Retrieving Information in Distributed Multimedia Databases

Christoph Baumgarten, Ulrich Marder, Günter Robbert,
Klaus Meyer-Wegener *

Dresden University of Technology
Germany

Abstract

In this paper a new model and architecture for information retrieval in a widely distributed heterogenous multimedia document collection is described. The model generalizes existing probabilistic models for non-distributed information retrieval. The architecture is a conceptual realization of this model. It is hierarchically built in order to provide extendability and scalability and designed to integrate existing dynamic multimedia databases.

Keywords: Information Retrieval, Multimedia databases, Probabilistic Models, Distributed Systems

1 Introduction

The internet provides access to a large amount of data which is growing from day to day. Most of the data are simple text documents, but the fraction of multimedia data like image, audio and video documents is increasing rapidly. Hence, for users, it is getting more and more difficult to find documents containing relevant information. This is true especially for multimedia documents, because often one cannot clearly decide whether a document is relevant or non-relevant. That means, traditional boolean search techniques are not sufficient in this case and therefore have to be replaced by search techniques which enable the handling of 'vague' queries. Such techiques have been developed within the research area of Information Retrieval (IR) during the last decades. However, most of them are restricted to non-distributed text databases. Extending them to be applicable to distributed multimedia databases is still a challenge.

The remainder of this paper is organized as follows: In section 2 different approaches to distributed information retrieval are introduced. Discussing these approaches yields the motivation for our *probabilistic model* for information retrieval in a distributed multimedia document collection, described in section 3. This model serves as the basis for a distributed scalable IR system. The IR system's open hierarchical architecture, sketched in section 4, allows to integrate (almost) any number of autonomously managed, highly dynamical databases. Databases can be based on file systems, commercial DBMSs, multimedia data servers etc. Section 5 concludes the paper.

2 Distributed information retrieval

For querying traditional database systems usually boolean query languages are used. Therefore, it is absolutely deterministic which data belong to a query's answer set and which do not. SQL or QUEL for relational database systems are examples of such query languages. If we consider multimedia data like images, audio, and video sequences, but also texts, it turns out that finding relevant information using boolean query languages becomes very difficult: To enable content based searching for multimedia data *indexing* the data items is indispensable. However, such descriptions are always incomplete and highly dependent on the indexing methods being used. Moreover, when indexing manually this process is almost certainly influenced by the opinion and knowledge of the indexing person(s). Searching these descriptions using boolean queries is often

*E-mail: {baumgart, marder, robbert, kmw}@is2201.inf.tu-dresden.de

ineffective, because data items matching the query terms only in parts will not belong to the answer set. It is also difficult for a user to specify an information need in terms of a boolean language, if the underlying database consists of multimedia data. Consequently, information retrieval systems for multimedia data should be able to deal with 'vague' queries.

Such methods have already been developed in the research field of information retrieval. Instead of computing fixed answer sets, IR systems sort the whole *document collection* with respect to a query by assigning a so-called *retrieval status value* (RSV) to each document. The RSV determines a document's position within a *ranking list* which is returned to the user as an answer to his query. Of course, this relevance assessment — usually called the *ranking* of a document — is only an approximation of a document's real degree of relevance (or non-relevance) to a query. After having examined some of the retrieved documents, the user can mark them as being relevant or not. This *relevance feedback* can be exploited by the IR system to compute an improved ranking list.

In the past, IR methods concentrated on text retrieval. Recently, some methods have been extended for multimedia information retrieval. Most of these approaches combine indexing methods for multimedia documents. These methods create descriptions that can be used with conventional IR methods.

The ongoing expansion of the internet enables on-line access to a continuously increasing number of multimedia documents. It is an obvious thing to extend conventional IR methods which only consider locally accessible documents for distributed heterogeneous document databases. In principle the entire data volume can grow to any size. However, response time of an IR system should be kept limited. That means, the IR system has to be able to make a relevance assessment (with respect to a query) for a whole document database in order to decide the posting of the query to that database. This should be done automatically, because manually choosing document databases overstrains a user.

Another problem is that the computation of the ranking lists of different document databases is based on variable contexts (constituted by only locally available statistical data). That means, two documents being stored in the same database and obtaining the same RSV would gain different RSVs when being stored in different databases. Thus, ranking lists from different document databases cannot be merged without prior normalization (the *collection fusion problem* [VGJL94]).

The statistical data on which the local ranking is based should be propagated in order to achieve an efficiency comparable to non-distributed IR. With other words, the dissemination of statistical data is necessary in order to influence the local ranking by statistical data from other databases.

Some approaches for information retrieval in distributed document databases already exist. [CZC95] for instance, employs a so-called inference network for distributed IR. However, this approach assumes globally standardized methods for indexing and the computation of the RSVs. Therefore, its applicability to multimedia document retrieval is limited. Moreover, the underlying data volume can only be scaled to a certain extent. The problem of normalizing the RSVs is solved in a heuristic way.

The latter is also true for the approach proposed in [VGJL94]. Here, for selecting document databases, the (stored) results of previously evaluated queries are exploited. However, this seems to work well only for more or less static databases.

The MEDOC Project [BDGj96] is working on a system for information retrieval including a huge number of distributed heterogenous text- and literature-databases. It is assumed that an integrated database has its own ranking functionality which estimates the RSV of a document as the probabilitiy of the document's relevance. This way, the normalization of RSVs from different databases is not required, RSVs may simply be interleaved (we will use a similar assumtion in the following). The dissemination of statistical data is not supported. The algorithm for selecting databases [Fuh96a] is based on certain cost factors; the aim is to receive the maximum number of relevant documents for minimal costs.

Our approach being presented in the following sections attempts to fullfil the given requirements while overcoming most of other approach's limitations.

3 A probabilistic model for distributed IR

In this section a new formally founded *probabilistic model* for IR in a widely distributed document collection is introduced. The major properties of this model are:

- coherent ranking of the distributed documents by also considering non-local statistical data,

- the retrieval process can be automatically limited to parts of the distributed document collections while minimizing the loss of relevant documents,
- different methods for indexing and computing RSVs can be employed in order to support the integration of heterogeneous multimedia document collections (the latter have to be based on probabilistic models for non-distributed IR),
- relevance feedback is utilized, and
- the model takes unlimited growth of the data volume into consideration (i. e. it can be realized by a scalable system architecture).

Probabilistic models like the *binary independence retrieval* model (BIR) [RSJ76] [Sch96] for non-distributed IR are based on the so-called *probability ranking principle* (PRP) [Rob77]: This principle states that presenting the documents to the user in decreasing order of their probability of relevance with respect to a certain information need is optimum. Using an order-preserving transformation, it is possible to reduce these probabilities of relevance to probabilities which can be estimated by means of statistical data and — if available — relevance feedback.

If documents and queries are interpreted as *events* which occur with a certain probability, then the probability of relevance of a document d with respect to a query q can be denoted as the conditional probability $P(R|d,q)$. In accordance to the literature on probabilistic IR we use the following abbreviations:

$$\begin{aligned} E(X,Y) &:= E(X) \cap E(Y) \\ P(X) &:= P(E(X)) \\ P(X|Y) &:= P(E(X)|E(Y)) \end{aligned}$$

where $E(\cdot)$ denotes an event. Hence, $P(R|d,q)$ means the probability of the relevance event $E(R)$ provided that both events $E(d)$ and $E(q)$ occur together.

We define

$$f(x) := \frac{x}{1-x} \qquad \text{and} \qquad g(q) := \frac{P(\overline{R}|q)}{P(R|q)}$$

where $\overline{R}$ means the event *not* R, such that

$$RSV(d,q) := f(P(R|d,q))g(q)$$

is an order-preserving query-dependent transformation of $P(R|d,q)$ which can be simplified as follows:

$$\begin{aligned} RSV(d,q) &= \frac{P(R|d,q)}{P(\overline{R}|d,q)} \frac{P(\overline{R}|q)}{P(R|q)} \\ &= \frac{P(d|R,q)}{P(d|\overline{R},q)} \frac{P(R|q)}{P(\overline{R}|q)} \frac{P(\overline{R}|q)}{P(R|q)} \\ &= \frac{P(d|R,q)}{P(d|\overline{R},q)} \end{aligned} \tag{1}$$

In order to model a distributed document collection we assume that the collection is divided *hierarchically* into *disjunct sub-collections*. That means, a sub-collection either contains subordinate sub-collections or — if it belongs to the lowest layer of the hierarchy — documents.

A sub-collection D^i of layer i is — just like documents — considered as the event

$$E(D^i) := \bigcup_{D^{i-1} \in D^i} E(D^{i-1}).$$

For $i = 1$ we define $D^{i-1} = D^0 := d$.

Let n be the number of layers of the hierarchy. Then (1) can be expanded as follows:

$$RSV(d,q) = \frac{P(d|R,q)}{P(d|\overline{R},q)}$$

$$
\begin{aligned}
&= \frac{P(d|D^1,R,q)}{P(d|D^1,\overline{R},q)} \frac{P(D^1|R,q)}{P(D^1|\overline{R},q)} \\
&= \frac{P(d|D^1,R,q)}{P(d|D^1,\overline{R},q)} \frac{P(D^1|D^2,R,q)}{P(D^1|D^2,\overline{R},q)} \frac{P(D^2|R,q)}{P(D^2|\overline{R},q)} \\
&\vdots \\
&= \left(\prod_{i=1\ldots n} \frac{P(D^{i-1}|D^i,R,q)}{P(D^{i-1}|D^i,\overline{R},q)} \right) \frac{P(D^n|R,q)}{P(D^n|\overline{R},q)} \\
&= \prod_{i=1\ldots n} \frac{P(D^{i-1}|D^i,R,q)}{P(D^{i-1}|D^i,\overline{R},q)}\,, \qquad (2)
\end{aligned}
$$

if we assume exactly one sub-collection D^n at the highest layer.

In order to estimate the expression $\frac{P(D^{i-1}|D^i,R,q)}{P(D^{i-1}|D^i,\overline{R},q)}$, we make the assumption in analogy to the non-distributed BIR model (other more efficient, but also more complicated non-distributed probabilistic models as e. g. [RW94] could have been used here as well) that:

1. Documents or sub-collections D^{i-1} which are included in a sub-collection D^i are indexed by *features*. These descriptions $\Phi^i(D^{i-1})$ — also called feature sets — are taken from an indexing vocabulary Φ^i which is assigned to D^i. A feature $\varphi \in \Phi^i$ is considered to be the event
$$E(\varphi) = \bigcup_{D^{i-1}:\varphi\in\Phi^i(D^{i-1})} E(D^{i-1})\,.$$
2. For a query q, there also exists a description $\Phi^i(q)$. Features in $\Phi^i(q)$ are called *query features*.
3. The D^{i-1} *within* D^i can be uniquely identified by the features in $\Phi^i(q)$. From this follows
$$E(D^{i-1}) = \left(\bigcap_{\varphi\in\Phi^i(q)\cap\Phi^i(D^{i-1})} E(\varphi) \right) \cap \left(\bigcap_{\varphi\in\Phi^i(q)-\Phi^i(D^{i-1})} \overline{E(\varphi)} \right)\,. \qquad (3)$$
4. There exists a so-called *linked dependence* between the query features. Using (3), this yields
$$P(D^{i-1}|D^i,R,q) = C \prod_{\varphi\in\Phi^i(q)\cap\Phi^i(D^{i-1})} P(\varphi|D^i,R,q) \prod_{\varphi\in\Phi^i(q)-\Phi^i(D^{i-1})} P(\overline{\varphi}|D^i,R,q)$$
and
$$P(D^{i-1}|D^i,\overline{R},q) = C \prod_{\varphi\in\Phi^i(q)\cap\Phi^i(D^{i-1})} P(\varphi|D^i,\overline{R},q) \prod_{\varphi\in\Phi^i(q)-\Phi^i(D^{i-1})} P(\overline{\varphi}|D^i,\overline{R},q)\,,$$
where $C \in \mathbb{R}^+$ denotes a *common* constant.

   Note that without applying the transformation f we would have been forced to assume that the query features are completely independent. In reality this assumption would have been violated much more frequently than the linked dependence assumption which is made here.

Combining the previous leads to

$$\frac{P(D^{i-1}|D^i,R,q)}{P(D^{i-1}|D^i,\overline{R},q)} = \prod_{\varphi\in\Phi^i(q)\cap\Phi^i(D^{i-1})} \frac{P(\varphi|D^i,R,q)}{P(\varphi|D^i,\overline{R},q)} \prod_{\varphi\in\Phi^i(q)-\Phi^i(D^{i-1})} \frac{P(\overline{\varphi}|D^i,R,q)}{P(\overline{\varphi}|D^i,\overline{R},q)}\,.$$

In contrast to the BIR model this expression can not be simplified by another order-preserving transformation, because that would prevent RSV normalization which is required for merging the RSVs gained from different sub-collections.

Instead, the following approach is chosen: We assume a subset $\tilde{D}^i \subseteq D^i$ of documents for which the user has decided whether they are relevant to him or not (relevance feedback). That means, $\tilde{D}^i$ is the union of a set $\tilde{D}^i_{rel}$ of documents which are relevant to the user and a set $\tilde{D}^i_{non}$ of documents which are not relevant. Then $\frac{P(\varphi|D^i,R,q)}{P(\varphi|D^i,\overline{R},q)}$ can be estimated through the approximation

$$\frac{\tilde{D}^i_{rel}(\varphi)}{\tilde{D}^i_{non}(\varphi)} \frac{|\tilde{D}^i_{non}|}{|\tilde{D}^i_{rel}|} \approx \frac{\epsilon + \tilde{D}^i_{rel}(\varphi)}{\epsilon + \tilde{D}^i_{non}(\varphi)} \frac{\epsilon + |\tilde{D}^i_{non}|}{\epsilon + |\tilde{D}^i_{rel}|}, \epsilon \in \mathbb{R}, \tag{4}$$

where $D(\varphi)$ counts the occurrances of φ within the descriptions of the documents or sub-collections in D, respectively. Extending the terms by $\epsilon \overset{\text{e.g.}}{=} \frac{1}{2}$ allows us to handle the situation where no relevance information from the user is available (because $\frac{0}{0}$ is avoided): Then $\tilde{D}^i_{rel} = \emptyset$, $\tilde{D}^i_{non} = D^i$ holds and we obtain

$$\frac{P(\varphi|D^i,R,q)}{P(\varphi|D^i,\overline{R},q)} \approx \frac{\epsilon + |D^i|}{\epsilon + D^i(\varphi)}. \tag{5}$$

$\frac{P(\overline{\varphi}|D^i,R,q)}{P(\overline{\varphi}|D^i,\overline{R},q)}$ can be estimated analogously replacing $D(\varphi)$ by $|D| - D(\varphi)$ in (4) and (5).

Remarks:

1. Usually, $\tilde{D}^1$ results from the first few documents of a ranking list which was in a first pass computed without relevance feedback. It would be nice to be able to compute $\tilde{D}^i_{rel}$ resp. $\tilde{D}^i_{non}$, $i > 1$, from $\tilde{D}^1_{rel}$ resp. $\tilde{D}^1_{non}$. Such an algorithm could be derived from experimental results.
2. The indexing of sub-collections constitutes the local statistical data being disseminated to the distributed environment. Thus, the influence of non-local statistical data to the computation of the RSVs is given by Equation (2).
3. Obviously, documents can be considered as sub-collections of their parts (e. g. chapters, pages, images, video sequences etc.). Thus, the model described here also covers information retrieval on distributed, *hierarchically structured documents*. However, this aspect has not been further investigated yet.

The model described so far enables to perform a complete ranking of a distributed document collection. However, in order to increase the efficiency of the retrieval process, it should be possible to exclude some parts of the collection (i. e. sub-collections) from this process. An effective criterion (in the sense of the PRP) is required, that can be used for selecting sub-collections which are likely to contain many relevant documents. If we interpret the probabilities of relevance $P(R|d,q)$, $d \in D^1 \in \ldots \in D^m$, $m \geq 1$, as a *random variable*

$$\begin{aligned} P^m(q) \quad : \quad & \{d | d \in D^1 \in \ldots \in D^m\} \longrightarrow [0;1] \\ & d \longmapsto P(R|d,q)\,, \end{aligned}$$

then

$$RSV(D^m,q) \overset{(2)}{=} \prod_{i=m+1\ldots n} \frac{P(D^{i-1}|D^i,R,q)}{P(D^{i-1}|D^i,\overline{R},q)}$$

determines the *expectation value* μ^m of $P^m(q)$ transformed by f and g because of

$$\begin{aligned} RSV(D^m,q) &= \frac{P(R|D^m,q)}{P(\overline{R}|D^m,q)} g(q) \\ &= \frac{\sum\limits_{D^{m-1} \in D^m} P(R|D^{m-1},q) P(D^{m-1}|D^m,q)}{\sum\limits_{D^{m-1} \in D^m} P(\overline{R}|D^{m-1},q) P(D^{m-1}|D^m,q)} g(q) \\ &= \frac{\sum\limits_{D^{m-1} \in D^m} \left(\sum\limits_{D^{m-2} \in D^{m-1}} P(R|D^{m-2},q) P(D^{m-2}|D^{m-1},q) \right) P(D^{m-1}|D^m,q)}{\sum\limits_{D^{m-1} \in D^m} \left(\sum\limits_{D^{m-2} \in D^{m-1}} P(\overline{R}|D^{m-2},q) P(D^{m-2}|D^{m-1},q) \right) P(D^{m-1}|D^m,q)} g(q) \end{aligned}$$

$$
\begin{aligned}
&= \frac{\sum\limits_{D^{m-2}\in D^{m-1}\in D^m} P(R|D^{m-2},q)P(D^{m-2}|D^m,q)}{\sum\limits_{D^{m-2}\in D^{m-1}\in D^m} P(\overline{R}|D^{m-2},q)P(D^{m-2}|D^m,q)} g(q) \\
&\vdots \\
&= \frac{\sum\limits_{d\in\ldots\in D^m} P(R|d,q)P(d|D^m,q)}{\sum\limits_{d\in\ldots\in D^m} P(\overline{R}|d,q)P(d|D^m,q)} g(q) \\
&=: f(\mu^m)g(q)\,.
\end{aligned}
$$

If we were able to determine the corresponding (possibly transformed) *variance* $(\sigma^m)^2$ in a similar way (i. e. by exploiting information which is available at layer m), we could formulate the wanted criterion for selecting sub-collections at layer m. This criterion could be extended by taking certain *cost factors* into consideration [Fuh96a].

4 Architecture of the distributed IR system

The probabilistic model described in the previous section provides the basis for an IR system for distributed multimedia databases. The IR system consists of a number of hierarchically organized nodes (cf. Figure 1). Usually, these nodes will be geographically distributed. Communication between nodes can be realized e. g. by using CORBA as shown in [TMRMW96]. Nodes on the lowest layer of the hierarchy provide retrieval functionality for a *media object database* whereas nodes on the higher layers provide retrieval functionality for a set of sub-collections represented by nodes on the next lower layer. By using the notion 'media object database' we do not only refer to multimedia database systems but also to traditional database systems or even simple file systems. Some media object databases offer some sort of retrieval functions. As a rule, such functions are not used by the IR system, because each node is supplied with its own retrieval functionality satisfying the requirements of our probabilistic model.

A node on layer i consists of the following components:

- *IR server*: An IR server manages features of a large number of objects (see below). These features are taken from the indexing vocabulary assigned to the IR server. The server also registers the feature frequencies. According to the model the whole set of objects indexed by one IR server is called a sub-collection. A node may have several indexing vocabularies and hence, several IR servers or sub-collections, respectively. The IR server also performs the computation of the RSVs for a query received from a client of the next higher layer $i+1$: For each query a query object is created [1] and indexed by executing the appropriate indexing method (which is the one that uses the indexing vocabulary assigned to the IR server). This way, a set of query features is obtained which is needed to compute the RSVs. This results in a ranking list of objects which is shortened with respect to a selection criterion. If $i>1$ the IR server then passes the query on to the sub-collections (IR servers resp.) appearing in the modified ranking list. From these servers it will eventually receive partial results which are merged together and passed back to the client.

  As IR server we will prospectively employ the non-scalable IR system SPIDER [Sch93] developed at the Swiss Federal Institute of Technology (ETH) Zurich. SPIDER consists of an IR server and a synchronizer. The synchronizer has to keep the indexing of a dynamic database up to date (maybe slightly delayed, see below) [KS96].

- *Adapter*: An adapter provides either the data of an integrated media object database or several sub-collections of the next lower layer as *objects*. These objects are instances of *object classes* which define *object methods* like e. g. methods for indexing objects. Each object class must also define a method for creating query objects which can be indexed to extract the appropriate query features (cf. also [Fuh96b]).

  (a) $i=1$: A *media object adapter* provides the documents or *retrievable items*, resp., of a media object database as objects which are instances of several different media object classes M. The term

[1] This requires that the query format is accepted by an available query object creation method.

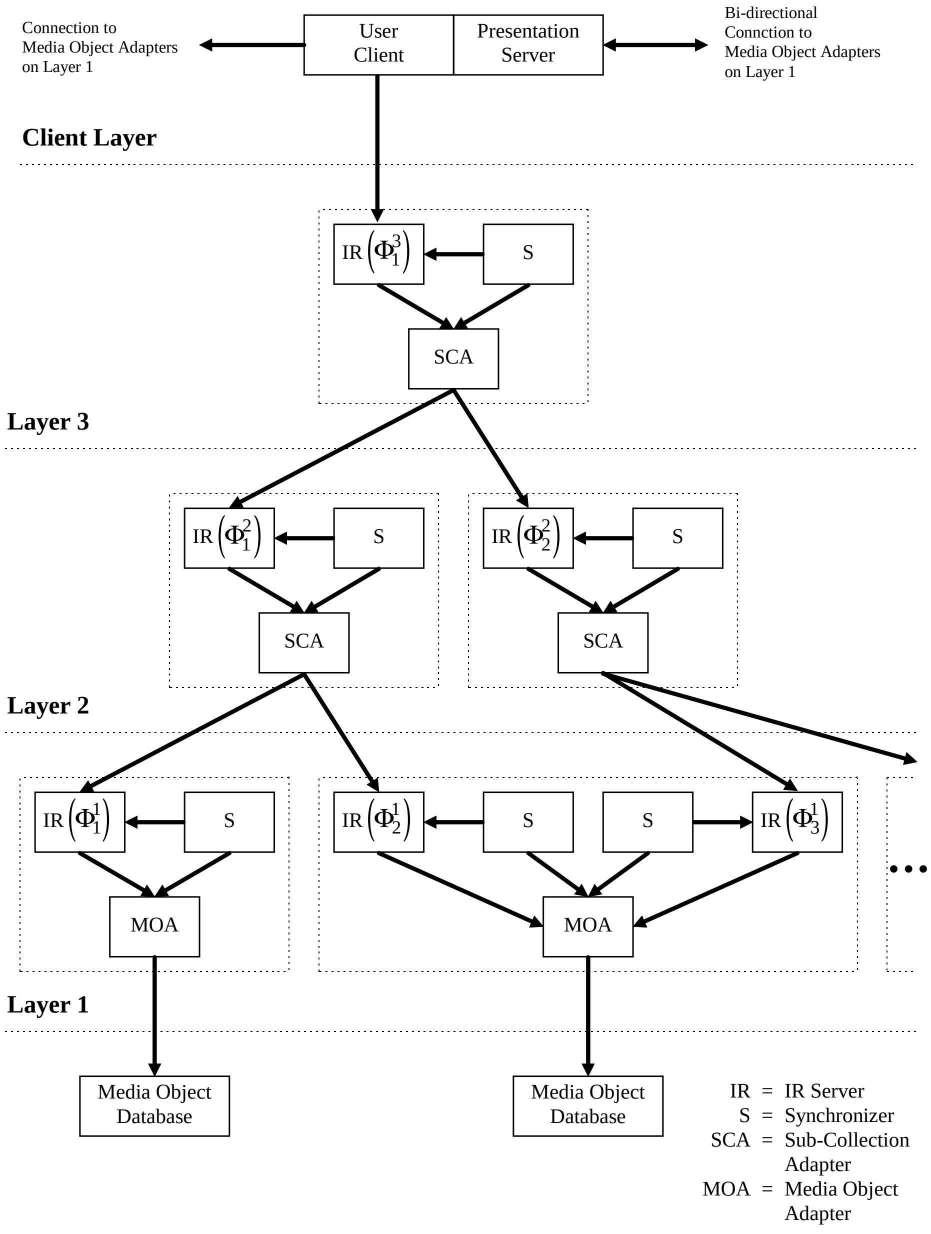

Figure 1: Architecture of the distributed IR system.

'retrievable items' should be interpreted extensively, thus, besides media objects like images, also tuples of a relational database can be retrievable items, for instance.

For objects of a class M, there must be defined a number of methods like indexing methods or presentation methods. The latter require a *presentation server* to be present on the client's workstation. The presentation server enables the client to receive, control, and play out real-time continuous data streams (e. g. audio and video data streams) and non-real-time data streams (e.g. still images). By 'control' we mean negotiation and adaption of quality of service as well as sending VCR-commands like *play, pause, fast-forward* etc.

(b) $i > 1$: A *sub-collection adapter* provides sub-collections of the underlying layer as objects. These objects are instances of sub-collection classes S with indexing methods. The state of such an object is constituted by the set of features of all elements of the corresponding sub-collection. Because this set may grow very large, the indexing method must work incrementally. That means, it only considers state changes that happened after its last execution.

Sub-collection objects can be indexed by simply using the object state (i. e. the feature set) *without modifying* it [CZC95]. However, on higher layers ($i > 2$) such indexes become larger and larger and also very dynamic. This would lead to low performance and scalability. To avoid this situation, indexing methods principally considering the *static parts* of the object state should be used. A part of the object state is characterized 'static' if it is likely to survive even if the underlying database is updated very often. Examples of static parts are meta data and features or parts of features with high frequencies within a sub-collection.

If an information provider intends to integrate a new node at layer $i-1$ into the IR system with one or more sub-collection(s), he or she needs to determine the level-i–sub-collection class(es) of the sub-collection(s). This is necessary in order to find appropriate sub-collections at layer i that would accept these sub-collection(s). We consider two sub-collections to be instances of the same sub-collection class if their indexing methods and vocabularies are similar (or equal, of course). Each sub-collection class S must define a test function that decides if a particular sub-collection is of type S or not. Such a test function requires a reference indexing vocabulary and method, a number r of reference documents and a suitable similarity measure for feature sets. To test the type of a sub-collection s all reference documents are indexed using both the reference indexing method and the indexing method of s. This results in r feature set pairs. If all pairs consist of feature sets which are eqivalent with respect to the similarity measure one can conclude that s is of type S.

In contrast to the model, the introduced adapter concept does not require sub-collections to be disjunct. An object can thoroughly be indexed using different methods. This means that it is assigned to several sub-collections. Of course, it is always possible to force sub-collections to be disjunct (by transforming non-empty intersections of sub-collections into new sub-collections). But we feel that enabling the model to handle non-disjunct sub-collections would be useful, because the number of sub-collections per node would be reduced and the process of integrating new sub-collections into the hierarchie would manage without large-scaled control mechanisms.

It is possible to have several adapters in a node, but this case is not further investigated in this paper.

- *Synchronizer*: Periodically the synchronizer uses the indexing methods provided by the adapter in order to index objects that are marked as new or updated. The resulting descriptions are then used to update the IR server's index. The synchronizer also removes descriptions of deleted objects from the IR server's index. The synchronizer's transaction concept relaxes the isolation (serializability) requirement of the ACID principle. This is possible because casual 'dirty reads' do not seriously affect the IR server [Knau96].

  If there are several IR servers in a node, it is not necessary to assign a separate synchronizer to each server: in contrast to the illustration in figure 1 a single synchronizer could do the job as well.

The user client sends a query to one or more root IR servers (if there exist several sub-collection classes at the highest layer, there will be one root IR server for each class). The query then passes a number of nodes of the hierarchy and eventually a ranking list containing document names (possibly accompanied by abstracts) is returned to the user. The user can then choose some documents for presentation.

Nodes — especially those at higher layers — can get overloaded because of concurrent access by multiple users. This can be avoided by replicating the nodes.

By allowing almost any sort of media object database system to be integrated in our architecture, we get an IR system which is very flexible and can be easily extended. But when dealing with media objects (especially audio and video objects) even a more sophisticated system architecture would not be able to compensate deficits in supporting media presentation of the underlying database systems. To put it in other words: high quality real-time presentation of media objects is only possible if adequately supported by the media object database management systems.

An example for such a system is the *Media Object Storage System* (MOSS) [KMM94] which originates from University of Erlangen-Nuremberg and is now under development at Dresden University of Technology. MOSS provides an interface for managing media objects like image, audio and video objects. These objects can be accessed without knowing their storage format (data independency). Currently, we are working on real-time services for MOSS that preserve the data independency property. Additionally, MOSS supports content-based management of media objects by introducing a concept called *search sets*. From the IR system's point of view such search sets provide meta data which can be exploited for indexing sub-collections that are stored in a MOSS-based database.

Another hierarchical architecture for a distributed IR system has been proposed by the CAFE project [CN96]. This project deals with information retrieval in distributed text databases of giga byte order. However, the modules which are responsible for integrating text databases do not have their own retrieval functionality — in contrast to the layer-1-nodes of our architecture.

5 Conclusions

In this paper, the basics of a probabilistic model for retrieving information in a distributed heterogeneous multimedia document collection have been presented. This model could be further extended in many ways. For instance, other non-distributed probabilistic models which are more powerful than the BIR could be integrated. A process of selecting sub-collections must be specified exploiting effective methods for estimating the variance of the probability of relevance of a sub-collection and taking certain cost factors into account.

If the underlying data volume is allowed to grow and alter dynamically, determining useful indexing vocabularies for indexing sub-collections is is a challenging issue as well. Currently, we think of so called N-gram profiles (N-grams are parts of document features of length N) as being one of (probably) several suitable solutions to that problem, compare [CN96]. This still has to be proven experimentally.

The proposed model provides the foundation for an implementation of a distributed IR system. The system's architecure is especially designed for integrating different types of highly dynamic database systems. In particular, it is intended to integrate the media object storage system which is continued being developed by our database group.

Acknowledgement: The authors would like to thank Prof. Dr. Peter Schäuble and his research group from ETH Zurich for many fruitful discussions.

References

[BDGj96] D. Boles, M. Dreger, and K. Groß johann. MeDoc Information Broker. Harnessing the Information in Literature and Full Text Databases. In *Proceedings of the SIGIR'96 Workshop Networked Information Retrieval (Zurich, Switzerland)*, 1996.

[CN96] G. Crowder and C. Nicholas. Recource Selection in CAFE: an Architecture for Networked Information Retrieval. In *Proceedings of the SIGIR'96 Workshop Networked Information Retrieval (Zurich, Switzerland)*, 1996.

[CZC95] J. P. Callan, L. Zhihong, and W. B. Croft. Searching Distributed Collections With Inference Networks. In *Proceedings of the 18th Int. ACM SIGIR Conf. on Research and Development in Information Retrieval*, 1995.

[Fuh96a] N. Fuhr. Object-oriented and Database Concepts for the Design of Networked Information Retrieval Systems. *To appear in: Proceedings of the 5th Intl. Conf. Information and Knowledge Management (CIKM '96)*, 1996.

[Fuh96b] N. Fuhr. Optimum Database Selection in Networked IR Systems. In *Proceedings of the SIGIR'96 Workshop Networked Information Retrieval (Zurich, Switzerland)*, 1996.

[KMM94] R. Käckenhoff, D. Merten, and K. Meyer-Wegener. MOSS as Multimedia Object Server Extended Summary. In R. Steinmetz, editor, *Multimedia: Advanced Teleservices and High Speed Communication Architectures, Proc. 2nd Int. Workshop - IWACA '94 (Heidelberg, Sept. 26-28, 1994), Lecture Notes in Computer Science vol. 868*, Berlin, 1994. Springer-Verlag.

[KS96] D. Knaus and P. Schäuble. The System Architecture and the Transaction Concept of the SPIDER Information Retrieval System. *To appear in: Bulletin of the IEEE Computer Society Technical Committee on Data Engineering*, 1996.

[Rob77] S. E. Robertson. The Probability Ranking Principle in IR. *Journal of Documentation*, 33(4), 1977.

[RSJ76] S. E. Robertson and K. Sparck-Jones. Relevance Weighting of Search Terms. *Journal Society of Information Science*, 27, 1976.

[RW94] S. E. Robertson and S. Walker. Some Simple Effective Approximations to the 2-Poisson Model for Probabilistic Weighted Retrieval. In *Proceedings of the 17th Annual Int. ACM SIGIR Conf. on Research and Development in Information Retrieval*, 1994.

[Sch93] P. Schäuble. SPIDER: A Multiuser Information Retrieval System for Semistructured and Dynamic Data. In *Proceedings of the 16th Int. ACM SIGIR Conf. on Research and Development in Information Retrieval*, 1993.

[Sch96] P. Schäuble. Multimedia Information Retrieval. Kluwer Academic Publishers, Boston, not yet published, chap. 1-3, 1996.

[TMRMW96] H. Thimm, U. Marder, G. Robbert, and K. Meyer-Wegener. Distributed Multimedia Databases as Component of a Teleservice for Workflow Management. In *Proceedings of the 3rd Pacific Workshop on Distributed Multimedia Systems 1996 (Hong Kong, June 25-28)*, 1996.

[VGJL94] E. M. Voorhees., N. K. Gupta, and B. Johnson-Laird. The Collection Fusion Problem. In D. K. Harman, editor, *Proceedings of the 3rd Text Retrieval Conference (TREC-3)*. NIST Special Publication 500 - 225, 1994.