read from OHIF files. An OHIF file represents a context and when an OHIF file is loaded by Webvise, links and annotations are dynamically merged into existing HTML pages, and the guided tours can be displayed in graphical browsers. The Webvise service consists of a structure server and a client application. See [12] section 3 and 4 for details about Webvise. In the following section we will describe different methods for transferring OHIF files over the internet and for loading them into Webvise, which in turn imposes the OHIF structures on existing HTML pages.

### 4.1 The Webvise approach

The structures stored in a given OHIF file can be applied by loading the file into Webvise. If the OHIF file is stored locally the user can open the file using a traditional file dialog in Webvise. If the file is stored on a HTTP server the user can open the file by clicking a URL in a Web page or by typing the URL in the address field of the Web browser. When navigating to an OHIF file residing on a HTTP server using Microsoft Interne Explorer, a handler, either a stand alone application, an ActiveX object or a plug-in, must be registered to handle the file. Only the use of stand alone applications and ActiveX objects are discussed here.

If the protocol of the URL is HTTP the action taken by Internet Explorer depends on what kind of handler that is associated with the OHIF file type. If the handler is an application, Internet Explorer downloads the OHIF file and then invokes the application with the path to the downloaded file. This might be a problem if the structure server (see [12]) used by the application handler is not running locally, since it is the structure server that must parse the OHIF file and build the open hypermedia structures and this would require that the OHIF file would be sent over the wire one more time.

One solution to this problem is to define another file type, e.g. OHL (open hypermedia locator) that only contains the URL of the “real” OHIF file and then use URLs to OHL files instead. The handler application would then be able to send only the URL of the OHIF file to the structure server instead of the whole content of the file. This would be similar to the approach taken by Real Media (http://www.real.com), with their .ram files.

Another solution is to define a new URL protocol and then register the handler application to this protocol. This will make Internet Explorer omit downloading the file and invoke the handler application with the URL. This is one of the approaches supported by Webvise, which registers the OHTP protocol when installed. An example of how to use the OHTP protocol is shown in Table 8.

```
   <A onclick="return confirmDefault()"
HREF="ohtp://www.daimi.au.dk/~les/ohiffiles/www9/xanadu.ohif ">Xanadu.</A>
```

***Table 8: An example of an OHTP link embedded in a standard HTML page. 'confirmDefault' is a java script that informs the user that he/she must have Webvise installed to be able to use the link.***

A third possibility is to use an ActiveX object (an In-Proc Server) as handler. Registering the ActiveX object as the default player for the OHIF MIME type can do this. Using this approach Internet Explorer loads the ActiveX object and the URL is passed to the ActiveX object. This method is used by Webvise to handle the situations where the protocol of the URL is HTTP. The ActiveX object used in the current implementation does not have a user-interface and simply passes the URL on to the Webvise application. However, we do plan to add a user-interface and (some of) the open hypermedia functionality found in Webvise to the ActiveX object. This will allow a tighter integration with Internet Explorer and remove the need for a stand alone handler application.

Regardless of the method used to load the OHIF file, the result is that the structures stored in the OHIF file are used to augment the original HTML pages that they refer to. Links and annotations are dynamically merged into existing HTML Web pages, i.e. the locations of links and annotations are decorated similar to how ordinary Web links are shown, but with distinct colours. Composites such as guided tours and collections are displayed in graphical browsers residing in the Webvise application. The merge of meta data and content is illustrated in Figure 2.

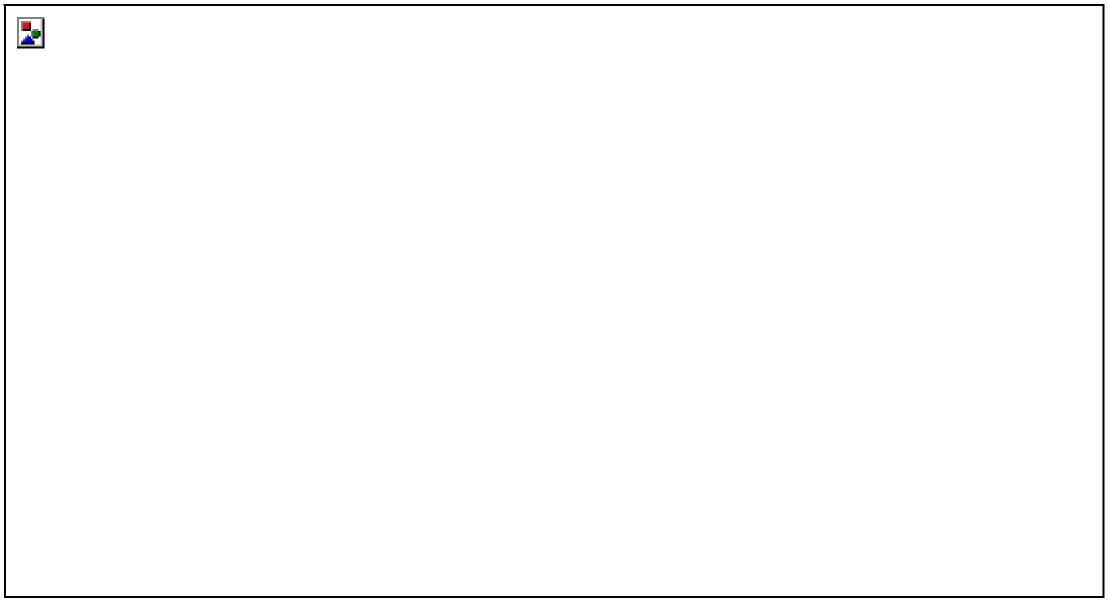


***Figure 2: HTML Web pages are augmented with open hypermedia structures stored externally from the Web pages in OHIF files.***

The Webvise approach demonstrates how meta data and out-of-line hypermedia structures can be integrated with ordinary Web content in a seamless manner, allowing meta data and structures to become first class citizens on the Web. The fact that the OHIF format is in XML also opens up for possibilities for letting the Web browsers display and utilise the data directly by means of XSL and transformations to dynamic HTML, thus not requiring a helper application for plain read access.

## 5. Collaborating on open hypermedia meta data

When a group of users are doing research on a common subject area they may wish to share their meta data among the group and collaboratively construct a common set of meta data for the subject area. Previous research in open hypermedia has proposed three layered architectures with servers being responsible for handling concurrency control, access control and event notifications on hypermedia objects at runtime [11, 31]. However, to support a simpler and less tightly coupled collaboration on open hypermedia structures via the WWW, WebDAV client support has been added to Webvise. This enables Webvise to offer controlled sharing of OHIF files. In this section we describe the WebDAV features currently available in Webvise.

Since each OHIF file contains one context, contexts represent the level of granularity at which open hypermedia structures can be shared. The WebDAV client features implemented in Webvise are: LOCK, UNLOCK and retrieval of the **`lockdiscovery`** property via the PROPFIND method. Standard HTTP PUT and GET are also supported. If the user has a lock on a resource PUT requests on that resource will contain the **`locktoken`** in an **`If`**header.

To collaborate on a given context a user of Webvise must first request a lock on the OHIF file from the WebDAV compliant server on which the file is stored. The dialog used to lock the file is depicted in Figure 3.

***Figure 3: The dialog used in Webvise to lock an OHIF file on a WebDAV server. http://www.sharemation.com/~les/demo1.ohif is the file on which the user les requests a lock. The owner resource http://www.daimi.au.dk/~les should contain contact information for the owner of the lock. The timeout field indicates the number of seconds after which the WebDAV server automatically should remove the lock.***

If the request for the lock is successful the file is downloaded. The context can now be modified using the usual editing facilities in Webvise and at any time the user can save the context and then upload it to the WebDAV server (these operations may be bundled into a compound operation). Having uploaded the modified context the user can choose to remove the lock on the OHIF file or he/she can continue to edit the context and then perhaps upload the context again. This way the user can take a long-term lock on a context and publish changes to the context in several smaller steps.

If it is not possible to obtain a lock on a given context, the context can still be downloaded and modified. To publish the modified context it is necessary however, to store it under a different name. When WebDAV servers that support versioning become available it should be possible to store the context under the same name but as a new version.

When taking a lock users are encouraged to supply a URL to a Web page that contains contact information to enable other users to get in contact with the lock owner. Webvise users can inquire the WebDAV server for lock information to get this information.

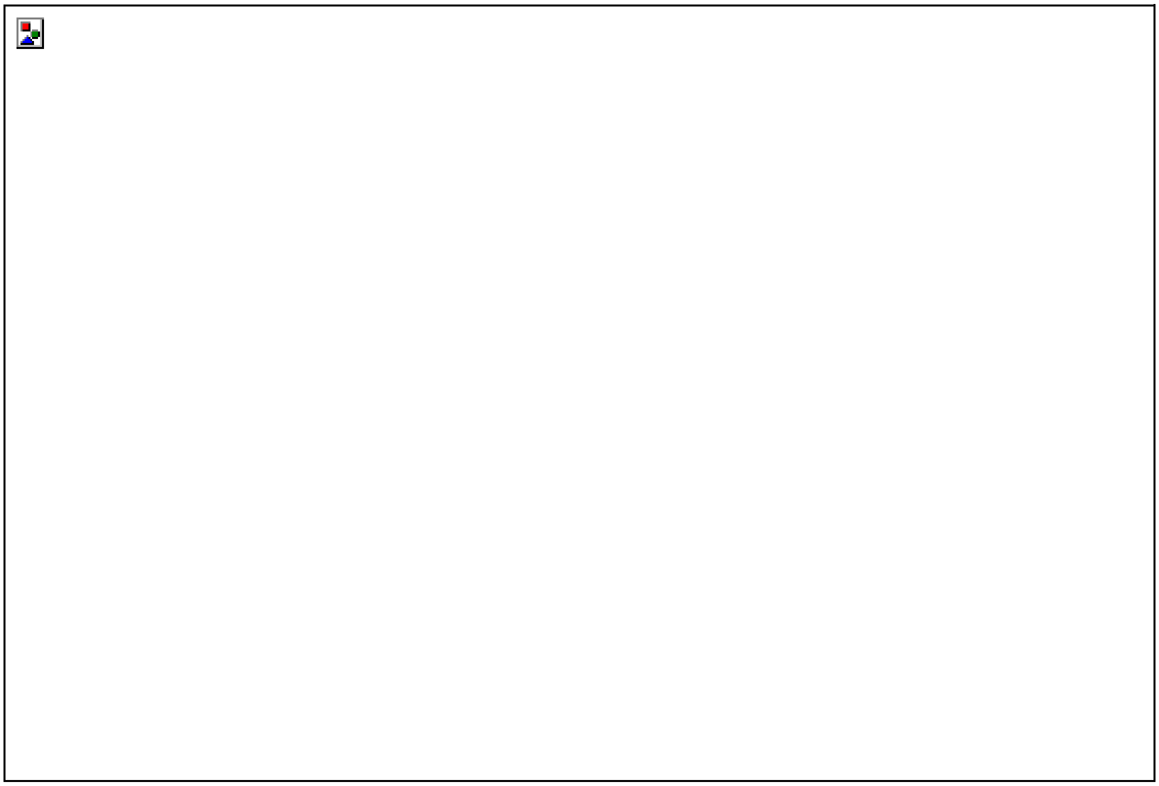


***Figure 4: A scenario, where Webvise is used by a group of users with accounts on Sharemation to link and annotate Web materials and Word files about Palm devises . The Word and OHIF files are shared on a Sharemation account, and they provide meta data for Web pages such as relevant product documentation and FAQs.***

Having integrated WebDAV support in Webvise, makes it possible to support distributed maintenance of the meta data and structure files, as well as accompanying files written in applications supporting WebDAV, e.g. Microsoft Office 2000. Apart from linking into Web pages, Webvise also support linking to and from Microsoft Word and Excel. Compared to ealier Open Hypermedia integrations with MS Office developed in Microcosm [14] and Devise Hypermedia [8] it is now for the first time possible to have fully distributed Open Hypermedia linking between MS Office documents and Web documents since we can get global URLs for the MS Office documents on the WebDAV servers (see Figure 4). Ealier integrations only worked locally on networks that could mount shared drives and use ordinary local filesystem addressing schemes. Thus the spreading of WebDAV to Web servers world wide is an important step to achieve fully distributed open hypermedia linking between Web documents and arbitrary WebDAV aware applications.

## 6. Conclusion

This paper has proposed an XML based format for open hypermedia structures to be utilised for supporting user controlled meta data and hypermedia structures on the Web. Moreover, it has proposed to transfer the meta data format with a new protocol prefix called 'OHTP', which can be used to start transfer of meta data (OHIF files). The transfer can be the responsibility of the Web browser per se, or a separate helper application. Currently two available helper applications Webvise and Arakne supports this OHIF format and protocol. OHIF covers some of the same territory as the proposed XLink language and the XPointer standard, but OHIF contains locating facilities for arbitrary non-XML data and a more rich set of structuring mechanisms, including composites, such as guided tours, and it would require extensions of the scope of the XLink language and XPointer to cover the full power of OHIF. It is our hope that the OHIF structuring and addressing mechanisms will become a new supplementary standard, or alternatively that the emerging standards will adopt the ideas and incorporate the mechanisms.

## 7. Acknowledgements

This project is supported by CIT - The Danish national centre for IT research grant no. 123 and The Danish Research Councils' Centre for Multimedia. Thanks to our colleagues in the Coconut and DMM projects for their contributions to the work. In particular we acknowledge Klaus M. Hansen and Christian Yndigegn for their work on extending the OHIF format with an open type system.

## 8. References

[1] Berners-Lee, T., Cailliau, R., Groff J.-F., Pollerman, B. World-Wide Web: The Information Universe. Electronic Networking: Research, Applications and Policy 1(2), 1992.

[2] Bouvin, N.O. Unifying Strategies for Web Augmenting. In the proceedings of Hypertext '99 (The tenth ACM conference on Hypertext and Hypermedia). Darmstadt, Germany, February 21-25 1999, ACM, New York, 1999 pp. 91-100.

[3] Bouvin, N.O., & Schade, R. Integrating Temporal Media and Open Hypermedia on the World Wide Web. In Proceedings of the 8th Inter-national World Wide Web 99 Conference, Toronto, Canada, 1999, pp. 375-387.

[4] Bray, T., Paoli, J., Sperberg-McQueen, C. M. Extensible Markup Language (XML) 1.0. W3C Recommendation 10-February-1998. **http://www.w3.org/TR/1998/REC-xml-19980210**

[5] Davis, H. Lewis, A. & Rizk, A. OHP: A Draft Proposal for a Standard Open Hypermedia Protocol. In *2nd Workshop on Open Hypermedia Systems*, Washington DC: University of California, Irvine, 1996 pp. 27-53.

[6] DeRose, S. J. & Durand, D. G. *Making hypermedia work: A user's guide to HyTime*. Boston/Dordrecht/London: Kluwer, 1994.

[7] Grønbæk, K. & Trigg, R.H. Toward a Dexter-based model for open hypermedia: Unifying embedded references and link objects. in HYPERTEXT '96 – Seventh ACM Conference on Hypertext. Washington DC, USA, March 16-20, 1996.

[8] Grønbæk, K. & Trigg, R.H. *From Web to Workplace: Designing Open Hypermedia Systems*. MIT Press, Boston Massechussets. July 1999, 424 pp (ISBN 0-262-07191-6).

[9] Grønbæk, K. OHS Interoperability - issues beyond the protocol. In proceedings of OHS Workshop 4.0 held at Hypertext '98 in Pittsburgh, June 20-24, 1998

[10] Grønbæk, K., Bouvin, N. O., and Sloth, L. *Designing Dexter-based hypermedia services for the World Wide Web.* In proceedings of Hypertext 97, Southampton, UK, April 6-11, 1997.

[11] Grønbæk, K., Hem, J., Madsen, O.L., & Sloth, L. Cooperative Hypermedia Systems: A Dexter-Based Architecture. Communications of the ACM, 37(2) 1994 pp. 64-75.

[12] Grønbæk, K., Sloth, L., Ørbæk, P.: Webvise: Browser and Proxy support for open hypermedia structuring mechanisms on the WWW. In proceedings of The Eighth International World Wide Web Conference, Toronto, Canada, ELSEVIER, Amsterdam, pp. 253-267, 11-14 May, 1999.

[13] Halasz, F. and M. Schwartz, The Dexter Hypertext Reference Model. Communications of the ACM, 1994. 37(2): p. 30-39.

[14] Hall, W. Davis, H. & Hutchings, G. Rethinking Hypermedia: The Microcosm Approach. Boston: Kluwer Academic Publishers 1996.

[15] Hansen, K.M., Yndigegn, C., and Grønbæk, K. Dynamic Use of Digital Library Material - Supporting Users with Typed Links in Open Hypermedia. In proceedings of the European Conference on Digital Libraries, September 21-24, Paris, France, 1999.Springer Lecture Notes in Computer Science no. 1696, pp. 254-273.

[16] Jühne, J. Jensen, A.T. & Grønbæk, K. *Ariadne: a Java-based guided tour system for the WorldWide Web*. In proceedings of The Seventh International World Wide Web Conference (WWW7) Brisbane, Queensland, Australia, 14 - 18 April 1998.

[17] Killough, R., & Leggett, J. Hypertext Interchange with the Dexter Model: Intermedia to KMS. TAMU-HRL 90-002. Hypertext Research Lab, Texas A&M University. August 1990.

[18] L. Carr et al. The Distributed Link Service: A Tool for Publishers, Authors and Readers. in Fourth International World Wide Web Conference : ``The Web Revolution''. Boston, Massachusetts, USA, 1995.

[19] Marshall, C.C. (1997) *Annotation: from paper books to the digital library* DL '97. Proceedings of the 2nd ACM international conference on Digital libraries, July 23-26, Philadelphia, PA. Pages 131-140

[20] Marshall, C.C. (1998) Making metadata: a study of metadata creation for a mixed physical-digital collection DL '98. Proceedings of the 3rd ACM international conference on Digital libraries, June 23-26, Pittsburgh, PA. Pages 162-171

[21] Maurer, H. Hyperwave – The Next Generation Web Solution. Addison Wesley: Harlow, UK. 1996.

[22] Meyrowitz, N. (1989). The missing link: Why we're all doing hypertext wrong. In Barrett, E. (ed.). The society of text: Hypertext, hypermedia and the social construction of information, pp. 107-114, MIT Press, Cambridge, Massachusetts, USA, 1989

[23] Nürnberg, P. J. Leggett, J. Schneider, E. R. & Schnase, J.L. Hypermedia operating systems: a new paradigm for computing. In *Seventh ACM Conference on Hypertext - Hypertext '96*. Washington, DC: ACM 1996.

[24] Nürnberg, P. J. Leggett, J. Schneider,. E. R. As we should have thought. *Proceedings of the Hypertext '97 Conference.* (Apr 6-11). Southampton, UK 1997.

[25] OHSWG. Open Hypermedia Systems Working Group WWW site (Sep 98). Website: **http://www.ohswg.org/**.

[26] Resource Description Framework (RDF) Model and Syntax Specification. W3C Recommendation 22 February 1999. **http://www.w3.org/TR/REC-rdf-syntax/**

[27] Röscheisen, M., Mogensen, C., & Winograd, T. (1994). *Shared Web Annotations as a Platform for Third-Party Value-Added Information Providers: Architecture, Protocols, and Usage Examples* (Technical Report CSDTR/DLTR). Stanford University, Stanford Integrated Digital Library Project, Computer Science Dept.

[28] Trigg, R.H. A Network-Based Approach to Text Handling for the Online Scientific Community. Ph.D. thesis, University of Maryland, TR-1346, 1983.

[29] Trigg, R.H. Guided Tours and Tabletops: Tools for Communicating in a Hypertext Environment. *ACM Transactions on Office Information Systems 6*(4) 1988 pp. 398-414.

[30] Whitehead, E. J. Jr. & Goland, Y. Y. WebDAV: A network protocol for remote collaborative authoring on the Web. In Proceedings of the Sixth European Conference of Computer Supported Cooperative Work, Copenhagen, Denmark, 1999, pp. 291-310.

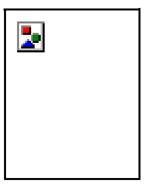
[31] Wiil, U. K. & Leggett, J. HyperDisco: Collaborative authoring and internet distribution. In *Hypertext '97*,. Southampton, England: ACM Press 1997 pp. 13-23.

[32] XML Linking Language (XLink) World Wide Web Consortium Working Draft 21 February 2000 **http://www.w3.org/TR/xlink.html**

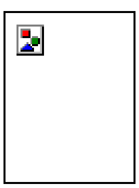
[33] XML Pointer Language (XPointer) W3C Working Draft 6 December 1999 **http://www.w3.org/TR/xptr**

[34] XML Path Language (XPath) Version 1.0 W3C Recommendation 16 November 1999 **http://www.w3.org/TR/xpath**

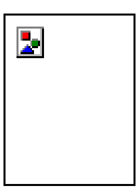
## Vitae



**Kaj Grønbæk** is professor at the Department of Computer Science, University of Aarhus, Denmark. He finished his master's degree in 1988 and his Ph.D. in 1991 both from the Dept. of Computer Science, University of Aarhus, Denmark. His research interests are: Hypermedia; Multimedia; Computer Supported Cooperative Work (CSCW); Cooperative Design (system development with active user involvement, cooperative prototyping); User interface design; object oriented tools and techniques for system development.



**Lennert Sloth** is research programmer at the Department of Computer Science, University of Aarhus, Denmark. He finished his master's degree in 1992 from the Dept. of Computer Science, University of Aarhus, Denmark. His areas of interests are: Design and development of hypermedia technologies. Development of graphical user interface libraries.



**Niels Olof Bouvin** is a Ph.D. student at Department of Computer Science, University of Aarhus, Denmark. His research interests include open hypermedia systems, Web augmentation, structural computing, and collaboration on the Web. Niels Olof Bouvin received his master's degree in 1996 from Department of Computer Science, University of Aarhus, Denmark.