The Webspace Method: On the Integration of Database Technology with Multimedia Retrieval

Roelof van Zwol
University of Twente
Department of Computer Science
P.O.box 217, 7500 AE, the Netherlands
zwol@cs.utwente.nl

Peter M.G. Apers
University of Twente
Department of Computer Science
P.O.box 217, 7500 AE, the Netherlands
apers@cs.utwente.nl

ABSTRACT

Large collections of documents containing various types of multimedia, are made available to the WWW. Unfortunately, due to the un-structuredness of Internet environments it is hard to find specific information when one is looking for it. Search engines available can only rely their results on information retrieval techniques and most of the time they lack the desired power in query formulation.

Modelling data on the web, as if it was designed for use within databases, should provide us with the necessary basis for enhancing this query formulation. This of course requires special care for dealing with the included multimedia data and the semi-structured aspects of data on the web. Modelling the entire web would be too ambitious, therefore we focus on a more feasible environment, like the intranet, where one can find large collections of related data.

With the webspace method we have already shown how to deal with the various aspects of semi-structured data in large collections of related documents. In this paper we focus on the integration of our webspace method for *concept*-based search with *content*-based multimedia information retrieval (IR).

A webspace consists of two levels. At the document level, a webspace is considered to be a collection of related documents. At the semantical level, concepts are defined to be used in the documents at the document level. By modelling these concepts using a webspace schema a semantical level of abstraction is gained. This supplies the necessary platform for querying data available within a specific webspace. For the integration with content-based information retrieval an existing IR model is adopted. We will discuss how this is used in the context of Mirror, a Multimedia DBMS, and how this framework is used for the integration with the webspace method for concept-based search.

Keywords

Modelling data on the web, concept-based search, content-based information retrieval, daemon data dictionary.

1. INTRODUCTION

The main objective of this paper is to (1) present the webspace method for modelling large collections of web documents, containing related information, and (2) to show how integration of the webspace method with a framework for multimedia retrieval results in powerful query facilities for a webspace, combining both conceptual search and content-based information retrieval. With the webspace method we aim at applying existing database technology for modelling data on the web. Modelling the entire WWW is too ambitious,therefore we focus on more feasible environments, like the intranet, where one can find large collections of related data.

A webspace consists of two levels. At the document level a webspace can best be seen as a collection of documents describing related information. The documents can contain all kinds of multimedia data and can have an irregular structure. At the semantical level of a webspace, concepts are defined, describing the content of documents at a semantical level of abstraction. These concepts are modelled in a webspace schema, using existing object-oriented modelling techniques.

Based on the model for webspaces, modelling aspects for data on the web are studied. Since data on the web is considered to have an irregular structure, modelling such data is hard. In the past we have described how these semi-structured aspects of data are dealt with in our model for webspaces [22].

To be able to deal with the various types of multimedia involved, our concept-based model for a webspace is extended with Mirror's framework [19] for content-based multimedia DBMS. It uses an IR model, which allows users to specify their information need in terms of keywords. Next, relevance feedback is used to calculate the relevance of various types of multimedia, like text-fragments, and images. In this article the implications of the integration with Mirrors framework for content-based retrieval are discussed.

Once a webspace is defined properly at both the document level and the semantical level, the webspace schema is used to generate a schema for our object server. The object server stores the meta-data, regarding concepts used on the webspace, and the content-based information retrieval. Based on the webspace schema, a webspace can be searched by formulating queries over the webspace schema. Instead of querying single documents, one can formulate a query over information stored in separate documents, using both concept-based search and content-based IR, e.g. queries can be formulated, where conceptual information is used to find postcards, coming from destinations, which lay inside a region, with the name 'Asia', and where the content of the postcard is searched for the words 'station' and 'railway'[1].

State of the art

Modelling data on the web is an ongoing research area, where many research projects have been positioned around. Closely related to our approach is the Araneus project [12] where also existing database technology is applied to the WWW. The web documents and hyper texts are modelled and stored using a relational data model, called ADM. This project is also concerned with handling both structured and semi-structured documents. The main difference with our approach is that we aim at combining concept-based search with content-based information retrieval, to come up with new query formulation techniques for data on the web. Modelling data, among navigational design and interface design is also an issue in OOHDM [17], the object-oriented hyper media design method. In WebML [5] a modelling language for designing web sites is proposed. A structural model, similar to an E/R model, is used to deal with the content modelling problems.

Others, like in Lore [10], XML-QL [6], XGL [4], WebOQL [2] use the structure of the XML document as their model. Adding support for regular path expressions does enable them to search for patterns and structure in the XML data. Except for [9], where text-based search is integrated into XML-QL, content-based queries are not supported over complex multimedia documents.

Of course in the field of information retrieval, and multimedia databases many sophisticated models are proposed to retrieve information. We do not aim to come with better IR techniques, but aim to combine existing IR techniques with conceptual search, using a database oriented approach. For those interested in information retrieval and MM-DBMS, we refer to [19], where these matters are discussed.

Organization of this paper

The remainder of the paper is organized as follows. First, we will go into the webspace model, and discuss the architecture set up to implement the webspace method (Section 2). Section 3 explains how the integration with multimedia retrieval is realized, and how a webspace can be queried, using concept-based search and content-based retrieval. Going towards an implementation, Section 4 discusses the retrieval of web-objects from XML documents and how meta-data is obtained from these objects, to populate the meta-database. Finally we will come to the conclusions in Section 5.

2. THE WEBSPACE METHOD

Searching for data on the web can become more powerful, when this data is modelled using database techniques. We already argued that this cannot be done straightforwardly, since the web is not a database [14]. When focusing on a more limited environment, like defined for a webspace, the techniques developed for databases can be applied. By following this approach two major obstacles were encountered. Dealing with data on the web, also implies dealing with multimedia. Others, like [20] already argued for the need to extend database technology to deal with these types efficiently in a database environment. As will be explained in Section 3 we adopt the Mirror framework for content and multimedia DBMS [19] for dealing with such data efficiently. The second problem addresses the semi-structuredness of the data involved. The current trend for dealing with such data is by adopting a graph-based data model like OEM [10] or others [15]. We have chosen not to follow that approach but instead an object-oriented approach to model such data. In our model for a webspace these two research areas are brought together, to come up with better modelling facilities for data on the web and more powerful query mechanisms for large collections of related data.

2.1 Webspace model

When looking at a collection of related documents, it is possible to identify a (finite) set of concepts, which describe the content of the documents at a semantical level [22, 1]. In our model for webspaces this is exploited by identifying two levels. At the document level one can find a collection of related documents, which should be (made) available to the Intranet. At the semantical level the before mentioned concepts should be defined and modelled in an object-oriented schema, called the webspace schema. Section 2.1.1 describes how webspaces should be defined at the semantical level in a webspace schema, and in Section 2.1.2 it is explained how this relates to a webspace at the document level.

2.1.1 Semantical level

For each webspace a set of concepts should be defined, describing the content of the documents involved. Such concepts are identified by a unique name. The semantical level is then formed by a webspace schema. This schema is based on an object-oriented data model, which allows concepts to be modelled in terms of (1) classes, (2) attributes of classes, and (3) associations over classes. The model also includes a generalization mechanism, allowing classes to be defined as subclasses of other classes. Together, the set of classes, attributes, and associations form a partition which is equal to the set of concepts defined for a webspace.

Once a concept is defined to be a class in the webspace schema, it cannot be reused as an attribute or an association in the same webspace. Likewise for the attributes and associations involved. Since authors publishing their documents on a webspace cannot be assumed to have the skills of a database administrator, notion of types, and cardinality are left out of the data model at this high conceptual level of abstraction. To our opinion such problems should not be visible to the users, and should be dealt with at the logical and physical level of the webspace system, as is discussed in Section 2.2. Figure 5 shows a fragment of a webspace schema, set up for our lonely planet example. In Section 4 this example is worked out in more detail.

2.1.2 Document level

The document level of a webspace stores the data involved. For this purpose XML [21] is used to mark up the data. When the data involved has a rather structured character, it is very well feasible to store the data using a regular DBMS, as proposed in [13, 7] and others. Although it is also feasible

[1] In Section 3.4 this query is worked out in more detail.

to deal efficiently with more semi-structured data stored in a DBMS [8, 24, 16], we have chosen to store the data in XML-documents. With XML one can easily mark up the content of a document, thus allowing authors to make the structure of their documents explicit. Others use this tree-based structure directly as the input for searching through the content of single documents. But only basing the search on the structure might lead to semantic misinterpretations. To bridge this semantical gap with the user searching the data within a webspace, concepts are defined in a webspace schema providing a semantical layer over the collection of documents involved. From this perspective, each element and attribute used in an XML-document, should correspond to a concept defined in the webspace schema.

Following this approach, ensures that any document available at the document level of a webspace is seen as a materialized view of the webspace schema. Any document on the webspace describes (a part of) the webspace schema, thus forming a view on the schema. The document materializes this view, since it also contains data.

2.2 Architecture of the webspace system

To implement the ideas for modelling and querying the content of webspaces the architecture as shown in Figure 1 is set up. It is based on a three layer architecture, consisting of a physical, logical, and conceptual layer. Going bottom up through the figure, the following parts are identified.

- **Object server.** The object server is formed by the physical layer and logical layer of the architecture. At the physical layer the meta-data obtained from the XML documents is stored in either Monet [3], a binary relational database kernel, or Postgres [18], a object relational DBMS. On top of both databases we use Moa [3, 19]. Moa consists of a structural object-oriented data model and algebra. This provides us the desired physical data independence at the logical layer. Moa uses an extensibility mechanism, which on one hand adopts the extensibility mechanism provided by the physical layer. But also provides structural extensibility, which is used to model the semi-structured aspects of the webspace model at both the type, and attribute level efficiently [22, 8]. The structural extensibility is also used to implement the Mirror framework for content-based multimedia information retrieval as proposed by [19]. In Section 3 the integration of the Mirror framework in the webspace method will be discussed. Section 4 will elaborate on the retrieval of meta-data from XML documents.
- **Web object retrieval and storage layer.** One of the central components in the webspace system is the layer responsible for retrieving the meta-data from the documents available. Using the webspace schema supplied by the authors of the webspace, several intermediate schemas are generated to populate the object server. As output this component delivers a Moa schema at the logical level, and a physical schema used by Monet (MIL), and Postgres (PSQL). Secondly it is responsible for obtaining web objects from the XML documents and retrieving all kinds of meta-data from the various types of multimedia involved. When all the meta-data from one web object is extracted, it is stored at the physical layer, using either Monet, or Postgres.
- **Webspace modelling tool.** It is responsible for all the tasks that need to be performed in order to set up a single webspace, at both the semantical level, and the document level [23]. From the author it expects (1) the concepts to be defined in a webspace schema; (2) a view, representing the structure of the document to be created; (3) the content that should be added, and finally (4) the information regarding the presentation.
- **Webspace and content query engine.** This component consists of two 'engines', one for composing queries regarding concept-based search on a webspace, and one for doing content-based querying over the multimedia involved. In Section 3.4 some queries are formulated, illustrating the integration of these two engines.
- **Webspace query front end.** With this front end we intend to offer the user a query interface that allows him to compose complex queries using a graphical notation of the webspace schema and combine this with content-based retrieval techniques.

3. INTEGRATION OF CONCEPT-BASED SEARCH WITH CONTENT-BASED IR

To realize the integration with content-based information retrieval, we adopted ideas developed for IR systems. There users can specify their information need, using some keywords and relevance judgments on previously retrieved text documents (called relevance feedback). Similar approaches are possible for other multimedia types, like images. These systems base the retrieval of documents on a probabilistic model. Section 3.1 discusses the IR retrieval model used for Mirror, a multimedia DBMS (Section 3.2). We adopted the Mirror framework to realize the integration of content-based information retrieval with the webspace method for concept-based search over a collection of web documents. In section 3.3 these integration issues are discussed. At the end of this section we will show by some query examples how a webspace can be queried by both its concepts and content.

3.1 IR model

The objective of IR models is to find **similar** multimedia objects, using a keyword based query and relevance feedback. Objects are judged to be similar, based on their content. Notice that this comparison between objects is usually based on meta-data extracted from original documents, such as words occurring in a text fragment.

In an IR model the similarity between a query and a multimedia object is calculated, using a probabilistic approach. Most IR models rank their documents using two parameters: the term frequency and the inverse document frequency.

- **Term frequency:** For each pair of term and document, *tf* is the number of times the term occurs in the document.
- **Inverse document frequency:** For each term, *idf* is the inverse number of documents in which the term occurs.

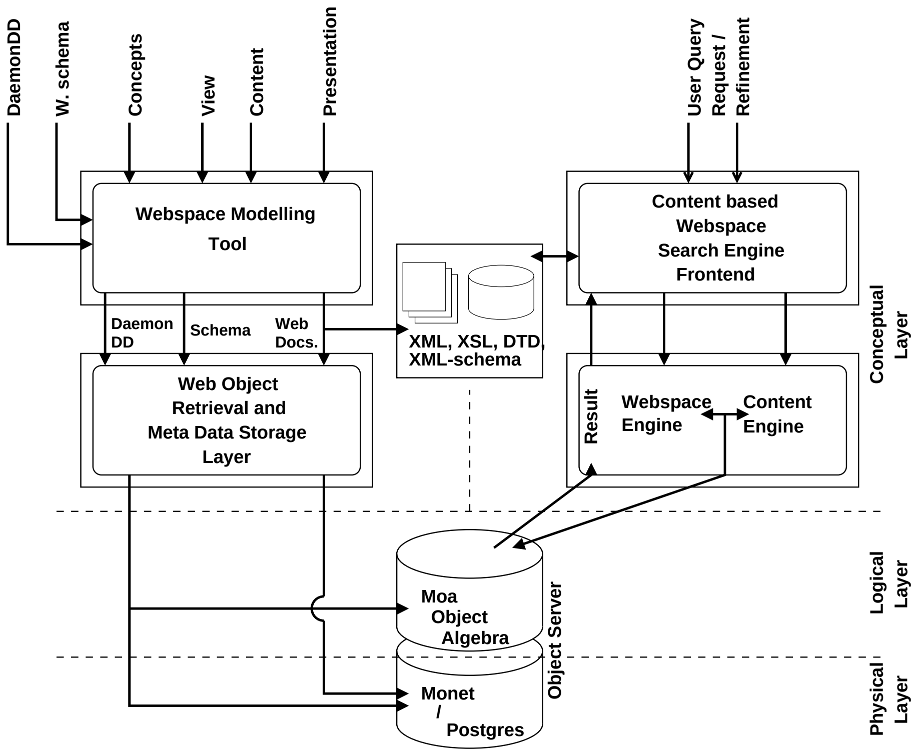


Figure 1: Architectural overview of the webspace system

The ranking of a document (or a multimedia object), given a query is then determined by the sum of the product between *tf* and *idf* of the query terms, occurring in the document, sometimes normalized with the document length. More detailed information on the IR model and its ranking formula can be found in [19, 11] In most IR systems query results are computed as follows:

- Select the term frequencies *tf* and inverse document frequencies *idf* for each query term.
- Compute the document's term ranking for each document (*tf* × *idf*).
- Update or add the document ranking each time a document's term ranking is computed.
- Repeat steps two and three for all documents and query terms.

3.2 Mirror, a multimedia DBMS

The IR model described in the previous section, is used for the retrieval of content-based multimedia in Mirror, a content-based multimedia DBMS. Mirror's implementation is based on the same object server as shown in Figure 1. The specific domain knowledge, concerning the IR model is implemented at the logical level, using Moa's structural extensibility mechanism. Lower level information retrieval techniques are implemented by extending the Monet database kernel.
In Mirror the term frequency and inverse document frequency are described by the database relations:

- `TF(term,document, tf)`
- `IDF(term,idf)`.

The terms used in a query are described in the relation `Q(term)`. Query processing is then handed to the object server, where specialized optimizers can be used to deal with the query efficiently.
The set-oriented nature of Mirror's IR query processing is illustrated in the steps below:

- Initialize the query process, given a query Q.
- Limit TF and IDF, by matching them with the query terms of query Q:
  $TF_Q = TF \ltimes Q$, $IDF_Q = IDF \ltimes Q$.
- Place IDF_Q values, next to the TF_Q entries:
  $TFIDF_{lineup} = TF_Q \bowtie IDF_Q$.
- Aggregate the *tf* and *idf* into terms of *tf* . *idf* for each term-document pair:
  $TFIDF =$
  $SELECT\ term,\ document,\ tf \times idf$
  $FROM\ TFIDF_{lineup}$.
- Compute the documents ranking, by aggregation all the terms for each document:
  $RANK =$
  $SELECT\ document,\ AGGR(tfidf)$
  $FROM\ TFIDF$
  $GROUP\ BY\ document$.

3.3 Integration issues

As mentioned before, to realize the integration of concept-based search with content-based IR, the framework offered by the Mirror MM-DBMS is used. Looking at the above presented procedure for content-based querying of single multimedia objects, some minor changes had to be made.
In Mirror the inverse document frequency *idf* is calculated over the entire set of one multimedia type, i.e. referring to the procedure again, the entire set of documents. This is useful, if the queries are always formulated over the entire set. But when querying a webspace a different query-strategy is followed. In that case the set of multimedia objects, used as the input for the content-based querying is already drastically reduced, due to the conceptual requirements of the query. For instance, when a query involves the concept class `Abstract` and one has specified some keywords which should be contained in the text of the abstract, then the conceptual requirements have already reduced the set, containing all text-fragments (`Abstract.section`), to those fragments, which are referred to by `Abstract` objects. Therefore we have chosen to compute the *idf* values not over the entire set of a multimedia type. Instead these values are calculated over subsets, based on the class concepts, to which the multimedia type belongs. Resulting in a horizontal fragmentation of the set of one multimedia type.

Discussion

Such a horizontal fragmentation can have both a positive and negative influence on the retrieval. Negative, if calculating the *idf* values over smaller sets of data, will enhance the amount of noise. On the other hand it will have a positive effect on the retrieval if the fragmentation is more or less categorizing the data based on the context. We expect the latter to be the case, when fragmenting the data, based on the conceptual information, supplied by the webspace method. For instance, the term used in text-fragments of the class `Abstract` will be characterized differently than the terms used to describe an `Attraction`[2]. For our future work, we have to prove this with some experimental results.

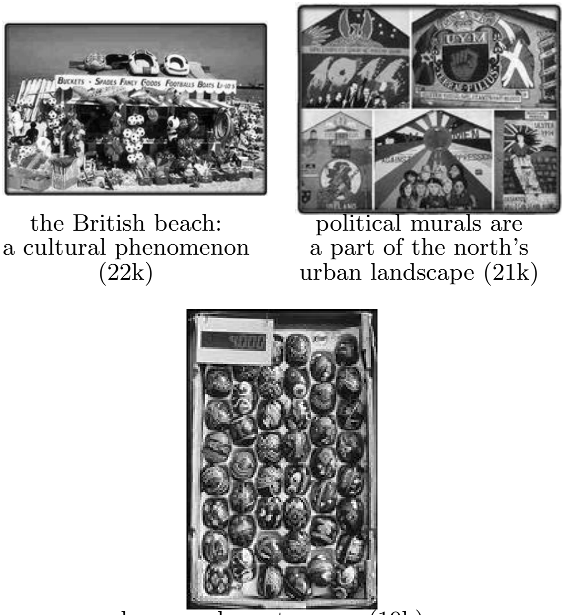
3.4 Querying a webspace

In this section we will go through two queries, which can typically be composed, when querying a webspace. We base the queries on our Lonely Planet webspace. The first query we discuss, shows how content-based querying of images is integrated in the webspace system. The SQL-query shown in Table 1.a queries the images contained in the webspace, based on their similarity. The similarity is based on the distance between the rgb- and hsb-histograms derived from the images, given a sample image. The query results are displayed in Table 1.b. The left most image represents the sample image, used as input for the query.

The second query is a bit more complex. It illustrates the power of combining conceptual search with content-based retrieval. The query is based on a fragment of the webspace schema for the Lonely Planet webspace given in Figure 2. It shows the classes `Destination`, `Region`, `Country`, `City`, and `Postcard`[3]. These classes are used in two different types of documents within the webspace. The boxes I and II show which classes are used in which of the two types of documents. Both boxes share the class `Destination`, meaning that in both documents destination-objects are defined. The query shown in Table 2.a searches for those postcards, coming from destinations, that lay inside the region with name 'Asia'. The Postcard should also contain the words 'station' and 'railway'. This query combines conceptual information stored in two different types of documents, and also shows the integration with content-based text retrieval. Table 2.b shows the resulting top 10 document-urls and their ranking.

```
SELECT i.src, i.caption FROM Image i
WHERE
 similar(i,'http://.../eur/graphics/bri27.jpg') < 0.35
ORDER BY
 similar(i,'http://.../eur/graphics/bri27.jpg');
```

(a) Content based image query.



the British beach:
a cultural phenomenon
(22k)

political murals are
a part of the north's
urban landscape (21k)

homemade easter eggs (19k)

(b) Query results.

Table 1: Content-based retrieval on images.

4. RETRIEVING THE META-DATA: TOWARDS AN IMPLEMENTATION

Going towards an implementation, we will discuss in this section how meta-data for both the concept-based search and content-based information retrieval is derived from the XML documents. Figure 3 shows the component responsible for this task in more detail. Once it is initiated with a webspace schema and a Daemon Data Dictionary (DDD) it will (1) generate the schemas used by the object server,

[2] These concepts are both defined for our Lonely Planet example. In Figure 5 a fragment of this webspace schema is shown.

[3] The Figure includes the class `tfidf_Postcard`, which contains the attributes `id`, `term`, and the corresponding `tfidf-value`. The current collection contains 4308 postcards

```
SELECT p.document, believe(p,′station′,′railway′) AS ranking FROM postcard p
WHERE p.destination_id IN
  (SELECT d.id
   FROMdestination ∗ d
   WHERE d.id IN
    (SELECT r.destination_id FROM insidei, region r
     WHERE r.name =′ asia′ AND r.id = i.region_id)
  )
ORDER BY rankingDESC;
```

(a) Query combining conceptual search and content-based IR.

document	ranking
http : //waterfall/lonely.planet/pc/nea/tai_pc36.xml	7.2931
http : //waterfall/lonely.planet/pc/sea/indo_pc96.xml	6.6659
http : //waterfall/lonely.planet/pc/nea/chi_pc201.xml	6.0853
http : //waterfall/lonely.planet/pc/nea/jap_pc6.xml	6.0853
http : //waterfall/lonely.planet/pc/nea/tai_pc13.xml	6.0853
http : //waterfall/lonely.planet/pc/nea/tai_pc33.xml	6.08536
http : //waterfall/lonely.planet/pc/cas/tur_pc23.xml	5.4582
http : //waterfall/lonely.planet/pc/nea/chi_pc122.xml	4.8776
http : //waterfall/lonely.planet/pc/sea/mal_pc97.xml	4.8776
http : //waterfall/lonely.planet/pc/nea/chi_pc129.xml	4.2504

(b) Query results.

Table 2: the postcard query.

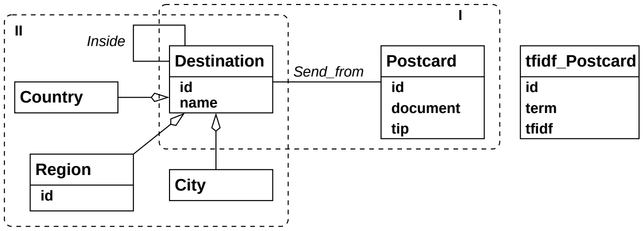


Figure 2: Fragment of webspace schema for Lonely Planet webspace.

and next (2) it will start mapping XML-documents, which are seen as materialized views on the webspace schema, in order to obtain web objects. This is explained in detail in Section 4.1. Finally, (3) the meta-data is retrieved from these web objects and stored in the object server, using the DDD (Section 4.2).

4.1 From XML to web object

We will illustrate by an example how web objects are derived from XML documents. Figure 4 shows a graphical representation of the DTD for one of our XML documents, based on our Lonely Planet example. The root element *d*estination corresponds with the class concept `Destination` of the webspace schema fragment shown in Figure 5. The attributes of this root element (*name, keywords*), are also modelled as attributes of the class `Destination`. Parsing the child-elements of the root-element *d*estination several associated web objects are found, referring to the class concepts `Abstract`, `Attraction`, and others. The element *has_facts* models the association concept `has_facts` and explicitly defines the relationship between `Destination` and `Fact`, while for the other concepts the relationship is defined implicitly. Once all the available properties of a web object are found it is handed to the DDD.

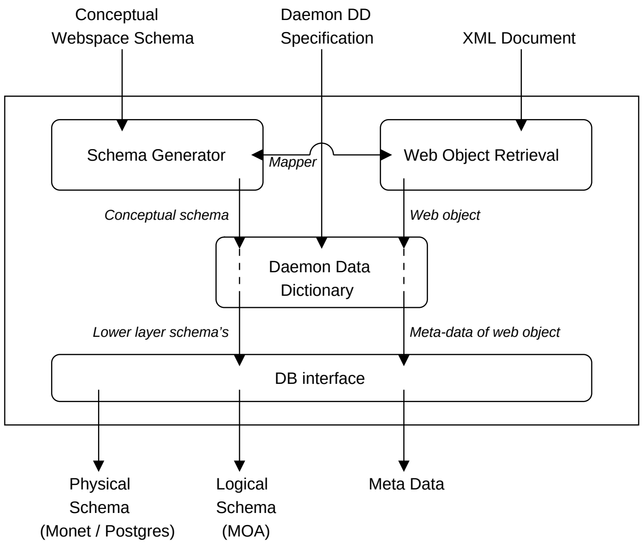


Figure 3: Web object retrieval and meta-data extraction

4.2 From web object to meta-data

Once the web objects are obtained, the meta data should be retrieved from the objects. For this purpose we use the DDD. Each daemon, administrated in the DDD performs a single task, on a single type of multimedia. To be able to perform this task correctly it uses three methods, the **initialize**, **getWork**, and **finalize**. During the initialize phase, a daemon can alter the webspace schema and the schemas used for the object server, such that the meta-data retrieved by that daemon is stored properly in the database. The get-

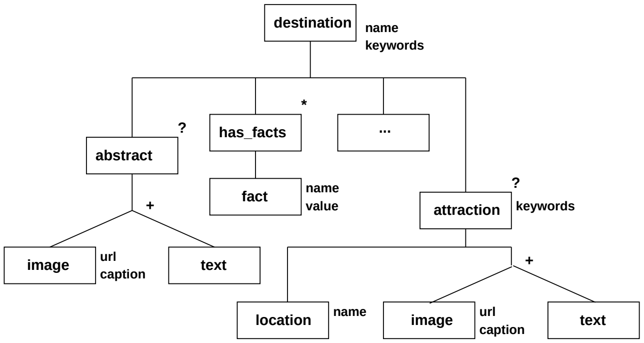


Figure 4: DTD specification of Destination pages.

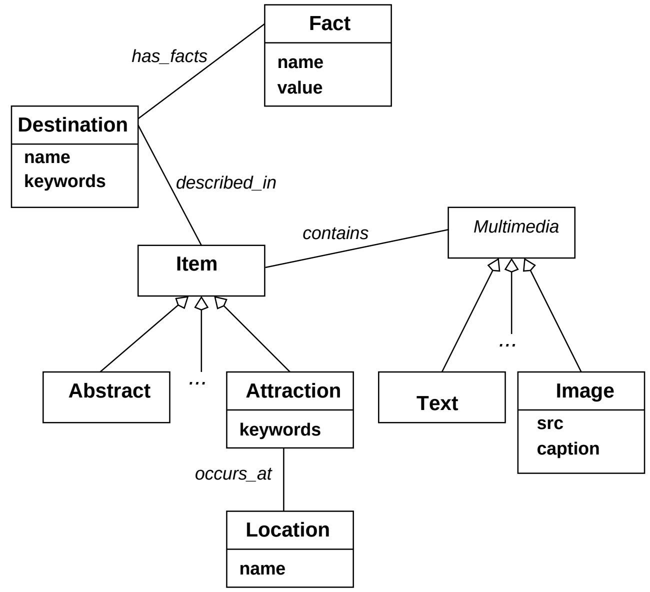


Figure 5: Fragment of Webspace schema for Lonely Planet case.

Work is responsible for retrieving and storing the meta-data given a web object. Finally, once all the data is obtained, the finalize-method performs some post-processing steps on the database. e.g calculating *idf* values over a set of multimedia objects.

Daemon Data Dictionary

In the DDD all the available daemons are administrated. Once a web object is handed to the DDD it uses a trigger mechanism to determine which daemons should work on that object. Before a new mapping session starts, the DDD is loaded with a number of daemons, which should be called sequentially. For this purpose we use XML as an exchange format, as shown in Table 3. There is specified which daemons to load and what triggers the DDD uses to start a daemon. This approach allows us to dynamically load a new daemon, whenever it becomes available.

5. CONCLUSIONS

This paper presents (1) the webspace method for modelling large collections of documents containing related information, using existing database technology, and (2) the integration of the webspace method with content-based information retrieval. The webspace method distinguishes two levels, at which a webspace should be defined. Besides a document level, where the document, specified in XML can be found, we also identified a semantical level, where concepts, derived from the documents are modelled in terms of an object-oriented schema. This schema forms the basis for concept-based search over a webspace.
The integration with content-based IR is essential, in that sense that it offers the user not only to query a webspace by its concepts. But it also allows them to take the content of the webspace into account. We have shown how this integration with content-based multimedia IR is achieved, by adopting a commonly used IR model, together with the Mirror framework for multimedia databases. Making some minor changes, concerning the retrieval method, allowed us to realize the integration. For our future research we still need to do some retrieval performance evaluations. Further improvements can certainly be made, by adopting more sophisticated IR models. These models can easily be integrated into our approach. To realize this we only need to administer a new daemon to the daemon data dictionary, which implements the new IR model.

```
<?xml version = "1.0" encoding = "UTF − 8"? >
<!doctype ddd system "./ddd.dtd" >
< ddd >
  < daemon
      class = "java.daemon.url.URLDaemon" >
    < trigger string = " ∗ .src. ∗ "/ >
    < trigger string = " ∗ .document. ∗ "/ >
  ...
  < /daemon >
  < daemon
      class = "java.daemon.image.rgb.RGBDaemon" >
    < trigger string = "Image.src. ∗ "/ >
  < /daemon >
  < daemon
      class = "java.daemon.text.TEXTDaemon" >
    < trigger string = " ∗ .section.text"/ >
    < trigger string = "Postcard.tip.text"/ >
  ...
  < /daemon >
< /ddd >
```

Table 3: XML exchange: daemon data dictionary

6. ACKNOWLEDGMENTS

We wish to thank Arjen de Vries and Erik van het Hof for their work on content-based multimedia databases. They supplied the basic ideas for our daemon architecture for content-based information retrieval. We also wish to thank Robert van Utteren, for his work on the implementation of several parts of the conceptual layer.

7. REFERENCES

[1] M. Agosti, R. Colotti, and G. Gradenigo. A two-level hypertext retrieval model for legal data. In A. Bookstein, Y. Chiaramella, G. Salton, and V. Raghavan, editors, *Proceedings of the 14th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages

316–325, Chicago, Illinois, Oct. 1991. ACM Press.

[2] G. Arocena and A. Mendelzon. WebOQL: Restructuring documents, databases and webs. In *proceedings of Fourteenth IEEE International Conference on Data Engineering (ICDE98)*, 1998.

[3] P. A. Boncz, A. N. Wilschut, and M. L. Kersten. Flattening an Object Algebra to Provide Performance. In *Proceedings of the IEEE International Conference on Data Engineering (ICDE)*, pages 568–577, Orlando, FL, USA, February 1998.

[4] S. Ceri, S. Comai, E. Damiani, P. Fraternali, S. Paraboschi, and L. Tanca. Xml-gl: a graphical language for querying and restructuring XML documents. In *proceedings of the International World Wide Web Conference (WWW)*, pages 1171–1187, 1999.

[5] S. Ceri, P. Fraternali, and A. Bongio. Web modeling language (WebML): a modeling language for designing web sites. In *proceedings of WWW9*, Amsterdam, the Netherlands, May 2000.

[6] A. Deutsch, M. F. Fernandez, D. Florescu, A. Levy, and D. Suciu. A query language for XML. In *proceedings of the International World Wide Web Conference (WWW)*, pages 1155–1169, 1999.

[7] M. Fernandez, D. Florescu, J. Kang, A. Levy, and D. Suciu. Catching the boat with Strudel: Experiences with a web-site management system. In *proceedings of ACM SIGMOD Conference on Management of Data*, Seattle, WA, 1997.

[8] D. Florescu and D. Kossmann. A performance evaluation of alternative mapping schemes for storing XML data in a relational database. Technical report, INRIA, Rocquencourt, May 1999.

[9] D. Florescu, I. Manolescu, and D. Kossmann. Integrating keyword search into xml query processing. In *proceedings of the ninth international WWW Conference*, Amsterdam, the Netherlands, May 2000.

[10] R. Goldman, J. McHugh, and J. Widom. From semistructured data to xml: Migrating the lore data model and query language. In *proceedings of the 2nd International Workshop on the Web and Databases (WebDB '99),*, Philadelphia, Pennsylvania, June 1999.

[11] D. Hiemstra and W. Kraaij. Twenty_one at trec-7: Ad-hoc and cross-language track. In *proceeding of the seventh Text Retrieval Conference TREC-7*, 1999.

[12] G. Mecca, P. Merialdo, and P. Atzeni. Araneus in the era of xml. *IEEE Data Engineering Bullettin, Special Issue on XML*, Sept. 1999.

[13] G. Mecca, P. Merialdo, P. Atzeni, and V. Crescenzi. The Araneus guide to web-site development. Technical report, Dipartimento di Informatica e Automazione, Universita' di Roma Tre, Mar. 1999.

[14] A. Mendelzon, G. Mihaila, and T. Milo. Querying the world wide web. *Journal of Digital Libraries*, pages 1(1):54–67, Apr. 1997.

[15] A. S. On views and xml. *symposium on Principles of Database Systems (PODS'99)*, May 1999.

[16] A. Schmidt, M. Kersten, M. Windhouwer, and F. Waas. Efficient relational storage and retrieval of xml documents. In *International Workshop on the Web and Databases*, Dallas TX, USA, May 2000.

[17] D. Schwabe and G. Rossi. Developing hypermedia applications using oohdm. In *Workshop on Hypermedia Development Processes, Methods and Models, Hypertext'98,*, Pittsburgh, USA, 1998.

[18] M. Stonebraker and G. Kemnitz. The POSTGRES next generation database management system. *Commun. ACM 34*, (10):pages 78 – 92, Oct. 1991.

[19] A. d. Vries. *Content and multimedia database management systems*. PhD thesis, University of Twente, Enschede, The Netherlands, Dec. 1999.

[20] A. d. Vries and A. Wilschut. On the integration of ir and databases. In *Database issues in multimedia; short paper proceedings, international conference on database semantics (DS-8)*, 1999.

[21] W3C. Extensible markup language (XML). Technical report, World Wide Web Consortium (W3C), Feb. 1998.

[22] R. v. Zwol and P. Apers. Modelling the webspace of an intranet. In *proceeding of 1st international conference on Web Information Systems Engineering (WISE00)*, Hong Kong, June 2000.

[23] R. v. Zwol and P. Apers. Using webspaces to model document collections on the web. In *proceedings of WWW and Conceptual Modelling(WCM00), in conjunction with ER2000*, Salt Lake City, Oct. 2000.

[24] R. v. Zwol, P. Apers, and A. Wilschut. Modelling and querying semistructured data with Moa. In *proceedings of Workshop on Query Processing for Semistructured Data and Non-standard Data Formats*, Jerusalem, Israel, Jan. 1999.