On Indexing in Digital Libraries: Cooperation, Personalization, and Evolution

Claudia Niederée, Ulrike Steffens,
Hans-Werner Sehring, Florian Matthes, Joachim W. Schmidt
Software Systems Institute,
Technical University Hamburg-Harburg, Hamburg, Germany
{c.niederee,ul.steffens,hw.sehring,f.matthes,j.w.schmidt}@tu-harburg.de

June 16, 1998

Abstract

In databases and traditional paper documents, an *index* is often tightly coupled with the information resources being indexed. Digital libraries require a generalized notion of indexing that is suitable for distributed and heterogeneous collections. The main contribution of the generalization described in this paper is to introduce an additional intermediate layer between the navigational structure of the index and the information resources. This layer is populated by so called *index elements* providing decision support and access information concerning the underlying information resources.

The generalized indexing concept provides a framework for the definition of generic services on indexes and their index elements. Different categories of services supporting the characteristics of the generalized concept are discussed in this paper with a special focus on the evolution of indexes, the personalization of indexes, and on cooperative work supported by indexes.

As a concrete example for a cooperative service on index elements a tool for merging bibliographic index elements is introduced.

1 Introduction

The construction and use of indexes is a well accepted methodology to improve access to information resources of various kinds and to structure collections of such resources. Accelerating the access via primary or secondary keys, indexes come up in database systems providing additional access paths for the query optimizer. In the area of information retrieval, large indexes are a crucial prerequisite for retrieval. The use of indexes is of course not restricted to electronic media. Indexes in paper documents like books are a well-known method to provide alternative entry points to the content.

However, in all the mentioned indexes, there is a tight coupling between the information resources and the index to improve and to accelerate access, degrading indexes to mere auxiliary components. The information resources and the index information are typically managed by the same provider. Considering for example a book, there are only two parties involved: the reader of the book using the index for quick access and the publisher of the book providing the information resources, namely, the content of the book as well as the index. This tight coupling of indexes and information resources is also a characteristic of database indexes, where the database system manages the data records as well as the index.

The situation is somewhat different in digital libraries, where large collections of heterogeneous information resources from distributed information providers have to be structured by an index.

- Index and information resources may reside at different places making the actual step from the index to the information resource a potentially time-consuming and/or expensive one.
- There is no longer a single information provider for the entire indexed collection of resources.

- An index in a distributed environment often has to integrate selected material from several different resource collections.

These characteristics lead to a looser coupling between indexes and information resources. As a consequence, indexes may be provided by third parties as a value-adding service, which structures and annotates the existing material according to the needs of a specific user group or application domain.

These considerations lead to the generalization of the indexing concept as proposed in this paper. The main contribution of this approach is the introduction of an additional, intermediate layer enriched by surrogates for the information resources. These surrogates are called *index elements* in our generalized approach and are first-class objects with useful operations in addition to a simple dereference operation.

An index element contains three kinds of information:

- **partial information** reflecting selected information from the information resources in a possibly condensed form;
- **access information** describing the location of the resource, access modalities and access methods;
- **value-adding information** resulting from evaluating the resource in the context of other documents, personal experiences, projects etc.

The surrogate role of the index elements is based on several services and processes working on the partial information (and the other kinds of information) provided by the surrogate instead of accessing the underlying information resources. This includes retrieval and query evaluation but also decision support, where the user judges the usability of the resource based on the information in the index element. The use of such surrogates avoids unnecessary possibly time-consuming or expensive accesses to remote information resources.

The third kind of information makes the index element more than a pure surrogate for the information resource. Value-adding information added by the index maintainer may provide valuable hints in retrieving information resources relevant for the task at hand.

In the next section of this paper the generalized index approach is presented in more detail. Starting from an examination of the building blocks of an index, the discussion focuses on index elements and their components. The concepts are partially formalized and illustrated by examples. The generalized indexing approach provides the basis for a framework of generic services for the cooperative use and for the construction of generic indexes described in section 3. The fourth section presents a concrete cooperative service on index elements. A service merging BibTeX files based on the similarity of index elements has been implemented for this purpose. The paper concludes with a comparison of the presented approach with related work and with a summary of the results.

2 A Generalized Approach to Indexing

In this section, a generalized model for indexing is presented. The main contribution is an additional intermediate layer populated by so-called index elements. The introduced generic model is specialized considering concrete indexes as illustrating examples. Focussing on the central component of the generalized indexing approach, the index elements are examined in more detail. A partial formalization for the components of an index element is presented for this purpose. This formalization is based on an abstract, index-oriented view on the underlying resource space.

2.1 A Model of Generalized Indexing in Digital Libraries

A collection of information resources underlies an index. The index is an additional information structure providing meta information about the collection for the following purposes:

- **Structuring:** The underlying collection of information resources is structured by the index according to some syntactic or semantic ordering principle.

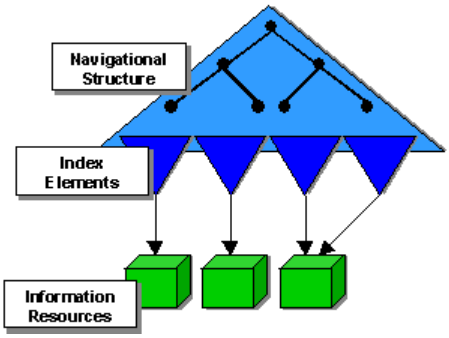

Figure 1: The Layers of a Generalized Index

- **Improving access:** The improvement of access to the information resources may be in speed, in quality, or by providing new entry points to the information.

Starting from these commonalties of indexes, it was our goal to find an approach to indexing that is general enough to encompass the wide variety of different indexes, yet specific enough to provide a useful basis for the development of applicable, generic services and tools for the use and management of indexes. Taking a more abstract view on indexing, we identified two major building blocks for generalized indexes that are illustrated by Figure 1:

- A *navigational* structure that organizes the information resources into subcollections and steers navigation through the index
- *Index elements* that populate an intermediate, linking layer between the navigational structure and the information resources. The index elements describe the indexed information resources.

Note, that the information resources are not part of the index. The index elements only provide references to these resources. The components of a generalized index model and their relationships are described in the following sections and summarized in the UML class diagram depicted in Figure 2.

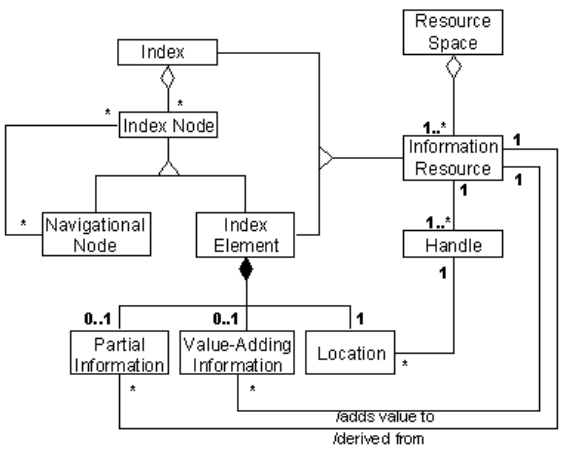

Figure 2: Components of the Generalized Index Model

In this model we assume that a (global) possibly distributed and heterogeneous resource space containing the collection of the indexed information resources is underlying the index. The information resources are equipped with handles enabling operations on the resources.

Navigational Structure

It is the main purpose of an index to allow the user (person or machine) to navigate from an entry point of the index to relevant information resources. This is supported by a graph of navigational nodes. The traversal is steered by meta information contained in the navigational nodes allowing the user to decide which path to follow. This navigational part of an index is called *navigational structure*. Concrete examples of navigational structures are B-Trees used in database indexing and alphabetic orderings in book indexes.

Each inner navigational node carries meta information for navigation plus references to further related nodes. A navigational node may also point to a collection of index elements linking these elements to the navigational structure. The navigational nodes and the index elements are, therefore, subsumed under the term *index node* in Figure 2.

The different kinds of indexes vary in the content of the navigational nodes as well as in the way the navigational structure is organized (alphabetical list, tree, etc.).

Alternative navigational structures may be provided for the same collection of information resources (and also for the same collection of index elements) implementing different approaches towards the information in the resource space. As an example, databases contain alternative indexes based on different attributes; books may contain additional indexes like lists of figures or tables in addition to traditional subject oriented indexes. In this sense, the navigational structures may just be considered as materialized queries or access expressions reflecting frequently used access paths.

Index Elements

In many cases the navigational structure does not directly point to an information resource, i.e. a text document, an image etc., but to an intermediate object surrogating the information resource. The population of such an intermediate layer is of special importance in distributed environments and in other contexts where the access to the underlying information resource is expensive or time consuming. This is true to a growing extent for information resources from the WWW and for digital libraries with respect to the numerous efforts to introduce payment methods for the consumption of information resources in this area. Therefore, we consider this part of an index as an important concept of its own and term this kind of objects *index elements*.

Examples for index elements are the entries in a BibTex File, representing the underlying publications. Furthermore, index elements play a central role in the PI-Index that has been our starting point to the subject: In the Warburg Electronic Library (WEL) project ([20], see also [17]), an interdisciplinary project of our department with the Art History department of the University of Hamburg, we are developing a digital library for art history research. It is part of the WEL project to transfer the PI-Index (PI = *Political Iconography*) into an adequate electronic counterpart. This index consists of an elaborated ontology of concepts of political iconography and of 250.000 paper cards classified according to these concepts. The cards collected in the index are index elements: They contain information about and references to the underlying art historic artifacts, but they are also artifacts of their own that are subject to art historic research without accessing the underlying resources. Further concrete examples for index elements may be found in Section 2.2.

The index elements are referenced by the nodes of the navigational structure and point to the information resource they represent. In order to meet their role as surrogates of the information resources, they contain information for two purposes: On the one hand, meta information about referenced information resource allowing to decide about the usability and potential profit of accessing the information resource and, on the other hand, information about methods and modalities for accessing the related information resources. These two kinds of information are called *decision information* and *access information*, respectively, in the rest of the paper.

Decision Information: A main purpose of index elements is to provide enough information for the user to decide, whether fetching the information resource is valuable in the context of the specific task.

Different kinds of information are needed for this decision depending on the task, the kind of index, and the underlying information resource. Decision information may include:

- **preview information** like thumbnail images, titles and abstracts of text documents, snapshots, keys of a data record; the user gets some kind of condensed summary of the information resource by this information;
- **quality judgements** like ratings, annotations, SOAPs (seals of approval) and other kinds of third party meta information (or links to it) as proposed, for example, in [15]; this is an important basis for the decision about the usability of the underlying resource;
- **classification information** like keywords, subjects, or categories grouping and classifying the underlying information resources.

The information for decision support is aggregated in two different components of an index element, one containing the *partial information* that can be directly derived from the information resource and the other encompassing *value-adding information* that results from evaluating the resource in its context.

Access Information: Access information varies from simple memory addresses of local objects through references to remote objects to complex queries for remote information resources like databases or other content providers (digital libraries). This information may be completed by access modalities like access costs or the availability of the information resources and administrative information like sizes and formats of the resources.

An important aspect in the provision of access information is the fact, that at least in the near future traditional as well as most digital libraries are *hybrid libraries*, containing digitized as well as paper resources. As a consequence, the access information may also refer to paper material. An interesting service in this context, illustrating the wide variety of possible access information, is *digitization on demand* as offered e.g. in the context of the project LEA of the University of Karlsruhe ([11]). The desired information resource is digitized on request and delivered to the user in digital form. A reference to such a service may constitute the access information in an index element.

As suggested by the UML diagram, index elements as well as entire indexes may be considered themselves as information resources of other indexes. In particular, it is possible that an index element, providing important comments and annotations about the underlying information resource, evolves into a valuable information resource of its own on an equal footing with other resources about the topic.

Each index element references exactly one information resource that, indeed, may contain a further internal structure like a collection of information resources. The precise definition of the granularity of the information resources depends on the context and the intended use. Considering the other direction, an information resource may be described by different index elements reflecting different personal or task specific views on the same resource.

2.2 Specializing the Generic Model

Indexes and index elements have been considered on a rather abstract level in the previous sections. For a better understanding of generalized indexing, the introduced concepts are illustrated with different kinds of indexes in this section.

Index in a book

An index in a book is a list of entries each consisting of a keyword and one or more page numbers. The list often contains sublists below some of the keywords in the list. Splitting such lists into single keyword/page number pairs, each pair is considered as an index element in our approach, the page number representing the access information and the keyword representing the partial information component. The associated information resource is the part of the text related with the considered keyword. The page number points to this part of the document or more precisely to the page containing it.

Street Index of a City Map

Another example from the world of the paper documents is the street index in a city map. This example is interesting for its special coding for the location of the information resource. The entries in the street index, identified as index elements in this kind of index, typically consist of the street name (partial information) and some coding for the grid square (access information) which contains the street on the map. The information resource in this case is the part of the map where the graphical representation of the considered street can be found.

For both examined example indexes (book and map) electronic counterparts exist replacing or extending the coded access information by a hyperlink to the information resource.

Database Index

Database indexes have already been mentioned several times throughout this paper as an example of a simple index. In this kind of index, the stress is on the navigational structure. The index elements containing primary or secondary keys as partial information and the memory addresses of the underlying data records as access information are rather restricted. The information resources are the data records referenced by the index elements.

WWW Bookmarks and Lists of Links

Another kind of index, a list of bookmarks referring to WWW resources, is a good example for the evolution from simple to more complex index elements of the generalized approach. In the case of pure WWW Bookmarks, the index elements only have two components, partial and access information. The access information is formed by a URL, the partial information is a document title or another kind of a short descriptor. The information resources are WWW resources like HTML or VRML documents.

Bookmark lists for subjects of special interest tend to be transformed into lists of links on an HTML page. Each link may then be annotated with further information like ratings, comments e. g. about the relevance for the actual work, and classification information. This additional information typically is context dependent and goes into the value-adding information of the index element in our approach.

Information Retrieval Index

According to their purpose, the index elements of the inverted files in information retrieval contain index terms related to information like frequencies used by the retrieval algorithms. These components form the partial information. The underlying information resources are documents. The access information consists of the references to the indexed documents.

PI-Index

The PI-Index stresses the importance of indexes for scientific work. Political Iconography (PI) is the area of art history which examines political messages conveyed in images showing regents, politicians, ceremonies, political acts, etc. The underlying assumption of the PI is, that the effects of political actions are not restricted to contracts and political documents, but are also depicted in paintings, monuments and buildings. These artifacts form the underlying information resources of the PI-Index. As mentioned above, the art history department has developed an elaborated ontology for the classification of images according to their political messages. This ontology consists of a hierarchy of terms referring to politics, political acts, and social phenomena. It includes terms as varying as science, marriage, democracy, shepherd, or revolution.

The PI-Index is based on this ontology that provides the navigational structure of the index. The index elements are represented by physical objects, a collection of about 250.000 paper cards organized by the navigational structure. This classification is not disjoint; many cards are assigned to different navigational nodes of the index. Each card contains partial information about the indexed artifact in the form of a small (thumbnail) image and further meta information, like the creator or creation of the artifact. In addition,

Index	Index Element	Partial Information	Value-Adding Information	Access Information	Information Resource
Book Index	entry	keyword	none	page number	paragraph containing keyword
City Map	entry	street name	none	grid square	map sector
Database Index	index entry	primary or secondary key	none	memory address	data record
WWW Bookmark Collection	bookmark	document title or descriptor	annotations	URL	WWW resource
Inverted File (IR)	entry	index term, frequencies, etc.	frequencies relative to document collection, etc.	document reference	document
PI-Index	index card	thumbnails	classification, comments	image source or location of original	multimedia document
BibTeX File	bibliographic entry	author, title, etc.	comments, keywords	URL or derived from partial information	electronic or paper document

Table 1: Generalized Indexing: Examples

the cards also contain value-adding information including their classification according to the ontology and comments concerning this classification.

BibTeX Files

Bib-files used as a basis for the BibTeX System are a further example for generalized indexes. A bib-file contains any number of entries of the following form, where these entries can be easily identified as index elements surrogating the publications they actually reference.

```
@bookLamport95,
   author = "Leslie Lamport",
   title = "LATEX: a document preparation system",
   pages = "272",
   publisher = "Addison-Wesley",
   year = "1994",
   isbn = "0-201-52983-1",
   price = "30,50"
   annote = "A useful introduction, covering the essentials
                  of LaTeX and BibTeX."
```

These publications, paper as well as electronic documents, in turn are the information resources. The entry types like *book, article* etc. enable a classification of entries, thus representing the basis for a simple navigational structure consisting of only one layer of navigational nodes. Similarly, the entry keys provided directly after the opening bracket impose a lexicographical order upon the entries, which can also be used for navigation.

The individual fields within an entry represent the different information components of an index element. Obviously, a variety of partial information, which directly emerges by projection or condensation of the underlying publications, is contained within the entry fields, as for example *author, title* or *pages*. In addition, fields like *annote* or *price* incorporate value-adding information, which is dependant on the

context in which the entry was created. The price of a publication may vary with time as well as an annotation is the result of the personal judgement of a BibTeX user.

Access information is not as easy to detect within a BibTeX entry. In the general case, a share of the partial information like author, title, ISBN number serves as access information, too. Equipped with this information, the user might find the desired publication in a library or a bookstore. Beyond, some BibTeX fields may also include direct access information, like a hint to the owner of a document copy in the local environment, a field *url*, which provides WWW-links to online-publications or a field *lib-congress* carrying the card number of publications available in the Library of Congress.

Bib-files are a good example of indexes where the index is used for a wide variety of operations as a unit of its own independent of the underlying documents. The ideas of this section are summarized in the following table.

2.3 A Formal View on Generalized Indexing

Having motivated the role of index elements in an index in the previous sections, we now want to examine the internal structure of an index element more formally.

The semi-formal description of the index elements chosen for this purpose is based on an abstract view on the underlying information resources called *resource space*. In this section two components of an index element, partial information contributing to the decision support and access information are identified. In the following section, the model of an index element will be refined by a further component (value-adding information) improving the decision support provided by an index element.

The formalization of the index elements and their internal structure is intended to clarify the concepts and to provide a precise basis for later modeling and implementation of the index elements. During the formalization process, we realized, that there are several aspects whose formalization is impossible or becomes overly complicated. Furthermore, because of the hybrid character of digital libraries, the domain in question is not restricted to the digital enterprise, thus introducing further problems for a complete formalization. We, therefore relaxed towards a semi-formal approach.

Information Resources in Resource Spaces

Before we are examining index elements in more detail, a closer look at the information resources is necessary to provide a basis for the further considerations. Being no part of the index, the internal structure of the information resources is not in the focus of this work. We, therefore, take a rather abstract view on the information resources driven by the requirements of the index.

An index *i* is conceptually based upon an information space, called *resource space* in our approach, which contains all the information resources indexed by *i*. A resource space *RS* is a triple consisting of a query language *QL*, an operation *fetch* managing the access to the resources, and the collection *IR* of indexed information resources:

$$RS \;\; = \;\; < QL, fetch, IR >$$

The choice of this abstraction of a resource space is motivated by the viewpoint, that one of the main tasks of the index with regard to the resources is accessing them. Expressions from the query language QL are used to identify subcollections or individual resources from *IR*. Furthermore, QL may also contain expressions for accessing resource components as described later in this section. Whereas the query expressions identify the information resources, it is the task of the operation *fetch* to actually obtain a handle for an information resource.

A *handle* in this context is defined by its characteristic to allow a user (man or machine) to perform operations on the associated information resource. This notion of a handle includes references in a programming language sense and local copies of resources that may be manipulated independently of the original, as well as physical copies of paper documents providing the only way to operate on paper material. This general understanding of a handle requires an adequate definition of the *fetch* operation. Given a query expression as its parameter, this operation returns a set of handles, i.e. a subset of *Handle(IR)* which denotes the set of all handles of the information resources in *IR*:

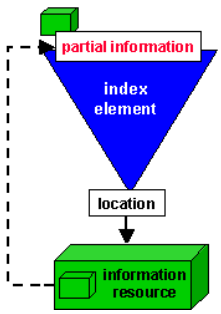

Figure 3: Index Element (Simple Case)

$$\forall \ q \in QL : fetch(q) \subseteq Handle(IR)$$

In this context, we further assume that each resource in the information space may be identified by a query expression, i.e.

$$\forall \ ir \in IR \ \exists \ q \in QL : fetch(q) \in Handle(ir)$$

where *Handle(ir)* is the set of handles for the information resource *ir*. Even if *ir* has different handles, it is assumed here, that the operation *fetch* returns only one handle requiring no further choices or operations before using this handle.

The operation *fetch* may include very different actions. The variety of these actions is illustrated by the following examples: In the case of an index in a book, all existing page numbers may be considered as expressions forming the query language. Furthermore, special styles like bold face or italics may be used to express special semantics. The *fetch* operation, here, is performed by the user, who, with the page number "as a parameter", turns to the according page, where he finds the adequate information resource in form of a text passage. As another example we may consider an index consisting of a list of links to WWW resources. In this case, the query language is constituted by URLs. The *fetch* operation is the process of following a hyperlink to obtain a local copy of the related web resource. In the case of information resources contained in a database, the query language may be understood in a more traditional sense.

So far, we only considered examples of resource spaces with homogeneous collections of information resources. There are also indexes where the underlying resource space is heterogeneous containing different kinds of information resources (e.g. text documents and images). In this case, the resource space may be considered as a collection of subspaces, where each subspace comes up with its own query language, *fetch* operation and collection of resources. In the enclosing resource space *RS*, *QL* is the union of the member query languages and *IR* is the union of the resources of the enclosed resource spaces. It is the task of the *fetch* operation of RS to choose the correct member *fetch* operation for a given query expression.

An Initial Model of Index Elements

We start with a model for a rather simple index like a database index. In this kind of index, the decision information of an index element consists of partial information obtained by component selection from the underlying information resource. A pointer to the location of the information resource is the access information provided by the index element. In this straightforward case, an index element *ie* is a pair of two components:

$$ie \ = \ < loc, pi >$$

The components of the index element and their relationship to the information resource is illustrated in Figure 3 and described in the following subsections.

Access Information

The purpose of the component *loc* of an index element is to provide access for the information resource represented by this index element. More precisely, the information contained in this component describes how to obtain a handle for the information resource.

Following our formal approach, the component *loc* contains an expression of the query language *QL* of the underlying resource space *RS*. This expression identifies the relevant information resource:

$$\forall\ ie:\ \ ie.loc \in QL\ \ \wedge\ \ \mid fetch(ie.loc) \mid \leq 1$$

Passing the component *loc* to the *fetch* operation returns a set of handles. Since each index element represents exactly one information resource and the *fetch* operation only returns one handle per information resource, the set of returned handles has cardinality 1 or 0 (invalid *loc* component).

Partial Information

In the simple case of an index element, taken as a starting point, the partial information is just based on projecting the information resource on one or more of its components. In the case of relational database indexes, this is relational projection on record fields. For information resources with a complex structure, more sophisticated operations are required in *QL* to derive adequate partial information for their description.

Starting from projections on components, the extraction of partial information by query language expressions seems to be a natural extension. In addition to the characteristics of the query language *QL* described above, we assume that expressions from *QL* may also be applied to an information resource to extract partial information. The adequate syntax and semantics of the query expressions used for this purpose strongly depends on the type of the indexed information resources. That's why the query language comes as a part of the underlying resource space *RS*.

Following these ideas, the partial information may be described by a predicate based on the *fetch* operation and expressions from the query language:

$$\forall\ ie\ \exists\ n\ \forall\ 1 \leq k \leq n\ \exists\ q \in QL:\ ie.pi = < pi_1, \ldots, pi_n > \ \wedge\ q(fetch(ie.loc)) = ie.pi.pi_k$$

According to this definition, *pi* is a tuple where each component results from applying a query *q* to the related information resource. Note, that the expression *fetch(ie.loc)* returns the handle of the resource referenced by the index element via its component *loc* enabling operations like queries on the information resource.

The decision information in the index element provided by the partial information is not restricted to components of the information resource. It also comprises condensed values like word counts, thumbnail images or snapshots from videos. Other, more powerful methods, condensing information from the information resource rather than just selecting components, are thus required to gain adequate partial information. Examples are computing thumbnail images for image resources or counting term occurrences in text documents. In our formalization approach, such functionality would be part of the query language *QL*.

The above predicate also holds for index elements with a heterogeneous composition of partial information, i.e. the component *pi* may vary in the number of its components as well as in the queries computing the corresponding *pi*-components for different index elements. Since the partial information is often subject to automatic extraction, it might be adequate for many indexes to restrict the partial information to a homogeneous structure for all index elements as follows:

$$\exists\ n\ \forall\ 1 \leq k \leq n\ \exists\ q_k \in QL\ \forall\ ie:\ ie.pi =< pi_1, \ldots, pi_n > \ \wedge\ q_k(fetch(ie.loc)) = ie.pi.pi_k$$

This predicate only holds for the homogeneous case.

2.4 The Influence of Context

The partial information considered so far is determined only by the related information resource, selecting, combining and/or condensing one or more of its components. In the general case, the evaluation of an

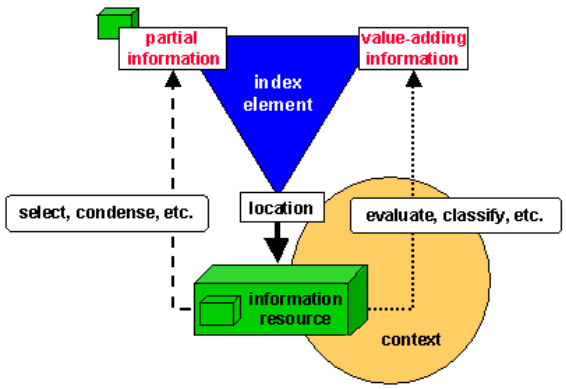

Figure 4: The Context of an Information Resource

information resource takes place in a specific context, influencing the resulting meta information. Moreover, taking into account the context provides added value for the selection, retrieval, understanding and use of information resources.

Before introducing the concept of context into our formalism, we give a short definition and some examples of context illustrating our notion of this concept.

The Context of an Information Resource

From our point of view, the *context* of an information resource is all the information apart from the resource itself influencing the understanding and use of this information resource.

"All the information" is indeed a rather fuzzy notion, but it is chosen consciously to reflect the wide variety of factors, that may influence the creation of meta information for an information resource. The context includes factors as different as

- the collection the considered information resource belongs to,
- other related collections of information resources,
- classification schemes of related application domains,
- thesauri and other language related tools and mechanisms,
- background knowledge and experience of the user, and
- general environmental conditions like time, social conditions etc.

From this enumeration, it becomes obvious that context dependent information may have an important impact on decision support when included into index elements. The incorporation of context dependent information like judgements, classification information, etc. into an index brings a new quality into it, providing value-adding information and, as a result, a framework for value-adding services for the work with information resources.

Value Adding Information

It is a clear consequence of the previous discussion of context and its value-adding character for the use and understanding of information resources, that index elements also contain context dependent information. This leads to the introduction of an additional index element component for so-called *value-adding information* into our formalism for generalized indexing. Partial information and value-adding information both provide decision information describing the related information resource. In spite of this similarity,

we decided to introduce a separate component into the index element stressing the new quality introduced by value-adding information.

The index element in the generalized approach, thus, is a triple where the third component *vai* holds value-adding, context dependent information about the related information resource.

$$ie \;\; = \; < loc, pi, vai >$$

The predicate used for the description of the component *vai* is rather similar to the predicate defined for the component *pi* in the previous section (heterogeneous case).

$$\forall\, ie\; n\; \forall\, 1 \leq k \leq n\; q \in QL\; \exists\, c \in Context\; \exists\, eval: \quad \begin{array}{l} ie.vai = < vai_1, \ldots, vai_n > \; \wedge \\ vai_k = eval(c, q(fetch(ie.loc))) \end{array}$$

As *pi* the component *vai* is also a tuple and the computation of the components of this tuple involves query expressions applied to the described information resource. In difference to *pi*, the computation is not restricted to these query expressions, but includes the application of a function *eval* that takes the context of the information resource into account. This is expressed in the predicate by passing a Context *c* as a parameter to the function *eval*.

Some words have to be said about the function *eval*. In a certain sense, it has to be regarded from a similar viewpoint as the operation *fetch*: The introduction of the function *eval* into the predicate means, that there is some evaluation process that takes the context of the resource into account. This may be a mental process of the user judging a document as well as a machine oriented process evaluating the resource in the context of the enclosing collection.

3 A Service Framework for Generalized Indexing

The purpose of this section is to categorize systematically the numerous services expected from a generalized index. This categorization leads to a service framework, which offers two major advantages. On the one hand, it promotes the development of new services for each category. On the other hand, it serves as the basis for the construction of a generic software architecture that is able to deal with a broad range of different indexes, different information resources and different application scenarios.

Our study is based on the model developed in Section 2 and distinguishes indexing services along three dimensions discussed in more detail in the following subsections.

- Categorization based on the **index components** involved (index, index element, reference to information resource);
- Categorization based on **service complexity** (elementary operations in the construction and use of generalized indexes and advanced services satisfying the requirements emerging from specific application scenarios like cooperative work with indexes);
- Categorization based on **user role** (services for passive users just consuming the contents of indexes and services for active users also participating in the construction and evolution of indexes)

Since indexes structure information resources according to the particular requirements of individuals or of a user group, they constitute an important tool in personalized workspaces as well as an artifact for cooperative work in project teams. Therefore, our discussion pays special attention to services supporting cooperation and personalization.

Section 3.4 is dedicated to services for index evolution, which is crucial for preserving the quality of long-lived indexes.

3.1 Categorizing Services by Index Component

One key achievement of the generalization of indexes described in section 2 is the explicit separation of information resources on the one hand, which are no longer tightly coupled with the index, thus supporting

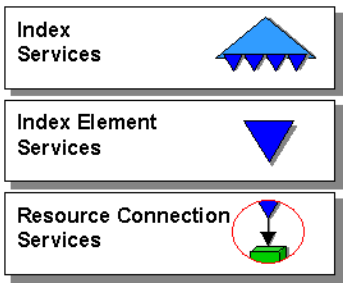

Figure 5: Categorizing services by index component

indexing in distributed environments, and of index elements, on the other hand, serving as surrogates of the indexed resources. According to this approach, we can derive a first categorization of index services: Services on generalized indexes either work on the index elements or the whole index without the necessity to access the underlying information resources. A third category of services connects the index elements and the respective information resources in some way (see Figure 5).

Within this categorization, services on index elements play an essential role, since they often satisfy information needs without access to the underlying information resources. To support this kind of work on surrogates, several services should be taken into consideration. Of course, insertion, deletion, and updates of index elements are of great importance in behalf of both user requirements as well as changes of the information resources, which affect the information contained in the respective index elements. Visiting an index element, a potential user may expect an adequate visualization of the information included. This may require different visualization modes enabling a quick overview over collections of index elements as well as a close examination of single elements in order to decide about the usability of the underlying resource. Furthermore, the user may employ advanced services upon index elements like annotating them, integrating them as important items into a personal profile or possibly imposing access restrictions for other users.

Many of the services required for index elements can also be applied to the index as a whole. Yet, the underlying algorithms may have to meet different requirements. That is why separate services like updating, deletion, visualization and annotation may be necessary for indexes as a whole. Furthermore, the entirety of an index demands additional services. Creating new indexes is essential, especially if several indexes are used to develop different viewpoints on the underlying information resources. Other services exclusively emerging on the level of whole indexes are traversing the paths of the navigational structure, managing user sessions on the index or administering subscriptions to the index or parts of it. Retrieving single index elements or groups of index elements provides direct entry points into the index in addition to following the navigational structure.

The services described so far can be carried out on either indexes or index elements alone. Nevertheless, a user may decide to actually access some of the underlying information resources. To support this access and to furthermore guarantee that the information contained within the index elements is in sync with the information resources, another service category is introduced. It covers all services connecting the index with the index resources, like several kinds of access services, the automatic derivation of partial information, and support for the construction of value-adding information in providing adequate parts of the information resources. Finally, the quality of a generalized index has to be assured over time by (semi-) automatic change management services propagating changes of both the contents as well as the location of information resources to the respective parts of the index.

3.2 Categorizing Services by Service Complexity

Investigating service categories on the component level of a generalized index as performed above also uncovers another important criterion for the categorization of index services. Some of the services enumerated seem to be essential to an index, whereas other services appear as advanced features desired in special

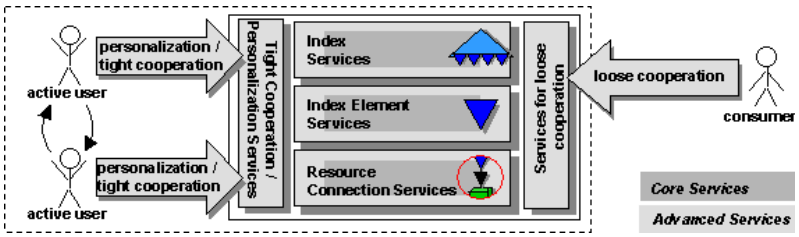

Figure 6: Categorizing Services by User Role

application scenarios like cooperative index use or personalization of indexes. Of course, services like creation, insertion, update and deletion are crucial to the use of an index in digital libraries. The possibility of editing the value-adding information of an index element is particularly stressed in this context. Navigation and retrieval are important entry services for indexes. Visualization of index components is another basic requirement. Finally, a reasonable change management ensures the usability and preserves the quality of the index over time.

Yet, services like profiling, session management, versioning, the administration of access rights, accounting, subscribing to notifications about changes in the index need not be considered, unless several users employ the index extensively for their personal work or for cooperation between each other. However, it may be difficult to implement such advanced services if solely by the use of a small set of core services.

3.3 Categorizing Services based on User Role

Considering the different scenarios in which generalized indexes may be used leads to the conclusion, that mainly two different groups of index users can be distinguished. Some users actively work on an index. These active users maintain and improve it by indexing new information resources, i.e., by inserting new index elements, by providing more and more value-adding information as a result of their deepened understanding of the information resources, or by restructuring the index for a better categorization of its elements.

In this respect, the active usage of generalized indexes can be regarded both a personalized employment of indexes reflecting the opinion of the respective user towards the information resources as well as a tight intra-project cooperation by means of indexes making the existing knowledge about the information resources accessible to all participants of a project team. Thus, generalized indexes are used for the personalization of information structures as well as to support cooperative work in project teams which implies manifold additional service requirements.

The active usage of an index has to be supported by core services like creation, insertion, deletion, updating and adaptation services for value-adding information. Advanced services like the possibility of restricting access to the index, combining and merging several different indexes, managing and tracing user sessions, or versioning of indexes and parts of indexes open new options for the creative and cooperative use of generalized indexes.

Other users are merely consumers of an index. They just follow the paths of the navigational structure or retrieve single elements for a closer inspection. Yet, they don't change any component of the generalized index and, thus, do not take part in its evolution. Services required for index consumption are e.g. retrieval, navigation, administration of access rights, subscription to an index or parts of it, notification in the case of changes, or accounting of index usage, and an adequate visualization of the navigational structure as well as of index elements, as discussed above.

Index consumers are autonomous agents, i.e. they do not belong to the group of persons maintaining the index. Instead, they own read-only access rights, eventually restricted to parts of the index, guaranteed by contract. Consumption of indexes, hence, is the basis for inter-project cooperation. Yet, index consumers and active index users do not belong to disjoint groups of persons, since an active user might also take the role of a consumer, to just navigate through the index, to investigate its current state, and to find out what

changes are to be carried out in the future. Thus, to offer a beneficial indexing system both consumption services and active usage services have to be integrated smoothly.

3.4 Services for Index Evolution

The major goal of using generalized indexes is to gain an overview over large amounts of information being available in form of often distributed and heterogeneous information resources. This task is pursued by extracting relevant pieces of the information into index elements to provide decision support and by further structuring these index elements imposing a navigational structure upon them. In this regard, a generalized index is subject to a permanent evolution resulting from two different sources. On the one hand, the underlying collection of information resources, especially if it is as large and widespread as e.g. the WWW, changes rapidly. This, of course, has to be reflected within the components of an index, in order to guarantee the compliance with the goal described above. Thus, the index should follow the ongoing evolution within the indexed collection. Different services may be used to support evolution including change management on the basis of subscription and notification, regular checks of the underlying resource, and the inclusion of version information into the index element if the underlying resource supports versioning. Which of these services can be used in a concrete situation, strongly depends on the autonomy of the content provider and his will to cooperate with the index provider.

On the other hand, the active usage of an index leads to an evolution taking place inside the index itself, where the changing personal background and experience of the index users are mirrored in a permanent semantic and structural variation concerning all of the index components. These requirements are to a large degree covered by the services for the active index use as discussed above.

4 Merging BibTeX Collections: An Example for Cooperative Indexing

Bib-files as the basis of the BibTeX system are a good example for personalized and cooperative indexes which are subject to a constant evolution. In particular, Bib-files illustrate the significance of index elements as surrogates for the actual information resources, as pointed out in section 2, demonstrating how many operations can actually be carried out on index elements without accessing the original source.

The original task of the BibTeX system is to facilitate the generation of bibliographic reference lists within LaTeX documents and to support the maintenance of bibliographic references. In practice, a BibTeX user tends to maintain one or several personal Bib-files, containing numerous entries collecting references to *relevant* documents read, written, cited, or considered for further study. This leads to the assumption, that a bib-file does more than merely collecting references. It can be regarded as a tool for the personalized and also for the cooperative handling of references sharing such files between project members. As discussed in Section 2.2, a Bib-file is a generalized index. Thus, developing index services for Bib-files enables an evaluation of our approach and can lead to an amplified understanding of generalized indexing. In this section, an exemplary cooperative service upon BibTeX indexes is introduced, which enables users to intelligently merge the contents of their respective BibTeX collections ([23]).

4.1 Cooperative Work with Bib-files

Over time the purpose of the BibTeX system has more and more evolved from simply supporting the generation of bibliographic lists to handling bibliographic references as valuable information of its own. This is reflected by the permanent growth of information summarized within single entries like the inclusion of long annotations or of whole abstracts expressing the growing need for decision information addressed by the introduction of the index elements. Furthermore, new types of entry fields are introduced time and again by providing additional style files. These changes within the structure of bib-files as well as the invention of new tools in connection with the BibTeX system, like e.g. search engines, macro expansion, sorting tools or even database management software, point out the desire to use BibTeX within personalized or cooperative contexts and the necessity to follow the evolution in the area.

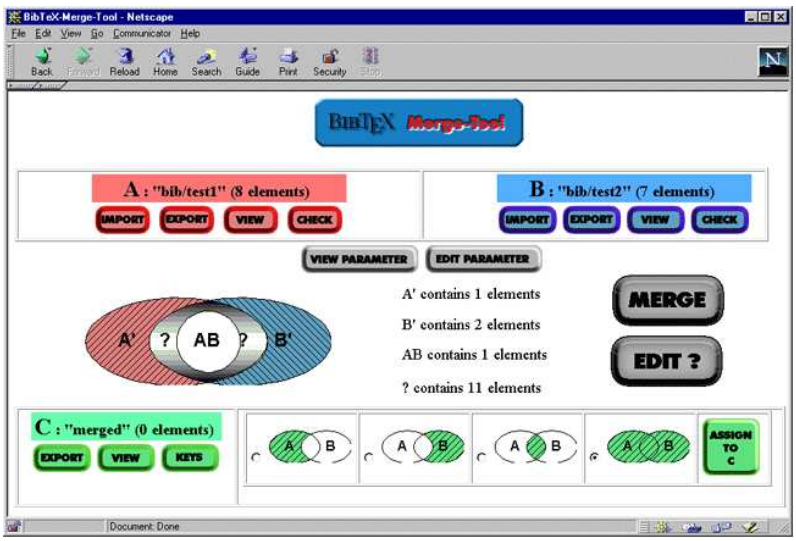

Figure 7: Main Window of the BibTex Merge Tool

As part of the *Kolibri* project [19], Bib-files are modeled as generalized indexes and a infrastructure is being implemented that realizes several of the index services described in the previous section. In this section, we focus on a service for the cooperation between multiple users with personal (or group) Bib-files.

As a concrete application scenario, imagine two researchers aiming to write a publication together. This work would certainly be facilitated, if the references collected by both persons were not only used together, but if moreover both persons were capable of comparing and completing their knowledge on the background of the index elements of the other. This simple use case already shows that a facility for combining Bib-files developed and maintained independently is of high relevance for cooperative work.

Although Bib-files are often regarded as a simple kind of databases, their merging cannot be accomplished by standard database technology on exact-match operators. For example, even highly structured Bib-fields like *author* or *month* may differ syntactically for two entries that refer to the same information resource (book, article, ...). Thus, a vague approach (in the sense of probabilistic information retrieval concepts) seems to be more adequate for Bib-file merging. This necessitates the employment of a broad, in most cases linguistic, domain knowledge for the respective entry fields on the one hand. On the other hand, an interaction between user and merge tool has to be considered in cases, where the equality of two entries cannot be detected automatically by the merge tool itself.

4.2 Tool Support for Cooperative Work with Bib-Files

Implementing the BibTeX-Merge-Tool, we extensively rely on the merits of object-orientation to follow the idea of categorization as described in Section 3 and to achieve a clear cut separation between the application as such and the underlying domain knowledge. The domain knowledge is collected within a separate class containing all algorithms for the comparison of two entry fields. This separation enables the introduction of improved algorithms without changing the whole application on the one hand and the reuse of the existent algorithms for further applications on the other hand, setting a clear focus for our future work in the Kolibri project.

Taking a closer look at the BibTeX-Merge-Tool shows that the whole application is controlled from of the main window depicted in Figure 7. The user can import two different Bib-files in order to combine them. To ensure that none of the input files contains multiple references to the same document (a prerequisite for the merge step), a duplicate check operation is provided. Moreover, the input fields can be viewed and edited element by element or exported in Bib-file format.

The actual merging process takes place in the center of the main window. Clicking the *Merge* button

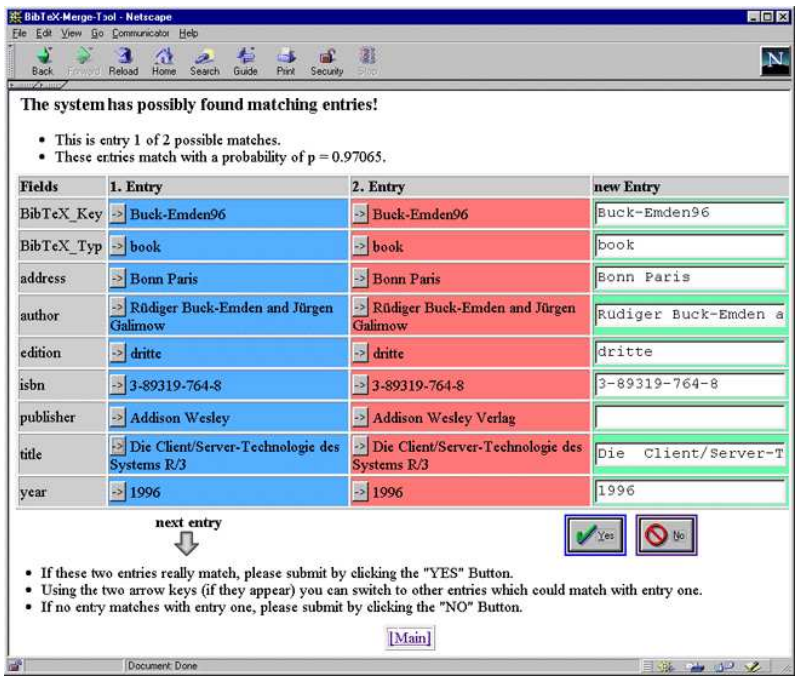

Figure 8: Comparison Window of the Merge Tool

produces four output sets as represented in the Venn Diagramm. The BibTeX-Merge-Tool detects, which entries in the two files refer to different, to the same or possibly to the same information resource. Unique entries are included in the set *A'* and *B'*, respectively. Entries considered to refer to the same document are stored in the set *AB*, whereas entries which possibly refer to the same information resource can be found in a data structure called *"?"*.

The partitioning of the result is derived from an entry-by-entry probabilistic comparison of the input sets *A* and *B*. If an entry's probability stays below a certain lower bound for all other entries, this entry is put into set *A'* or *B'*. If it exceeds a certain upper bound for exactly one other entry, one of the entries is put into set *AB*. For probability values between the lower and upper bound, the respective entries (and their probability values) are inserted into the set *?*. The lower and upper bound both are parameters of the tool, which can interactively be changed by the user.

Making use of vague algorithms, the BibTeX-Merge-Tool strongly relies on interactive user feedback. Hence, all groups of entries in the set *?* can be explored by the user and then be assigned manually to the set *A'*, *B'* or *AB*. This activity is launched by choosing the option "edit ?" and leads to the comparison window depicted in Figure 8.

Here the user can compare the entries in question one at a time. If he comes to the conclusion that two entries refer to different documents, he will choose "no". If the entries refer to the same document, the user has the possibility to merge information from both entries into the fields of a new entry. Proceeding like this until the end leads to a decision for each of the entries in set *?*, moving it into one of the other sets.

5 Related Work

Constructing and using lists of links to WWW resources for topics of current interest, indexing is often part of daily scientific work. The (centralized) development of commented link lists for specific user and interest groups is an important value-adding service for novices as well as experts of an area, structuring the available material and facilitating the retrieval of relevant information resources. Numerous of such indexes may be found in the Web. For example, the project SOSIG provides resources relevant for social science

research and education ([8]) and Flybrain, collects information resources about the drosophila (([3]))

This paper unites the different ideas of supporting and exploiting surrogates of information resources as proposed in the literature within a general model, emphasizing the existing commonalties. In the Dublin Core Project [22] the usefulness of such surrogates for resource discovery described by the Dublin Core element set is stressed. In [10] the lack of surrogates is identified as one of the key restrictions to the retrieval capabilities of WWW indexers. The system GERHARD (German Automated Retrieval and Directory ([2])), built in the context of a DFG project at the University of Oldenburg, is an example of a concrete application of a generalized index containing a navigational structure and index elements. The index elements, called *detailed indexes* in this approach, contain a title, a link (URL) to the underlying information resources, and some classification information. They may be inspected before accessing the linked information resource. The navigational structure is based on a classification scheme which is applied to the information resources automatically.

Sharing the widely accepted opinion that meta information is an important factor for digital libraries, we enable the integration of any form of meta information which proves to be useful in the course of other projects by the introduction of the general notion of an index element. Meta information is promoted by various projects focusing on different aspects like architectural issues ([1]), classification schemes [4, 9], or models for their description [21, 22]. Value-adding information, in particular, regarded as important component of the index element in our model, comes up in a wide variety of flavors supporting different kinds of services: This includes, for example, content ratings and seals of approval as described in [[21, 22]. Value-adding information, in particular, regarded as important component of the index element in our model, comes up in a wide variety of flavors supporting different kinds of services: This includes, for example, content ratings and seals of approval as described in [14], ratings for content selection (see for example PICS [13]), annotations and personal comments [14, 12], and classification information as considered in the WEL project.

The indexing model described in this paper permits to directly integrate services for the management, maintenance and use of generalized indexes and index elements. This notably eases the effort necessary for extending the functionality provided for an index like e.g. by services for change management and for the merging of index elements and supports the value-adding character of our approach

[16] provides a classifying framework for change management on the basis of events and notifications. An interesting mechanism for change detection and change management may be found in [6, 5]. Especially the query subscription service proposed in [6] is applicable for change management on partial information. This subscription service is based on an adequate representation of changes in selected parts (defined by a *polling query*) of the underlying information resource and querying these changes by *filter queries* defining changes relevant to the subscriber. Using the definition of the partial information for the polling query, the service may be used to update partial information and to inform the index user about these changes. A further possible application that may profit from the proposed representation for changes (and differences) is the comparison of indexes developed by different persons with similar underlying resource spaces.

The problem of merging sets of index elements comes up in the context of cooperative index use and has been addressed in section four for the concrete case of BibTeX entries. Different merging algorithms are needed depending on the format and content of the index elements. For the concrete BibTeX merging tool, the method proposed in [7] has been adapted. Algorithms and methods proposed for example for copy detection [18] may be exploited as starting points for merging approaches on other information resources.

6 Concluding Remarks

In this paper we proposed a generalized approach to indexing that avoids the strict coupling between the index and the underlying information resources. Examining existing indexes and the special requirements in distributed environments with heterogeneous collections of information resources, we motivated the introduction of an additional, intermediate layer populated by so-called index elements. Providing partial information, access information and value-adding information, the index elements are surrogates for the indexed information resources. A semi-formal description with predicates specified on an abstract level the components of an index element.

Starting from this model, we have identified and classified services based on criteria like usage modes.

Special focus was on index evolution, cooperation based on indexes and personalization aspects. This classification provides a firm basis also for an implementation framework of generic indexing services.

By implementing a merge tool as a concrete cooperative service on BibTeX collections we verified the indexing model and we further developed the model for cooperative index use and evolution.

Acknowledgements: We would like to thank André Wittenburg who did a great job in implementing the BibTeX Merge Tool and to Gerald Schröder for the numerous discussions on the topic.

The research described in this paper was supported by the HSPIII Project WEL and the DFG Project KOLIBRI (DFG Schm 450/7-1).

References

[1] M. Baldonado, C-C K. Chang, L. Gravano, and A. Paepcke. The Stanford Digital Library metadata architecture. *International Journal on Digital Libraries*, 1(2):108–121, September 1997.

[2] BIS, University of Oldenburg. Homepage of the project GERHARD, http://www.gerhard.de.

[3] Flybrain Database Review Board. Homepage of the project FLYBRAIN, http://flybrain.uni-freiburg.de.

[4] K. Böhm and T.C. Rakow. Metadata for Multimedia Documents. *Sigmod Record*, 23(4):21–26, December 1994.

[5] S. Chawathe and H. Garcia-Molina. Meaningful Change Detection in Structured Data. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pages 26–37, Tuscon, Arizona, May 1997.

[6] S. S. Chawathe, S. Abiteboul, and J. Widom. Representing and Querying Changes in Semistructured Data. In *Proceedings of the International Conference on Data Engineering*, pages 4–13, Orlando, Florida, February 1998. http://www-db.stanford.edu.

[7] J.C. French, A.L. Powel, J.L. Pfaltz, and E. Schulman. Automating the Construction of Authority Files in Digital Libraries: A Case Study. In C. Peters and C. Thanos, editors, *Research and Advanced Technology for Digital Libraries, Proceedings of the first European Conference ECDL'97, Pisa, Italy, September 1-3, 1997*, volume 1324 of *Lecture Notes in Computer Science*, pages 55–71. Springer, 1997.

[8] Institute of Learning and Research Technology, University of Bristol. Homepage of the Social Science Information Gateway (SOSIG), http://www.sosig.ac.uk.

[9] V. Kashyap, K. Shah, and A. Sheth. Metadata for Building the MultiMedia Patch Quilt. In V.S. Subrahmenian and Sushil Jajodia, editors, *Multimedia Database Systems*, pages 297–319. Springer, 1996.

[10] C. Lagoze. From Static to Dynamic Surrogates: Resource Discovery in the Digital Age. *D-Lib Magazine*, June 1997. http://www.dlib.org/dlib/june97/06lagoze.html.

[11] Library of the University of Karlsruhe. Homepage of the project LEA, http://www.ubka.uni-karlsruhe.de/docdel.

[12] T.A. Phelps and R. Wilensky. Multivalent Annotations. In C. Peters and C. Thanos, editors, *Research and Advanced Technology for Digital Libraries, Proceedings of the first European Conference ECDL'97, Pisa, Italy, September 1-3, 1997*, volume 1324 of *Lecture Notes in Computer Science*, pages 287–303. Springer, 1997.

[13] P. Resnick and J. Miller. PICS: Internet Access Control Without Censorship. *Communication of the ACM*, 39(10), October 1996.

[14] M. Röscheisen, C. Mogensen, and T. Winograd. Shared Web Annotations as a Platform for Third-Party Value-Added Information Providers: Architecture, Protocols, and Usage Examples. Technical Report CSDTR/DLTR, Computer Science Department, Stanford University, Stanford, CA 94305, U.S.A., November 1994. http://www-pcd.stanford.edu/COMMENTOR.

[15] M. Röscheisen, T. Winograd, and A. Paepcke. Content Ratings and Other Third-Party Value-Added Information: Defining an Enabling Platform. *D-Lib Magazine*, August 1995. http://ukoln.bath.ac.uk/dlib/dlib/august95.

[16] D.S. Rosenblum and A.L. Wolf. A Design Framework for Internet-Scale Event Observation and Notification. In *SIGSOFT '97/EESEC '97 Proceedings of the Sixth European Software Engineering Conference/ACM SIGSOFT Fifth Symposium on the Foundations of Software Engineering, Zurich, Switzerland, September 22-25, 1997*, volume 22. ACM, November 1997.

[17] J.W. Schmidt, G. Schröder, C. Niederée, and F. Matthes. Linguistic and Architectural Requirements for Personalized Digital Libraries. *International Journal of Digital Libraries*, 1(1), 1997. http://www.sts.tu-harburg.de/papers/1996/SSNM96a.

[18] N. Shivakumar and H. Garcia-Molina. Building a Scalable and Accurate Copy Detection Mechanism. In *Digital Libraries '96, 1st ACM International Conference on Digital Libraries, March 20-23, Bethesda, Maryland, USA*, 1996.

[19] Software Systems Institute, Technical University Hamburg-Harburg. Homepage of the KOLIBRI project, http://www.sts.tu-harburg.de/projects/Kolibri.

[20] Software Systems Institute, Technical University Hamburg-Harburg. Homepage of the Warburg Electronic Library project (WEL), http://www.sts.tu-harburg.de/projects/WEL.

[21] W3C. Resource Description Framework (RDF) Model and Syntax, W3C Working Draft, http://www.w3.org/TR/WD-rdf-syntax, February 1998.

[22] S. Weibel and J. Hakala. DC-5: The Helsinki Metadata Workshop. *D-Lib Magazine*, February 1998. http://www.dlib.org/dlib/february98/02weibel.html.

[23] André Wittenburg. Informational Retrieval Unterstützung zur kooperativen Arbeit mit Bibliographiedatenbanken, 1998.