

# Picture Metadata and its Associations: Using Web Technologies for Representing Semantics

Erik Wilde  
Computer Engineering and Networks Laboratory  
Swiss Federal Institute of Technology, Zürich

TIK Report 124  
January 2002

## Abstract

Web technologies today go far beyond simply enabling the creation of Web pages. XML and metadata formats based on it make it possible to manage metadata in a powerful and flexible way. In this paper, we describe the concept and the prototype of an application for the management of metadata for a specific domain, metadata associated with pictures. The goal of the paper is to highlight the benefits which result from employing Web technologies instead of proprietary data formats. While we think that both application developers as well as users could benefit from such an approach, we are aware that in many real-world cases other issues (such as the ability to bind users to a certain product) also play an important role. Nevertheless, in this paper we show that open and well-documented technologies not only can make software development easier, but also open up synergies between standards-compliant products. While the prototype we present in this paper is not sophisticated enough to be released to the general public, we hope that software vendors will consider incorporating some of the concepts introduced in this paper.

**Keywords:** Metadata, XML, Topic Maps, XLink, Browser, PicArc

## 1 Introduction

The *Semantic Web* [2] has been the focus of attention for some time now. While a lot of research efforts go into technology helping to automate the process of creating and interpreting semantic information, for a long time to come humans will remain vastly superior to machines when it comes to rich and versatile semantics. Even though we are considering this paper to be relevant to the Semantic Web vision, we explicitly rely on human intelligence to create and manage semantic information. We simply provide a way to represent and manipulate this information efficiently and flexibly using existing and already established Web technologies. Throughout this paper, we will use the term *metadata* whenever we refer to data about data, with metadata being the model through which semantics can be represented.

Many Web technologies can be viewed from two perspectives: Either as data formats to be used within applications (making software development easier by re-using existing software components), or as exchange formats for exchanging data between applications using

standardized protocols and data formats (thus creating software that can easily be used together with other software because it has been built using open standards). In this paper, we present an application which, in principle, could have been based on any internal format (even though we do use numerous established Web standards), but which has been heavily inspired by current and upcoming Web standards, and thus could be easily adapted to support access through these formats, for example enabling smart Web agents to discover information by interpreting metadata.

The goal of this paper is to present a possible model for building applications, and to present an application based on this model, which can be easily adapted to upcoming standards for metadata. Starting from an application scenario (which is presented in Section 2), we look at what can be done with currently available and upcoming Web standards (Section 3), and then describe the implementation of a prototype based on existing software components for these Web standards (Section 4). This usage of existing software components not only significantly reduced the time we needed to implement the prototype, but also enabled us to react quickly if new standards were being published or existing standards evolved to support more sophisticated features. Future work (Section 5) and concluding remarks (Section 6) finally give an overview of what we are planning to do, and what we have achieved so far.

## 2 Application Scenario

Starting from the general observation that content is only one side of the coin, and the links between resources being the other (very important) one [33, 34], we wanted to develop a prototype for a highly interlinked set of information resources. These resources should also be described by extensive metadata, because we believe that linking becomes much more powerful if the links as well as the linked resources are characterized with machine-readable semantics.

Furthermore, we wanted to have a set of interlinked resources which are embedded into the Web by linking not only to each other, but also to generally available Web resources. A very good example for this type of resources are pictures, which can have a large number and variety of metadata associated with them:

- *Classification Topics*

This is a very broadly defined term, because generally speaking anything can be regarded as classification. We see classification topics as topics which are user-specific and categorize pictures according to certain aspects, which in most cases are orthogonal to each other. Examples of classification topics are:

- Events (wedding, race, party, concert, ...)
- Picture types (funny, beauty shots, documentation, ...)

Users may define as many classification topics as they need. Naturally, classifying pictures according to a large set of classification topics is more time-consuming, but it also makes pictures more easily accessible through all these different topics.

- *Content-related topics*

While classification topics are not directly linked to a picture's content, content-related

topics describe pictures according to their content. Possible examples for this type of topic are:

- Foreground (person, building, nothing, ...)
- Background (same possibilities as with foreground)

It is clear that classification and content-related topics are not always easy to separate. We make this differentiation by separating the topics into the ones which require knowledge of the circumstances under which a picture was taken (classification topics), and the rest which can be derived from simple looking at a picture using average world-knowledge (content-related topics).<sup>1</sup>

We assume classification and content-related topics to be orthogonal in the sense that in some retrieval scenarios only classification may be required (“I need all the pictures from a particular birthday party”), while in others content-related topics may be more useful (“I need all pictures with a particular person in the foreground and mountains in the background”).

- *Links to external resources*

In many cases, classification as well as content-related topics may not only exist as metadata, but also may be represented by resources on the Web, which can be addressed through a URI<sup>2</sup>. It would be very useful to include links to these resources along with the metadata (eg, a link to a website providing information about an event which is a classification topic, such as a conference).

Theoretically, anything can be identified by a URI, as long as a well-defined scheme for identification exists. However, for the near future, things like for example people will not be identified by URIs directly, but by URIs pointing to their Web pages (using the well-known `http` scheme), mailboxes (using the `mailto` scheme [14]), or phone numbers (using the `tel` scheme [30]).

- *Technical information*

Pictures may have associated with them technical information about the picture itself, such as the camera, the lens, filters, shutter speed, aperture, film type and scanner settings (if the picture was originally analog and subsequently digitized), light conditions, flashlight usage, and any other information which might be relevant for the photographer.

It can be seen that picture metadata can be very diverse and complex, and our goal was to create a conceptual framework and a prototype which can be used to manage and leverage this metadata. In order to find out the current state of the art in picture metadata handling, we assessed a number of commercially available programs for picture archives.

---

<sup>1</sup>This leads to an interesting discussion of how much of the classification could actually be derived by automatically analyzing the pictures, which we discuss in more depth in Section 4.3.

<sup>2</sup>URIs [1] are the way how Web resources are addressed. The concept is described in a recently published W3C note [9].

## 2.1 Existing Systems

Digital cameras have been a big success in the camera market in the last years, and all market predictions indicate that this trend will continue to take over the camera market from traditional film-based cameras. Many digital cameras are shipped with software for downloading the pictures from the camera to a computer, and often also software to manage the pictures on the computer. In a survey of a number of these programs, we discovered that none of them had substantial metadata handling capabilities. Some offer free text entry and have search facilities for these text fields, but none supported an elaborated concept for handling metadata.

The programs that we used for our evaluation of existing tools for managing pictures are *photodex compupic pro*, *Firehand Amber 2000*, *Casio Photo Loader*, *X-Equite Smart Pix Manager*, *Epson Film Factory*, *Progressive Logic PixSee*, *Pink Mouse Image Organizer*, *ULead Photo Impact Album*, *mediatracer.com*, *X-Equite MegaView*, and *X-Equite EZ-Pix*. An online version of the evaluation matrix can be found at <http://dret.net/netdret/docs/tikrep124-matrix>. While some of these programs have very neatly designed and sophisticated interfaces, the data model for managing the pictures in all cases is surprisingly simple, leaving us to wonder how useful these programs are for collections of thousands or tens of thousands of pictures, numbers which can be easily reached by industrious photographers.

This observation showed us that the application scenario of a picture archive not only was an excellent example for looking into the benefits and potential problems of linking and metadata, but also gave us the opportunity to create a data model which was superior to all the existing applications that we evaluated.

## 3 Conceptual Framework

Using the application scenario as the starting point, we investigated different possible ways to implement such an application. Based on the data model described in Section 3.1, we decided on a set of technical foundations that we thought would be useful for us, both in terms of re-use of existing software components, and in terms of interoperability with other applications. In Section 3.2, we describe the technical foundations of our concept. In Section 3.3, we describe the intended application area of our concept.

### 3.1 Data Model

The conceptual foundation of our work is the idea of semantically rich link information, which can be used to describe the associations between a set of resources. In terms of semantic concepts for the Web, currently there are two main candidates, W3C's *Resource Description Framework* [21,6], and ISO's *Topic Maps* [16]<sup>3</sup>. Both standards aim at representing semantic information about Web resources, but they do so in a slightly different way. It is not yet clear which standard will prevail, but currently RDF seems to have the bigger following (and a significant advantage since it is a W3C activity).

Since we intended the picture archive (called *PicArc*) to be a case study of our general model of a linkbase, we adopted the model that we used for the XLinkbase project. This model is a slight variation of the Topic Map model, avoiding its well-known weaknesses, which have been described by RATH [28] (the two most serious being the inability to make assertions

---

<sup>3</sup>Even though ISO Topic Maps are based on SGML, *XML Topic Maps (XTM)* [26] specifies an XML variant.

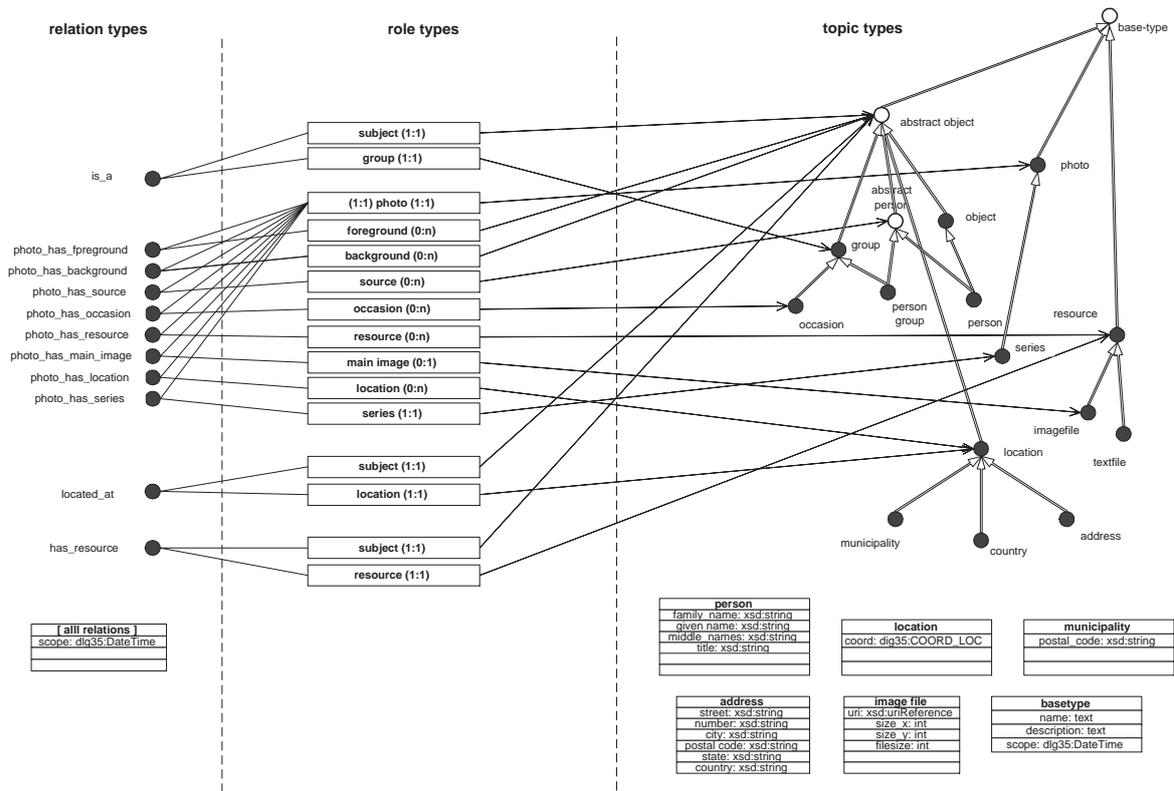


Figure 1: Picture archive data model

about the resources participating in associations, and the inability to clearly separate the schema of a Topic Map from instances).

The data model of the picture archive, shown in Figure 1<sup>4</sup>, thus is similar to the data model of Topic Maps, but is not strictly adhering to it. The schema for the picture archive, which is based on this data model, contains predefined *association types*, *role types*, and *topic types*, and this set of predefined types cannot be extended. The PicArc types have the following properties:

- *Association types* are always binary, which means that they have exactly two role types associated with them.

An example of an association type is *located\_at*, which has associated with it the two role types *subject* and *location* and can be used to specify spatial relations, such as “The Eiffel tower is in Paris”.

- *Role types* always have associated with them a *topic type*, and only topics of this or derived types may occur in the particular role.

<sup>4</sup>In this figure, thin lines connect related types, bold arrows indicate inheritance between topic types (with subtypes inheriting the attributes of supertypes), and white circle types are abstract types, meaning that no instances of these types are can be created.)

For example, the *location* role type requires an topic of type *location* or a derived type, and in the example from above, “Paris” is of type *municipality*, which is a subtype of *location*.

- *Topic types* are using a type hierarchy, and this hierarchy is used to check whether a topic may be used with a particular role type. The topic type hierarchy allows multiple inheritance.

For example, the *location* topic type has the subtypes *municipality*, *country*, and *address*, and is a subtype of the *abstract object* type.

All types have attributes associated with them, and these attributes are used to further describe the instances of the respective type. Some types do not have attributes, but all topic types do have attributes and sometimes a large number of them.

Users create their picture archive by instantiating topics and connection them through associations (using roles), and in effect this creates a semantic net which describes the particular area of interest for a particular user. Our data model is defined in a way such that roles are related to certain associations (ie, a given association requires two roles to be present), and topics are related to roles (ie, a topic of a given type or subtype must be used within a particular role). This way, the creation of PicArc data is more restricted than the general Topic Map model, but we believe that this restriction is useful in keeping a picture archive clearly structured.

### 3.2 Technical Foundations

Based on the PicArc data model, the following standards also are important to us: *Extensible Markup Language (XML)* [5] as the syntax which is used to represent application data. The *Extensible Linking Language (XLink)* [11] as the syntax for exporting linking information. *XSL Transformations (XSLT)* [7]<sup>5</sup> as the language to manipulate and transform application data. Rather than programming certain operations in Java, we use XSLT programs, which are better suited for transforming XML than a general-purpose programming language. There certainly are performance problems associated with this approach, in particular for large picture archives, but our main goal was to explore the data model and the possible implementation support by existing software components.

Furthermore, we use the *Hypertext Markup Language (HTML)* [27], or, strictly speaking, the *Extensible Hypertext Markup Language (XHTML)* [24, 25], as the language for exporting data from the application in a format which is understood by many clients. This list of Web technologies demonstrates that even though we were about to create a specialized application, we could benefit from many existing technologies and software implementing it.

### 3.3 Application Area

Since our goals were to create a metadata-centered application for picture archiving, we not only evaluated other software, but also tried to find other activities in this area. Collecting pictures is maybe one of the most universal activities, and even though digital images are

---

<sup>5</sup>Even though the development of XSLT 1.1 [8] (the version of XSLT that we are using) has been stopped, it is very likely that the features proposed for XSLT 1.1 will be supported by XSLT 2.0 [23]. We therefore think that we can easily upgrade to XSLT 2.0 as soon as it is stable and XSLT implementations start to support it.

becoming more popular, most people still have many film-based pictures. It was our goal to create a model which not only could be used to efficiently categorize new pictures, but which also could be used to create metadata about old pictures. Our focus, however, was on newly created pictures, and there is one standard which is of particular interest in this area.

The *DIG35 Metadata for Digital Images* [12] standard describes a lot of possible semantics to be associated with a picture. However, while DIG35 contains a lot of interesting ideas what kind of metadata would be appropriate (eg, one interesting idea is to include coordinates for identifying the place where the picture was taken), its biggest disadvantage is that there is no concept for a web of interlinked semantics connected by associations. In contrast, we think that this concept of creating a web is essential for making a large set of pictures accessible and manageable. DIG35, however, would be ideally suited as an exchange format for single images and could be easily generated from the metadata information contained in a web of image and metadata resources.

## 4 Implementation

Starting from the foundations described in the preceding section, we wanted to create a prototype which would be appropriate for us to be used to test the concepts and the way they should be implemented. The current PicArc prototype is implemented in Java, using Swing GUI components, and the XML technologies support (ie, an XML parser and an XSLT processor) is provided by the Apache Xerces and Xalan products.

Figure 2 shows a screen shot of the index view of the main application window. The left pane shows interface elements to browse through all topics which are inside the archive. The screen shot shows the list of all “series” topics, which are used to describe series of pictures (eg, series of pictures taken at a certain event or at a certain location). In the right pane, all pictures which are associated with the selected series are shown as thumbnails, and by clicking on a thumbnail it is possible to show the actual picture. The picture itself maybe stored locally or remotely, from the data model point of view it is simply identified by a URI (see the “image file” topic type attributes in Figure 1).

By using the index, it is possible to browse through the picture archive using the topics as navigational aid. However, using this view, it is not possible to see the associations between topics, which also may be very useful for browsing the archive. To see the associations of a topic, the topic editing view must be used, which is shown in Figure 3.

This view shows all information which is relevant for a given topic, most importantly its attributes and its associations. It also shows all thumbnails of pictures which are relevant for this topic. In this example, it is only one set of pictures, but if individual pictures have different associations with this topic (such as, for a topic representing a person, “the person is in the foreground” and “the person made this picture”), then the thumbnail pane groups the thumbnails according to the different role that the topic plays in the associations with the pictures.

A third view is the query view, which makes it possible to construct queries which are then executed, resulting in a set of topics which satisfy the query. Strictly speaking, we currently implement a “find” rather a “query” functionality, because there is no way how subsets of results can be recombined. However, even with this functionality the query view can be used to form complex boolean searches, such as “find all pictures with have been taken by Alex or Erik, and which have a bridge in the background”.

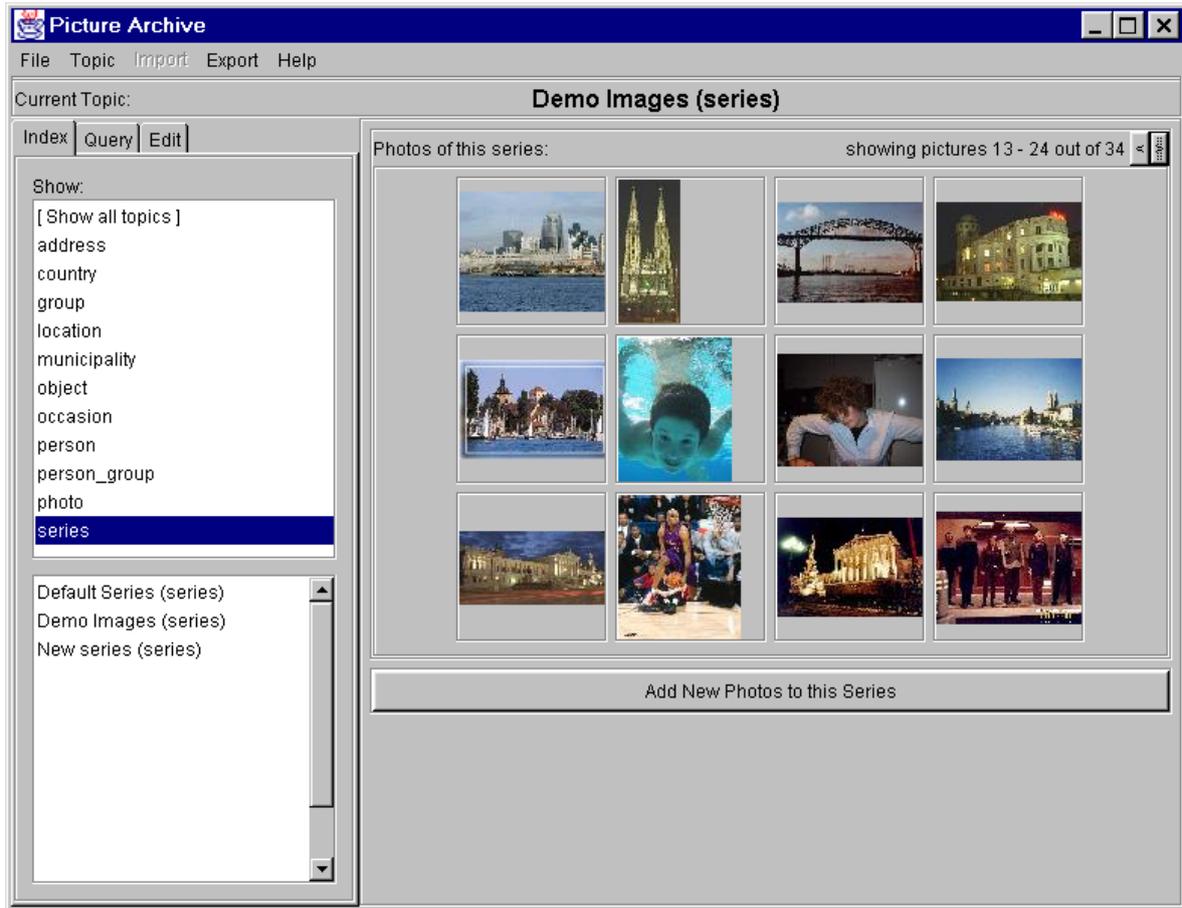


Figure 2: PicArc general application window (index view)

This way of navigation is sufficient to be able to browse through the picture archive, but in the long term it is desirable to have a more elaborated type of interface, such as hyperbolic trees [20] or other more graphically oriented ways of information visualization.

#### 4.1 Picture Handling

While PicArc is designed to manage the metadata associated with pictures, it is not designed to enable picture manipulation (such as resizing, cropping, or filtering) in any way. Thus, PicArc's only picture handling ability (next to its metadata browsing functionality) is the display of pictures. Currently, PicArc only supports ISO's *Joint Picture Experts Group (JPEG)* [15] format, or, more specifically, the *JPEG File Interchange Format (JFIF)* [13], but extending PicArc's support to more sophisticated formats such as the upcoming *JPEG 2000* [17] standard would be easy by including the appropriate software components.

However, while picture display is the most important task in a picture archive, it is also important to display summary pages using picture *thumbnails* (scaled down versions of the original picture, as shown in Figures 2 and 3). PicArc can be set to either calculate thumbnails on demand (ie, retrieving the original and scaling it down whenever a thumbnail needs to be displayed), or to cache thumbnails (see Figure 4). Caching thumbnails in almost

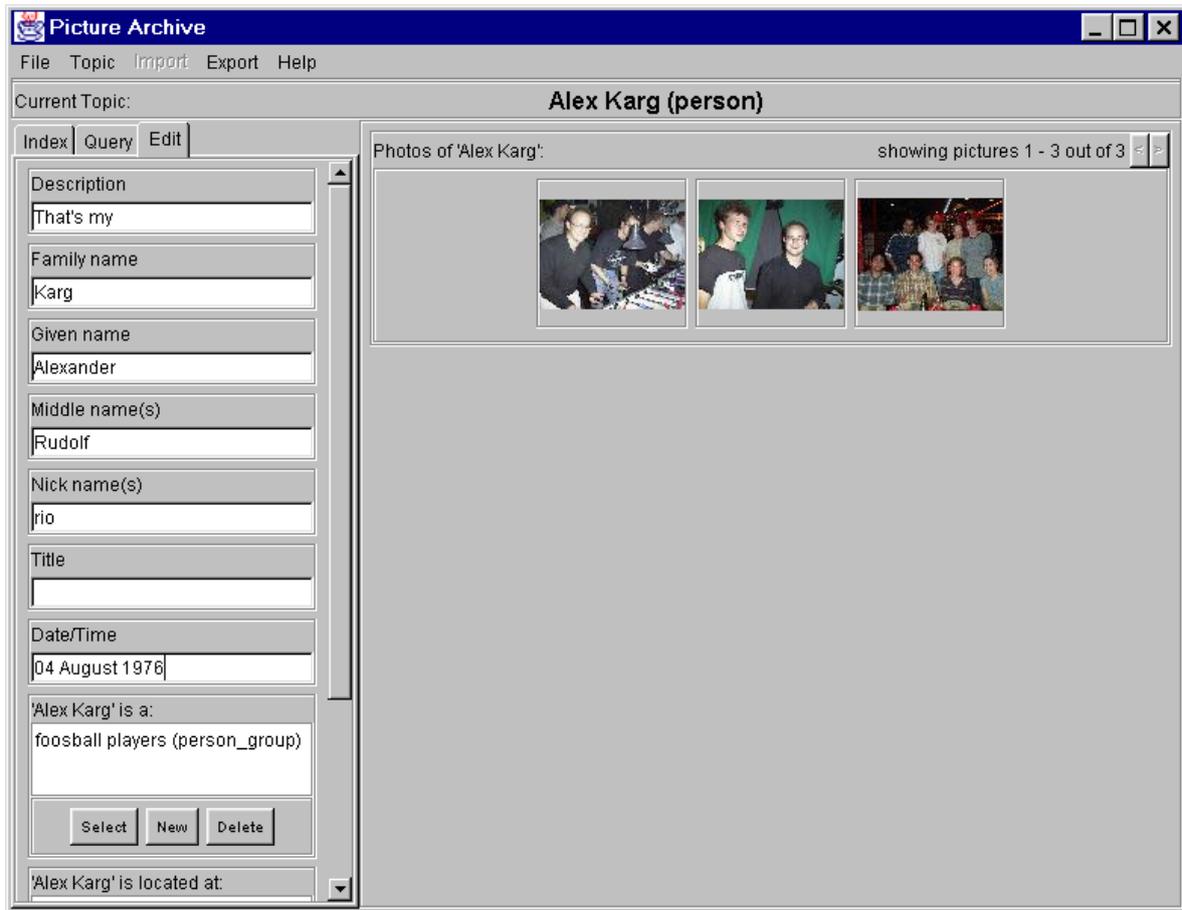


Figure 3: PicArc general application window (topic editing view)

all circumstances is the only practical way of working with picture archives, because pictures in PicArc archives can be anywhere, including remote servers accessed over possibly slow network connections. And even if the pictures are on the same computer as PicArc, the generation of thumbnails — even on fast machines such as GHz PCs — consumes a lot of computing time. Therefore, it is advisable to always use the cache, which can be flushed on demand, so that any modifications of pictures are taken into account when calculating new thumbnails.

## 4.2 Import and Export

In many applications today, import and export facilities are provided. However, often import and export operations are crippling application data because they fail to fully support the application's internal data model. One of the main goals of PicArc is to avoid such a lock-in of user data by fully exposing the internal data model through export operations. PicArc uses XML as file format, and the DTD and its semantics are documented and available, so that nothing of PicArc's internal data is hidden from users. While users can simply take PicArc's XML and process it in any way they want, they can also supply an externally generated XML file (which of course must conform to PicArc's schema) and run PicArc with it. For examples,

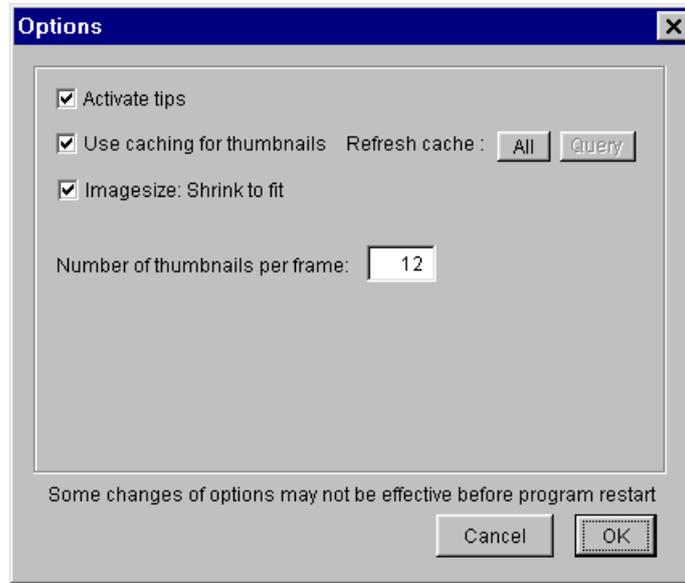


Figure 4: PicArc options dialog window

users having their own semantic network in alternative representations such as RDF or ISO Topic Maps could use transformation to translate their data into a PicArc archive, and then start using PicArc’s features to organize a picture archive using their semantic network.

In addition, PicArc also supports the export of archives or subsets of archives to HTML (see Figure 5, which also shows that some of the more advanced options are not yet implemented). Exporting archives as HTML is extremely important because many users do not have PicArc installed, but may also look inside an archive. Furthermore, the HTML export functionality can be used to publish the content of a PicArc archive on the Web and thus make it available to a large audience. HTML export can be done in a variety of ways, in particular with respect to how much of the metadata associated with the pictures should be exposed. The HTML export dialog makes it possible to control PicArc’s way of generating HTML. Internally, HTML is generated by running an XSLT program which transforms the internal PicArc XML data into a set of HTML pages.

### 4.3 Automatization

Clearly, creating a richly described picture archive takes a lot of work. Currently, all this work is done manually, because the PicArc program does not support any type of automatic picture categorization. There are some shortcuts for assigning links to a whole series of pictures (which is one of the main motivations for why using the “series” topic type often is a good idea), but this does not change the fact that using PicArc requires a lot of manual work.

We do not plan to enhance PicArc in a way which would include any automatic features for picture categorization, but using PicArc’s open XML data format and its export and import features, it is pretty easy to add automated services by using additional software:

- *Picture Information*

Many digital cameras today store some metadata in additional fields of the image data

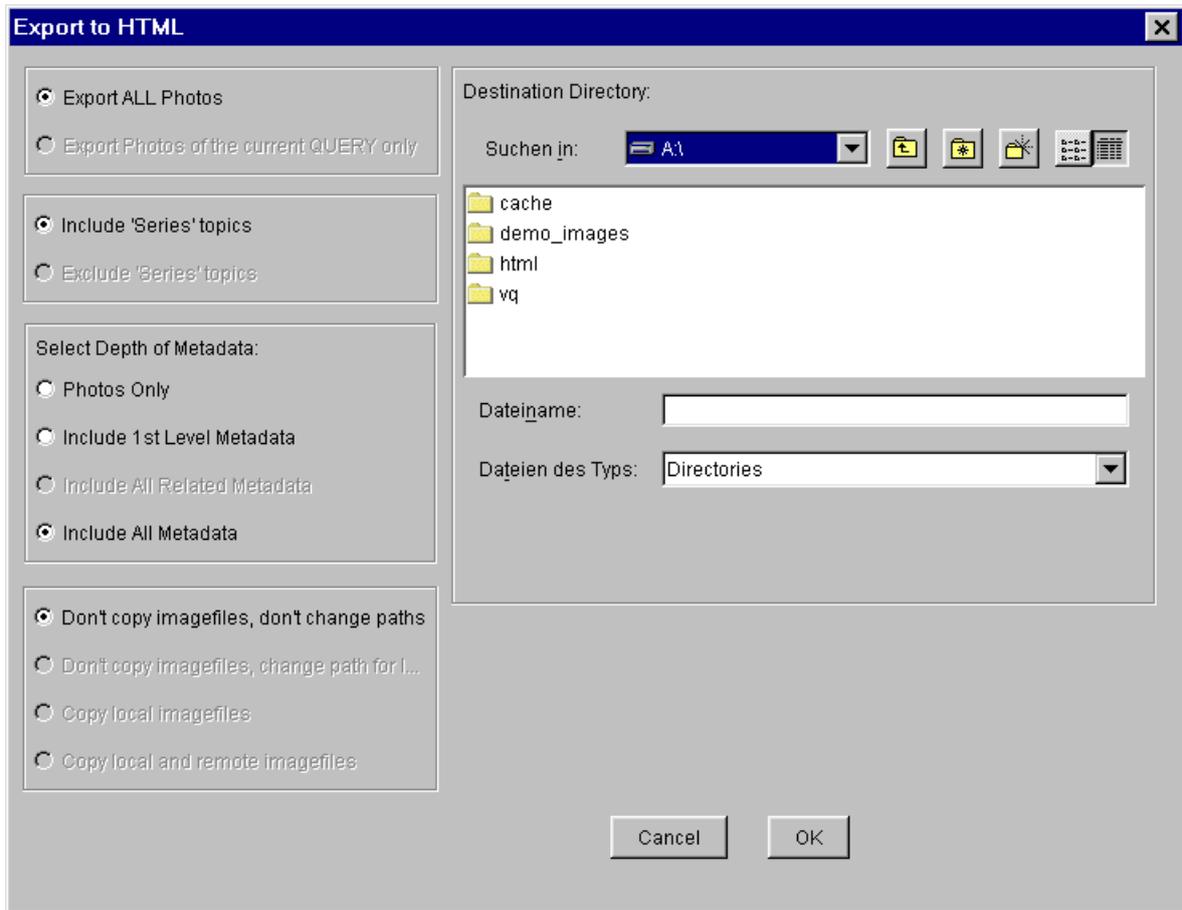


Figure 5: PicArc HTML export dialog window

format (often as special parts of the popular JPEG [15] picture format, which is allowed by the JPEG standard and standardized as EXIF [18]), and instead of just using the picture data from such an image, the metadata (often information such as shutter speed and aperture) could be extracted from the image data and used as metadata to be entered in PicArc.

This would require some kind of preprocessing, with the preprocessor extracting the metadata from the image data, and then expressing this metadata in XML and importing it into PicArc. This would also be a good way to harmonize metadata, because metadata from different cameras is often stored in different ways within the image data, but would be stored in the openly accessible PicArc format, regardless of the camera it has been created with. Furthermore, it makes more much sense to expose the metadata outside of the picture data itself, where it is easier to manage and to query.

- *Picture Interpretation*

In a more advanced scenario, PicArc could export all its metadata, which could then be used by a complex picture interpretation program for extracting more information from the images, for example concluding that if a picture shows mountain (part of the metadata), and has a lot of white tones in it (part of the actual picture data), then

it probably is a snow-covered mountain, which could then be stored as automatically generated metadata. This is a very simple example, but it shows the idea of applications external to PicArc extending the metadata with new information, which could then be imported into PicArc again and perhaps reviewed for accuracy, if required.

Clearly, picture interpretation is a very complex example, and we do not think that automatic interpretation can be successful without metadata. As we already mentioned in Section 2 discussing classification and content-related topics, this is the basic difference between these two topic types. For example, a picture interpretation program might clearly recognize a structure as being the Eiffel tower, but without access to the metadata describing whether this picture has been taken in Paris or in Disneyland, it would be impossible to actually decide the question whether this was the real Eiffel tower, or another tower simply looking exactly like it.

- *Ontology Management*

Currently, PicArc uses its own model for the topics that a user defines. Different users will probably define very different topics, but basically all these models are *ontologies*, ie representations of how these particular users structure their information (or, more broadly speaking, see the world). PicArc could be extended to include ontology import and export features, which would make it possible to merge PicArc's topic model with other ontologies, for example RDF graphs, to create combined data structures of existing ontologies and PicArc's picture archives.

PicArc is a prototype for demonstrating the feasibility of using Web standards for application scenarios which are not tightly connected to the Web. It also shows that using standards in a well-defined and well-documented way can open up many opportunities for interworking with other applications, if all of them use the same foundation. In particular, well-defined data formats and import/export features which do not cripple application data are essential for building applications that are open and flexible.

## 5 Future Work

Currently, our data model (the XLinkbase model) is based on an XML *Document Type Definition (DTD)*. It would be interesting to reformulate it using *XML Schema* [29, 3], which offers much better facilities for specifying schemas, in particular a powerful type system. Using XML Schema and a schema-validating parser for importing data would eliminate many of the errors which are possible today, because our prototype does not rigorously check all data types, and the current DTD validation is very weak in detecting type errors. Furthermore, using XML Schema types would enable us to fully take advantage of the features of the upcoming XSLT 2.0, which will probably support the XML Schema types.

Unfortunately, currently no available browser supports XLink, but as soon as XLink browsers become available, we will supplement the HTML export functionality with the ability to export XML with XLinks. This new export format (depending on the link presentation of the browser) will enable us to support a better way of visualizing picture archives than the current HTML version. Even the newest releases of the major Browsers<sup>6</sup> only support XLink *simple links*, and it is not yet clear when the first browser will fully support

---

<sup>6</sup>At the time of writing, these are *Internet Explorer 6.0* and *Navigator 6.1*.

XLink. Furthermore, when browsers start to support *XML Pointer Language (XPointer)* [10], we plan to explore the ways in which we could further improve the browsing experience using this standard. However, as long as the presentation of XLinked documents [31] is not clearly defined, it will be hard to fully implement XLink and XPointer in a browser without running the risk of making decisions which could later be proven wrong by an upcoming standard for XLink presentation.

Currently, we implement our own query mechanism. As soon as *XML Query Language (XQuery)* [4, 22] and software implementing it becomes available, however, we will re-work the query interface by basing it on the underlying XQuery processor, thus supporting a wider range of queries, and re-using existing software for executing these queries.

Currently, PicArc's HTML export feature (as described in Section 4.2) is fairly basic. We plan to extend this feature by supporting a richly interlinked topic presentation similar to the one shown in <http://wildesweb.com/glossary/>. As with most aspects of PicArc, we can simply do that by improving the XSLT which is responsible for post-processing the XML resulting from the selected filter(s) of the export operation.

## 6 Conclusions

The concept of a metadata-centered picture archive and the PicArc prototype for implementing such an archive have shown that existing Web technologies can be used to significantly decrease the cost associated with software development. Furthermore, the utilization of software components and the support of standardized data formats make it extremely easy to extend the prototype with new functionality, and to create software that can be used in close cooperation with the prototype.

PicArc is meant to be a demonstration of the benefits of standards-based software, and also as the proof of concept for the hypothesis that metadata is one of the key issues for making content more accessible. We hope that some of the concepts demonstrated with PicArc will find their way into software products. We also hope that work like this will help promoting the idea of open data formats, with applications being marketed because of their superior data model, functionality and interface design<sup>7</sup>, rather than their using of proprietary and undocumented data formats.

## 7 Acknowledgements

The PicArc picture archive is part of the XLinkbase project [32], which aims at implementing a general-purpose linkbase which is focused on capturing links and semantic information associated with them. YVES LANGISCH and MANFRED MEYER have been working on the implementation of the server side, while SIMON KÜNZLI and PETER ZBERG have implemented a prototype of a general-purpose client. In his Master's Thesis [19], ALEXANDER KARG has developed the special-purpose client PicArc described in this paper, and he also kindly provided the screen shots presented in this paper. MARCO AMREIN provided valuable input while discussing the general XLinkbase information model.

---

<sup>7</sup>It should be rather easy to surpass the PicArc *prototype* in terms of functionality and interface design, but this is exactly our point, because we believe that PicArc's *data model* could serve as the foundation for very sophisticated and powerful picture archival software.

## References

- [1] TIM BERNERS-LEE, ROY T. FIELDING, and LARRY MASINTER. Uniform Resource Identifiers (URI): Generic Syntax. Internet draft standard RFC 2396, August 1998.
- [2] TIM BERNERS-LEE, MARK FISCHETTI, and MICHAEL DERTOUZOS. *Weaving the Web*. Harper-Collins, San Francisco, California, September 1999.
- [3] PAUL V. BIRON and ASHOK MALHOTRA. XML Schema Part 2: Datatypes. World Wide Web Consortium, Recommendation REC-xmlschema-2-20010502, May 2001.
- [4] SCOTT BOAG, DON CHAMBERLIN, MARY F. FERNÁNDEZ, DANIELA FLORESCU, JONATHAN ROBIE, JÉRÔME SIMÉON, and MUGUR STEFANESCU. XQuery: A Query Language for XML. World Wide Web Consortium, Working Draft WD-xquery-20011220, December 2001.
- [5] TIM BRAY, JEAN PAOLI, C. M. SPERBERG-MCQUEEN, and EVE MALER. Extensible Markup Language (XML) 1.0 (Second Edition). World Wide Web Consortium, REC-xml-20001006, October 2000.
- [6] DAN BRICKLEY and R. V. GUHA. Resource Description Framework (RDF) Schema Specification. World Wide Web Consortium, Candidate Recommendation CR-rdf-schema-20000327, March 2000.
- [7] JAMES CLARK. XSL Transformations (XSLT) Version 1.0. World Wide Web Consortium, Recommendation REC-xslt-19991116, November 1999.
- [8] JAMES CLARK. XSL Transformations (XSLT) Version 1.1. World Wide Web Consortium, Working Draft WD-xslt11-20010824, August 2001.
- [9] TONY COATES, DAN CONNOLLY, DIANA DACK, LESLIE L. DAIGLE, RAY DENENBERG, MARTIN J. DÜRST, PAUL GROSSO, SANDRO HAWKE, RENATO IANNELLA, GRAHAM KLYNE, LARRY MASINTER, MICHAEL MEALLING, MARK NEEDLEMAN, and NORMAN WALSH. URIs, URIs, and URNs: Clarifications and Recommendations 1.0. World Wide Web Consortium, Note NOTE-uri-clarification-20010921, September 2001.
- [10] STEVEN J. DEROSE, EVE MALER, and RON DANIEL. XML Pointer Language (XPointer) Version 1.0. World Wide Web Consortium, Candidate Recommendation CR-xptr-20010911, September 2001.
- [11] STEVEN J. DEROSE, EVE MALER, and DAVID ORCHARD. XML Linking Language (XLink) Version 1.0. World Wide Web Consortium, Recommendation REC-xlink-20010627, June 2001.
- [12] DIGITAL IMAGING GROUP, INC. DIG35 Specification — Metadata for Digital Images (Version 1.0). Technical report, Millbrae, California, August 2000.
- [13] ERIC HAMILTON. JPEG File Interchange Format (Version 1.02). Technical report, C-Cube Microsystems, Milpitas, California, September 1992.
- [14] PAUL E. HOFFMAN, LARRY MASINTER, and JAMIE ZAWINSKI. The mailto URL scheme. Internet proposed standard RFC 2368, June 1998.
- [15] INTERNATIONAL ORGANIZATION FOR STANDARDIZATION. Information technology — Digital compression and coding of continuous-tone still images. ISO/IEC 10918, 1999.
- [16] INTERNATIONAL ORGANIZATION FOR STANDARDIZATION. Information technology — SGML Applications — Topic Maps. ISO/IEC 13250, 2000.
- [17] INTERNATIONAL ORGANIZATION FOR STANDARDIZATION. Information Technology — JPEG 2000 Image Coding System. ISO/IEC 15444, January 2001.
- [18] JAPAN ELECTRONIC INDUSTRY DEVELOPMENT ASSOCIATION. Digital Still Camera Image File Format Standard (Exchangeable Image File Format for Digital Still Cameras: EXIF) — Version 2.1, June 1998.
- [19] ALEXANDER KARG. Konzept und Implementierung eines Programmes für die Bildarchivierung und -verwaltung. Master's thesis, Computer Engineering and Networks Laboratory, Swiss Federal Institute of Technology, Zürich, Switzerland, August 2001.

- [20] JOHN LAMPING, RAMANA RAO, and PETER PIROLI. A Focus+Context Technique Based on Hyperbolic Geometry for Visualizing Large Hierarchies. In *CHI '95: Proceedings of the ACM Conference on Human Factors and Computing Systems*, pages 401–408, Denver, Colorado, May 1995. ACM Press.
- [21] ORA LASSILA and RALPH R. SWICK. Resource Description Framework (RDF) Model and Syntax Specification. World Wide Web Consortium, Recommendation REC-rdf-syntax-19990222, February 1999.
- [22] ASHOK MALHOTRA, JIM MELTON, JONATHAN ROBIE, and NORMAN WALSH. XQuery 1.0 and XPath 2.0 Functions and Operators Version 1.0. World Wide Web Consortium, Working Draft WD-xquery-operators-20011220, December 2001.
- [23] STEVE MUENCH and MARK SCARDINA. XSL Transformations Requirements Version 2.0. World Wide Web Consortium, Working Draft WD-xslt20req-20010214, February 2001.
- [24] STEVEN PEMBERTON. XHTML 1.0: The Extensible HyperText Markup Language. World Wide Web Consortium, Recommendation REC-xhtml1-20000126, January 2000.
- [25] STEVEN PEMBERTON. XHTML 1.0: The Extensible HyperText Markup Language (Second Edition). World Wide Web Consortium, Working Draft WD-xhtml1-20011004, October 2001.
- [26] STEVE PEPPER and GRAHAM MOORE. XML Topic Maps (XTM) 1.0. TopicMaps.Org Specification xtm1-20010806, August 2001.
- [27] DAVE RAGGETT, ARNAUD LE HORS, and IAN JACOBS. HTML 4.01 Specification. World Wide Web Consortium, Recommendation REC-html401-19991224, December 1999.
- [28] HANS HOLGER RATH. Topic Maps: Templates, Topology, and Type Hierarchies. *Markup Languages: Theory & Practice*, 2(1):45–64, 2000.
- [29] HENRY S. THOMPSON, DAVID BEECH, MURRAY MALONEY, and NOAH MENDELSON. XML Schema Part 1: Structures. World Wide Web Consortium, Recommendation REC-xmlschema-1-20010502, May 2001.
- [30] ANTTI VÄHÄ-SIPILÄ. URLs for Telephone Calls. Internet proposed standard RFC 2806, April 2000.
- [31] NORMAN WALSH. XML Linking and Style. World Wide Web Consortium, Note NOTE-xml-link-style-20010605, June 2001.
- [32] ERIK WILDE. URI scheme for GSM Short Message Service. Internet draft draft-wilde-sms-uri-01, January 2002.
- [33] ERIK WILDE and DAVID LOWE. From Content-Centered Publishing to a Link-Based View of Information Resources. In *Proceedings of the 33rd Hawaii International Conference on System Sciences*, Maui, Hawaii, January 2000. IEEE Computer Society Press.
- [34] ERIK WILDE and DAVID LOWE. *XML, XLink, and XPointer: A Practical Guide to Web Hyperlinking and Transclusion*. Addison-Wesley, Reading, Massachusetts, to be published in 2002.