Resource-limited information retrieval in a distributed environment

Daan Velthausz, Henk Eertink, Jack Verhoosel, Jeroen Schot

Telematics Research Centre
P.O. Box 589, 7500 AN Enschede
The Netherlands
{D.Velthausz, H.Eertink, J.Verhoosel, J.Schot}@trc.nl

Abstract

In commercial information retrieval, a trade-off exists between relevance, cost, and time. We present a sound theoretical framework for resource-limited information retrieval that enables a user to search for relevant information given time and cost constraints. To use this framework, information providers have to specify estimates of search and retrieval times in addition to contents and accounting information.

1 Introduction

Commercial information providers have accounting policies related to access, connection time, number of retrieved items etc. Most users of these services try to get relevant information at low cost. For non-distributed information services, like Excerpta Medica, and the National Library of Medicine, many organisations use trained staff. In a distributed information retrieval environment staff training is hard due to dynamics in information services and user requirements. At the moment little support exists for cost-effective information retrieval. In this paper we present a sound theoretical framework for resource-limited information retrieval that enables a user to search for relevant information given time and cost constraints.

A retrieval process starts when a user interacts with the retrieval system specifying a, for the system understandable, query reflecting his information need[1]. For example the need 'show me the two most relevant video fragments of a soccer match where AJAX is scoring a goal as fast as possible but within 5 minutes and not exceeding $3,-', expressed in an SQL like notation may look like: SELECT video scenes WHERE time < 5 m., money ≤ 3$, n=2, properties (AJAX, scoring goal, soccer match) ORDERED BY relevance OPTIMISED BY (relevance/time).

Given this query the retrieval system should decide where to start searching, as in general the user does not (need to) specify where to search. Before actually spending time (and money), an estimate of the most useful search action should be made, e.g. taking the probability of finding relevant information and the expected cost into account. As the search proceeds, the system explores unknown parts of the information environment and updates its view accordingly. When relevant information has been found, the system must decide whether to retrieve it or continue searching trying to find 'better' information (e.g. more relevant or cheaper). As the cost specified in the query refers to the total cost, the system needs to decide how much budget should be spent on searching and how much on retrieving. This is in fact a fundamental problem when developing a resource-limited retrieval strategy. We have visualised this problem in Figure 1. Determining what actions to choose when limited resources are available has been tackled in the areas of artificial intelligence, robotics, and theoretical computer science as part of a more general problem of a system acting rationally: do the 'right thing' given its view of the (partially known) environment and its intended goal. This has resulted in a variety of approaches including on-line algorithms, real-time heuristic search, robot exploration techniques, sensor-based planning, resource-limited reasoning etc.

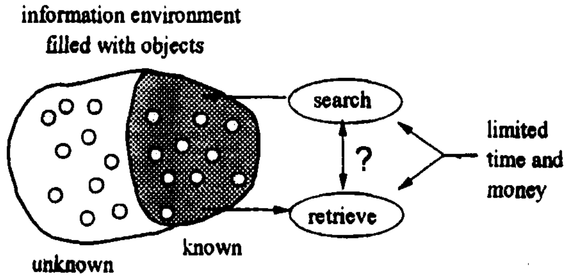

Figure 1. Deliberate between searching the information environment and retrieving objects from the searched environment with limited resources.

The rest of this paper is organised as follows. First we provide an overview of our approach in Section 2. Next we discuss the time and cost aspects we deal with. In Section 4 we present our resource-limited retrieval strategy. As the strategy is based on relevance and cost estimates we briefly

[1] When the user has no idea about the expected cost in relation to the quality of the results, the system should, as part of the interaction, inform the user about the expected quality-cost relation given a content-based query.

sketch how these can be obtained in Section 5. Section 6 is reserved for the conclusions.

2 Overview

We present a resource-limited information retrieval strategy that is able to search for relevant information given time and cost constraints in a partially known and distributed information environment. Fuhr (Fuhr 1996) tackles this problem using a probabilistic theoretical information retrieval approach. It allows the retrieval of a maximum number of relevant documents from multiple databases at minimum cost. This approach is based on many assumptions that are unrealistic in a real environment, e.g. availability of the (theoretical) recall-precision curves for all databases for all queries, the number of relevant documents within each database etc.

Given the partially known information environment, a search graph of possible actions can be constructed. The graph will gradually expand as the search proceeds, see Figure 2, and will be different for each search session, e.g. due to a different query, a different starting point, the dynamics of the environment etc. Considering the retrieval of relevant information objects as 'goal nodes', standard graph search strategies might be used, especially the ones capable of dealing with cost, such as A* (Hart, Nilsson & Raphael, 1968). They are, however, not suitable for resource-limited information retrieval as they have no imposed constraints on the resources used during the search. They try to find the best (or any) solution (or goal state) and use cost as a criterion to indicate the usefulness of (the path to the) found solution. Extending the criteria for these algorithms to stop searching when it runs out of time and/or money, ignores the optimisation between time, money and object relevance specified by the query.

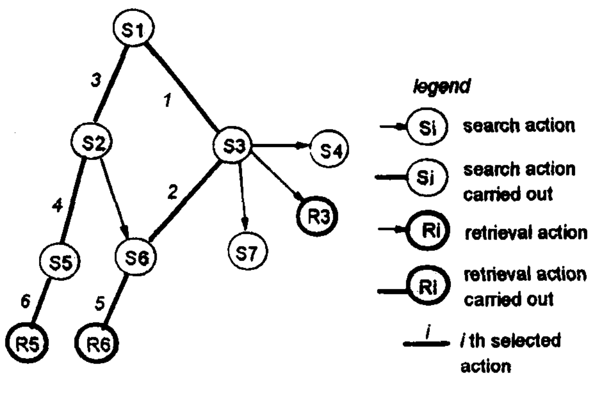

Figure 2. Example a gradually expanding action search graph

For resource-limited information retrieval we need an approach that selects the right action given its partial view of the environment under time and money constraints. We use Russell and Wefald's metareasoning decision theory (Russell & Wefald 1991) since it provides a sound theoretical basis. As far as we know, it has not been applied in the context of distributed information retrieval. A fundamental idea behind the theory is the principle of maximum expected utility, that is to choose the (external) action that yields the highest expected utility (reflecting the degree of usefulness), averaged over all possible outcomes of the chosen action. In this theory the computational steps, required to choose the 'best' action, are considered as (internal) actions and are therefore selected on the basis of their expected utilities (which is derived from the passage of time and the possible revision of the intended (external) actions). We apply the meta-reasoning decision theory to a resource-limited distributed information retrieval context. When considering a search action as an internal action and a retrieval action as an external action, Russel's theory provides a solution for the deliberation between search and retrieval actions, see Figure 3. Note that a retrieval action can only be performed when an object has been found. The expected utility of search and retrieval actions depends on the optimisation criterion specified in the query and is based on the provided estimates information for the (query dependent) relevance and involved cost of the actions.

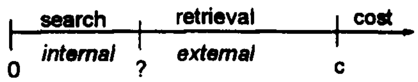

Figure 3. Problem of assigning allowed cost to search (internal) and retrieval (external) actions.

3 Cost aspects

In this section we discuss the time and cost aspects we deal with. First we show how cost aspects can be included in queries, thereafter we specify one cost model that basically consists of search and retrieval cost.

3.1 Query model

In our approach a query consists of different expressions, viz. a retrieval expression, a starting expression, a filter expression, a ranking expression, and an optimising expression. Using an SQL like notation, this is expressed as follows:

SELECT retrieval expression
FROM starting expression
WHERE filter expression
ORDERED BY ranking expression
OPTIMISED BY optimising expression

- The *retrieval* expression states which properties of the filtered and ranked information objects the user wants to retrieve (and or present), e.g. 'title of the books'.

• The *starting* expression states the potential starting point(s) from where the retrieval system should start searching.

• The *filter* expression restricts the amount of potential suitable information objects to those objects that fulfil the constraints of the filter expression. These constraints may refer to individual information objects, e.g. minimum relevance of an object, but also to the search session, e.g. total cost < c, total time < t, or may restrict the amount of objects to be retrieved. A degree of importance can be attached to the constraints to indicate their strictness, e.g. via weights, priorities or the risk of overspending. In our current strategy, we have included a simple mechanism resulting in three strategy variants, viz. a cautious strategy, an opportunistic strategy, and a common strategy.

• The *ranking* expression is used to determine the presentation order of the information objects satisfying the filter expression. It may include relevance and cost aspects. The filtering expression (e.g. value > threshold) can be used as ranking expression (using the value) when it is based on the same properties.

• The *optimising* expression (or objective function) specifies the 'optimal' selection of information objects that fulfils the constraints. For example, a user may want to maximise the relevance of the results at minimum cost. This means that the closer a set of information object is to the 'optimum', the more useful the retrieved set of the information objects is judged. See Figure 4 where, within the constrained area, a darker colour represents a more useful solution. When the objective function includes both time and money aspects they need to be combined. This requires a form of normalisation or transformation, e.g. assign 'monetary' cost to time, which is common in economics.

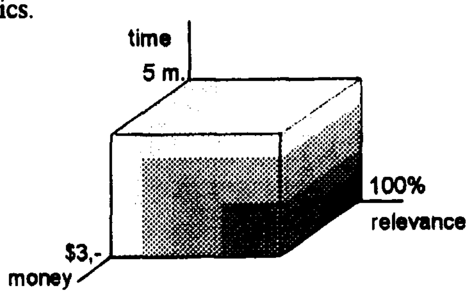

Figure 4. Visualisation of a cost related query including a time constraint (< 5 minutes), a money constraint (< 3$) and the coloured objective function: relevance/time).

Remember that the cost aspects included in the constraints and objective function reflect the *total cost*, and not separately for retrieval cost and search cost. As users may have different constraint and objective functions, the retrieval system should be capable dealing with any (user-defined) constraint and objective function including queries like minimise the total cost for the search and retrieval of a fixed number of relevant information objects; maximise the amount of (the top most) relevant information objects for fixed costs; maximise the sum of relevance scores/the score-cost ratio (for a fixed number of relevant information objects) for fixed cost.

3.2 Search and retrieval cost

We distinguish two categories of cost: *search cost* and *retrieval cost*. The search cost are all cost that are involved identifying relevant information objects given the query. The retrieval cost are all cost involved of retrieving, transporting, and presenting the requested properties from the selected (most useful) information objects to the user. The search and retrieval cost in a real information environment are comprised of multiple cost aspects, see Figure 5. Although the retrieval system ultimately needs to handle all these cost aspects, for clarity reasons we describe the strategy (in Section 4) using the (aggregated) search and retrieval cost as this is sufficient to illustrate the main problems we solve in our strategy.

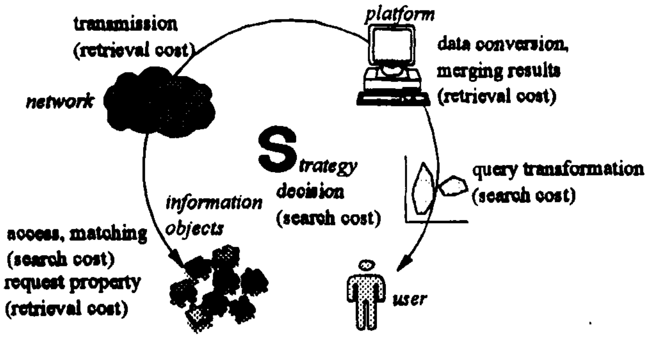

Figure 5. Cost aspect related to the context of a retrieval strategy

As each action costs time (and money), an estimate of the cost of an action is required preceding the execution of the action to avoid too costly actions with respect to time and money constraints. Determination of perfect estimates is impossible in our context, because the relevance of objects is query dependent and can only be exactly determined when the object is accessed. In Section 5 we briefly discuss how to obtain these estimates.

4 Retrieval strategy

In this section we present our proposed retrieval strategy. We start with an outline.

4.1 Outline of the strategy

At a high abstraction level our strategy works as follows:

1. Based on the query determine all possible start actions.
2. Remove all actions which do not satisfy the time, money and/or relevance constraints.

3. When all actions are filtered out, terminate the search session.
4. For each new action Ai passing the filter rule, compute its utility U(Ai)
5. Select the action (Aα) with the highest utility U.
6. Determine whether the utility U(Aα) is higher than the session utility U(**A**). If U(Aα) is not higher than U(**A**) then terminate the search session, else carry out action Aα.
7. Update the view of the information environment, the time-spent, money-spent, **A**, U(**A**) and determine all new possible search and retrieval actions.
8. Go to 2

In the next sections we discuss the steps thoroughly.

4.2 Determine start actions

For a large dynamic information environment it is very unlikely that it is completely known to the strategy. Furthermore, exhaustively searching all available collections within reasonable time seems not feasible. Hence, the strategy should determine where to start searching. The actual starting point for the strategy is the root of the 'action search graph', a virtual node representing the information environment the search is conducted in, see Figure 6. We assume that the strategy is aware of at least the access information (such as address and access mechanism) of the start (search) actions. This information is either provided by the query (in the FROM clause) or encoded in the strategy as default(s). In addition, other information about the action related object(s), such as coverage, type etc., may be available, e.g. stored from previous search results or obtained via polling (as is done by many Internet search engines). Nonetheless, due to the size and the dynamic nature of the information environment, it may not be worthwhile to maintain this information. When this information is not or only partially available, the strategy cannot determine the cost involved in executing an action nor the usefulness of an action.

The strategy may restrict itself to the fully labelled transitions or may actively acquire the missing information by accessing uninformed objects. Although additional (access) cost are made, this approach may eliminate unnecessary searches and hence save cost. We do not elaborate on this problem here, but assume now that the strategy only chooses among sufficiently informed actions.

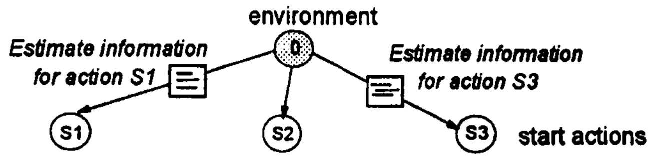

Figure 6. 'Environment root' node with and without estimate information for the start-actions.

4.3 Filtering actions

The strategy needs to remove all actions that do not satisfy the constraints specified in the query. This means that for each action that is not removed, it is estimated that it can be executed within the time and money budget left. Because searching is in general only useful to a user when (a property of) a relevant information object is retrieved, not only the cost of a search action itself should satisfy the constraints but also the (estimated) remaining search cost to find a relevant information object as well as the (estimated) retrieval cost of this object. This means that only when a relevant object is expected to be found and that can be retrieved within the constraints, the search action is acceptable. E.g. in Figure 7 search action S1 and retrieval action R3 are acceptable whereas search action S2 will be filtered out.

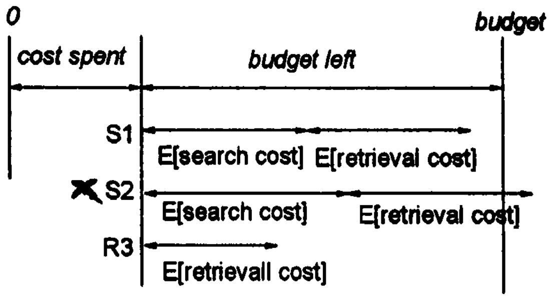

Figure 7. Example of action filtering. Search action S2 is filtered out because its estimated search and retrieval cost do not satisfy the budget left constraint.

As we will discuss in section 5, the cost estimates need not necessarily to be single valued (e.g. the average) but may consist of multiple values, like minimum, maximum etc., or in general be a probability distribution function. The use of probabilities implies that there exists a risk of overspending. This is tackled in our strategy via three different types of filter rules:

- Opportunistic filter rule: *remove an action if there is no or a very little chance that it will satisfy the constraint, e.g. upper bound < filter value.*
- Common filter rule: *remove an action if it is expected not to satisfy the constraint, e.g. average < filter value.*
- Cautious filter rule: *remove an action if there is a chance that it may not satisfy the constraint, e.g. lower bound < filter value.*

Our current strategy terminates when all actions are filtered out.

4.4 Action utility

Given the estimations of the time, money, and relevance of different actions, the question remains how these different estimates can be combined into a single decision criterion. As the search should be optimised according to the objective function specified in the query, we advocate to use this function to estimate the utility of individual actions. As the objective function is defined for the complete session, the cost and relevance parameters need to be adjusted to reflect the individual action aspects, see Figure 8. For the same reason as for filtering search actions, the utility of a search action should be based on the remaining search cost to a relevant object as well as the estimated retrieval cost and estimated relevance of this information object. For example, suppose we have an objective function relevance/cost. Then the estimated utility of a search action j will thus be E[relevance object j]/(E[search cost] + E[retrieval cost]). In addition, the estimated utility for a retrieval action is based on the estimated relevance and estimated retrieval cost.

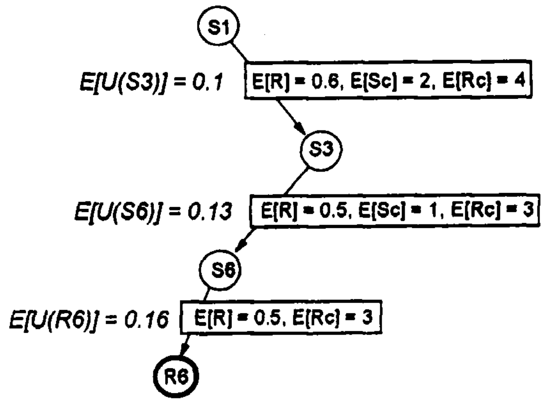

Figure 8. The estimated utility of search action S3, E[U(S3)], and S6, E[U(S6)], are based on the relevance estimate E[R], the estimated remaining search cost, E[Sc], and the estimated retrieval cost E[Rc]. The utility of the retrieval action R6, E[U(R6)], is based on estimated relevance E[R] and estimated retrieval cost E[R]. Legend cf. Figure 2.

In section 5 we discuss how these (query dependent) relevance and cost estimates can be obtained prior to the execution of an action. As these estimations are independent of the actions carried out by the strategy, the utility of an action does not change. Hence, only the utility of new actions need to be computed.

4.5. Action selection

It will be clear that, when money and time aspects are taken into account, the outcome of a search depends on the order of the performed actions. In case there are multiple actions satisfying the constraints, the strategy needs to choose. Throughout the search the strategy continuously has to decide what action to carry out. The best action to choose seems to be the one with the highest expected utility, cf. the maximum expected utility principle (Russel & Wefald 1991). Simply using the expected money cost in the computation of the utility works fine when a large number of searches are performed, e.g. for the search service provider, and these average cost are charged to the user. For a single search this is not satisfactory because the average does not reflect the real cost of the retrieved information object. From (Lomio 1985) we know that users are very cost sensitive and do not want to pay an unspecified amount of money for an unknown amount of data. Failures in estimating the costs in advance generally lead to complaints about the system. Instead of considering only the expected utility, different (measures of the) utility distribution can be used to cope with different variants of the strategy (e.g. cautious, common or opportunistic). For example in (Palay 1985) rules are described to select the most appropriate alternative when they are characterised by a minimum, average and maximum value for a chess game. The application of this approach in our context, assuming that the average, upper and lower bound utility scores are computed, leads to the following rules, each representing one of the three strategy variants:

- Opportunistic selection rule: *choose the action with the highest utility upper bound, if equal choose the action with highest utility lower bound, if equal choose randomly*
- Common selection rule: *choose the action with the highest average utility, if equal choose the action with the highest utility upper bound, if equal choose the action with the highest utility lower bound, if equal choose randomly*
- Cautious selection rule: *choose the action with the highest utility lower bound, if equal choose the action with the highest upper bound, if equal choose randomly.*

Instead of using the upper bound, lower bound and average values, one can use any (combination of) measurement(s) of the (multi-aspect) utility distribution to compare the alternatives, e.g. a virtual upper bound using the standard deviation, so that e.g. 95% of the values are below this point.

4.6. Termination

We use the filter and selection rules of the previous section, to choose between search and retrieval actions. In addition the utility of actions can also be used to determine whether to proceed the search session or to stop. In case no search or retrieval action passes the filter rule, the search stops. When the strategy still has actions to choose from, we use the relation between the *action utility* and *session* *utility* as a stop criteria. The utility of a search session is based on the objective function and includes the session results (e.g. the sum of the retrieved objects) and the total cost spent. On the contrary, the action utility is based on the results and cost related to the particular action. It may be clear that when the search proceeds the session utility is likely to change as cost is continuously spent. Remember that the utility of possible actions does not depend on the actions carried out by the strategy.

We argue that the search session should be proceeded with the 'best' possible action when the utility of this action is higher than the current session utility, because executing the action will increase the session utility. (Remind that for the utility of a search action both the search cost and retrieval cost are included).

We illustrate this with an example. Let **A** denote a search session: a sequence of actions (A1 ... Ai); Aj is action j. Aj can also be denoted as Sj when Aj is a search action and Rj when Aj is a retrieval action. U(Aj) is the utility of action Aj; U(**A**) is the session utility and U(**A**.Aj) is the session utility after executing action Aj.

It can be proven that when U(Rj) > U(**A**) this means that U(**A**.Rj) > U(**A**) for the object function: sum of relevance/cost. For example, when U(**A**) = r/c and U(Rj) = rj/cj then U(**A**.Rj) = (r+rj)/(c+cj). When rj/cj > r/c this implies that (r+rj)/(c+cj) > r/c.

When Rj is the retrieval action included in the estimation of U(Sj) and perfect relevance and cost estimations are available, then U(Rj) > U(Sj), and thus U(Sj) > U(**A**) hence U(**A**.Sj.Rj) > U(**A**). See Figure 9 for an illustration based on the search graph of Figure 2).

Remember that we have adapted the objective function (relevance/cost) to compute the utility for search and retrieval actions. We assume that for other objective functions the utility of search and retrieval actions should be adapted in a way that whenever U(Ai) > U(**A**) it implies U(**A**.Ai) > U(**A**).

Due to inaccurate estimations U(Rj) > U(Sj) may not be true. However, because the accuracy of the estimations is not known in advance and the utility of an action may be a probability distribution function, different (termination) rules can be applied reflecting different strategy's variants. We use the following rules:

- Opportunistic termination rule: *terminate the search when there is no current action Ai available with a utility upper bound U(Ai) > U(**A**)*
- Common termination rule: *terminate the search when there is no current action Ai available with an average utility U(Ai) > U(**A**).*
- Cautious termination rule: *terminate the search when there is no current action Ai available with an utility lower bound U(Ai) > U(**A**)*

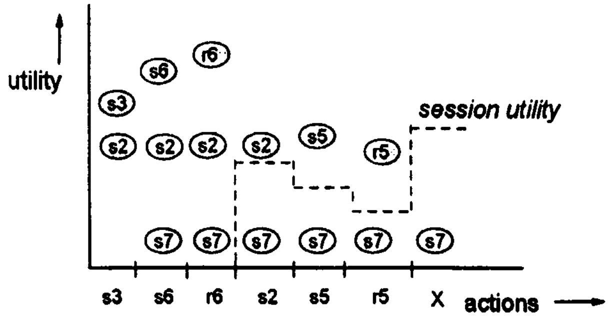

Figure 9. Utility of search actions (sj), retrieval actions (rj) and session utility (dashed line) after different actions are carried out cf. the action search graph of Figure 2. The session is terminated when no U(Aj) < U(**A**)

4.7 Execution of the best action

When the best chosen action is carried out, the current view of the information environment is updated accordingly. Furthermore, the time (and money) spent parameter(s) are updated, resulting in a smaller budget left. Also the new session utility U(**A**) is computed. Given the new view of the environment all new possible search and retrieval actions are determined.

5 Estimation of cost and relevance

As described in the previous section, the computation of the utility of an action might use relevance, search, and retrieval cost estimates. It is the responsibility of the information services to provide these (query dependent) estimates to the strategy. A practical problem might be the willingness of a provider to give insight in his tariff policy. Even if he provides descriptions, on which the estimations are based, one should be aware that this could deliberately be misleading information, e.g. resulting in high relevance and low cost, to attract customers. This can be avoided when the accuracy of the estimations are learned based on the results of previous search sessions. This can be done using similar approaches developed for dynamic network routing algorithms, like EGP or BGP.

To get an idea how to obtain the estimates we briefly sketch the ideas behind the mechanism of the ADMIRE information model (Velthausz, Bal & Eertink 1996). This model provides a general framework for modelling any information, regardless of type, size or abstraction level. See Figure 10 for an example of the information object structure. Via the notion of different kinds of information objects and (informed) relationships, it facilitates a gradually expanding action search graph, as illustrated in Figure 2 (Section 2), with labelled transitions that can be used to estimate the (query dependent) relevance and involved cost of the actions.

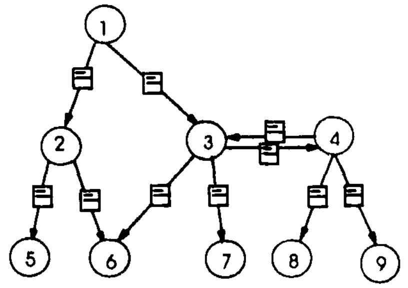

Figure 10.Example of ADMIRE information object structure

5.1 Relevance estimate

We need an indication of the relevance of unexplored information objects that are reachable via, or contained in, another information object is without accessing them. The problem is that the relevance of information objects is query dependent. Given the wide range of possible queries in a large and dynamic environment, it is impractical to reuse relevance estimations of previous searches. Another problem in a distributed environment is that it is likely that different matching methods (which determine the relevance between the query and an information object) are used. The scores obtained from multiple collections may therefore *not* be directly comparable. To combine different scores into a general, universally applicable, score is a problem that we will not diverge into. We refer to (Callan, Lu & Croft 1995; Voorhees, Gupta & Johnson-Laird 1995) that lists a number of possible approaches that can be used. Here we assume that the relevance scores of different information objects are 'normalised scores', meaning that they can be directly compared with each other.

The ADMIRE information model facilitates aggregation and propagation of information that characterises reachable information objects. Using the composite relationships in the object hierarchy enables a bottom up propagation of and aggregation of the lower layered object characterisations. This information can then be used to estimate the relevance of unexplored information objects. For example, the coverage of a database containing soccer and tennis video fragments might be characterised by (aggregated) concepts such as sport, game etc. Matching these concepts with the query provides a relevance estimate for the video fragments objects. A way to increase the accuracy of the relevance estimate is to enhance the query with related concepts, e.g. synonyms, hypernyms etc., before matching it with information object descriptions.

Using summarised information to describe particular aspects of the lower layered nodes in a hierarchy, has successfully been applied in (Gravano & Garcia-Molina 1995) and (Garcia-Molina, Gravano & Shivakumr 1996) for the content-characterisation for (hierarchical) databases containing textual documents.

5.2 Cost estimate

In addition to estimating the relevance, an accurate estimation of the remaining search cost to find the requested information objects as well as the retrieval cost of these objects is needed. As the cost may (partly) depend on the query, the structure of the query (e.g. the operators, the requested number of objects etc.), can be used to estimate the remaining search and retrieval cost, as for example is done in (Chaudhuri & Gravano 1996).

The ADMIRE model has not been developed to support cost estimations, but it can simply be extended using the bottom-up propagation and aggregation of cost-related information. In this manner cost distributions can be obtained reflecting different cost aspects. Instead of the actual distribution function, derived measures such as average, minimum, maximum, variance, etc. can be used. For example, the retrieval time can be estimated using the propagated object size distribution, the throughput and network delay. The latter two are either known to the strategy before the search is performed or can be estimated via the exchange of messages.

A potential problem with upwards propagation of information is the topicality of the aggregated information. Since this information is obtained prior to the actual search, changes in costs, adding new or deleting information objects might not be directly reflected in the aggregated information. Furthermore, a consequence of the use of the aggregation methods is that the cost and relevance estimates are independent of each other and therefore may lead to inaccurate utility estimations.

As non-composite relationships only enable accurate estimations of directly reachable objects (look-ahead of 1), estimated utilities based on composite relationships are likely to be more accurate and hence should be preferred over non-composite relationship based utilities.

6 Conclusions

In this paper we presented a sound theoretical framework for resource-limited information retrieval that enables a user to search for relevant information given time and cost constraints. We have shown that a limited meta-reasoning theory can be applied to resource-limited distributed information retrieval. To verify and validate the retrieval strategy we will use a prototype in a distributed and heterogeneous 'office' environment based on the ADMIRE model, that covers multiple servers, and contains multiple (different) information objects.

Acknowledgments. We thank R.R. Bakker and P.M.G. Apers for their contributions.

References

Callan, J.P., Lu Z., Croft Z. 1995. Searching Distributed Collections With Inference Networks. In Proceedings of 18th ACM SIGIR 1995.

Fuhr, N. 1996. A Decision-Theoretic Approach to Database Selection in Networked IR Systems, In Proceedings of the SIGIR 1996 Workshop on Networked Information Retrieval, ETH Zurich, Switzerland, http://SunSite.Informatik.RWTH-Aachen.DE/Publications/CEUR-WS/Vol-7/.

Garcia-Molina H., Gravano L., Shivakumr N., 1996. Finding Document Copies Across Multiple Databases, In Proceedings of 4th Int. Conference on Parallel and distributed Information Systems PDIS'96.

Chaudhuri, S., Gravano L. 1996, Optimizing Queries over Multimedia Repositories, In Proceedings of SIGMOD'96, Canada, pp91-102

Gravano L., Garcia-Molina H., 1995. Generalizing GlOSS to Vector-Space Database and Broker Hierarchies. In Proceedings of 21st VLDB Conference, Zurich.

Lomio J.P.,1985 The high cost of NEXIS and what a searcher can do about it. *Online* 9(5):54-56.

Ozsoyoglu et al., 1995, Time-Constrained Query Processing in CASE-DB. *IEEE Transactions on Knowledge and Data Engineering* 7 (6):865-884.

Hart, P.E., Nilsson N.J., Raphael B. 1968. A formal basis for the heuristic determination of minimum-cost paths. *IEEE Transactions on Systems Science and Cybernatics SSC* 4(2):100-107.

Palay, A.J. 1985, Searching with probabilities. Ph.D. Thesis, Carnegie-Mellon University, Pittsburg

Russel S., Wefald E. 1991. *Do the Right Thing*, MIT Press.

Velthausz, D.D., Bal, C.M.R., Eertink, E.H. 1996. A Multimedia Information Object Model for Information Disclosure. In Proceedings of the MMM'96, Third International Conference on MultiMedia Modelling, Toulouse, France, pp289-304

Voorhees, E.M., Gupta N.K., Johnson-Laird B., The Collection Fusion Problem. In Proceedings of TREC 3, 1995, pp95-104.